

Belial Hotel: Designing a Mystery Horror Game Environment

██████████ (██████████ Project unit
(6GZ1001) Supervisor: Rene Doursat

4/24/2017

Declaration

No part of this project has been submitted in support of an application for any other degree or qualification at this or any other institute of learning. Apart from those parts of the project containing citations to the work of others, this project is my own unaided work.

Signed.....

Contents

Declaration.....	1
List of Figures	4
<u>Chapter 1: Introduction</u>	5
1.1: Background	5
1.2: Report layout	5
1.3: Aims and objectives	5
1.3.1: Aims	5
1.3.2: Objectives	5
<u>Chapter 2: Literature review</u>	6
2.1: What makes a horror game scary?	6
2.2: Why do we play horror games?	8
2.3: Existing game engines.....	8
2.3.1: Unreal Engine 4.....	8
2.3.2: Unity Engine	10
2.3.3: CryEngine	10
2.4: Game development	11
2.4.1: 3d Models	11
2.4.2: Textures	11
2.4.3: Lighting.....	12
2.5: Existing Games:	12
<u>Chapter 3: Design</u>	13
3.1: Belial Hotel - Treatment Document.....	13
3.1.1: Abstract.....	13
3.1.2: Background	13
3.1.3: Game Genre	14
3.1.4: Target Audience	14
3.1.5: Look and feel.....	15
3.1.6: Game play	16
3.1.7: Goals and Challenges	16
3.1.8: Audio	16

3.2: Flow diagram.....	17
3.3: Main Menu (Start Screen) GUI.....	18
3.4: Options Menu GUI	18
3.5: Help Screen GUI	19
3.6: Gameplay Screen	19
3.7: Level design.....	20
3.8: UML.....	21
<u>Chapter 4: Implementation</u>	21
4.1: Unreal.....	21
4.2: Importing / Exporting and Asset creation.....	23
4.3: GUI	23
4.4: Blueprints	28
4.5: Characters	32
4.6: Matinee.....	32
<u>Chapter 5: Testing</u>	33
5.1: Protocol.....	33
5.2: Questions	33
5.3: Results.....	34
5.4: Feedback question	36
<u>Chapter 6: Evaluation</u>	37
6.1: Evaluation of aims and objectives	37
<u>Chapter 7: Conclusion</u>	38
Appendix	39
1.1: Terms of Reference	39
1.2: Course specific learning outcomes	39
1.3: Project Background.....	39
1.4: Aim	39
1.5: Objectives.....	40
1.6: Problems	40
1.7: Deliverables.....	40
1.8: Required Resources	40
References	41
Bibliography	41

List of Figures

Figure 1 - a screen shot from the horror game- " Amnesia: The Dark Decent"	7
Figure 2 – A screenshot from unreal engine 4.....	9
Figure 3 – A screenshot of the Unreal material editor.	11
Figure 4 – A screenshot of the Unreal Engine Prototype BSP Stage.....	15
Figure 5 – A flow diagram of the structure of the game in development	17
Figure 6 – Main Menu GUI (Design).....	18
Figure 7 – Options Menu GUI (Design).	18
Figure 8– Controls Menu GUI (Design).	19
Figure 9 – In game display (Design).	19
Figure 10 – General layout of hotel floor / Level design.	20
Figure 11 – UML design.....	21
Figure 12 – UML design (2).	21
Figure 13 – Screenshot from Unreal Engine BSP Editing	22
Figure 14 – Screenshot from Unreal Engine BSP Editing (2).....	22
Figure 15 – Fully implemented Main menu.	24
Figure 16 – Fully implemented Options menu.....	24
Figure 17 – Fully implemented Controls menu.....	25
Figure 18 – A gameplay Screenshot within the prototype.	25
Figure 19 – A gameplay screenshot of note widget.	26
Figure 20 – A gameplay screenshot of widget when key is collected.	26
Figure 21– Fully implemented game over end screen.....	27
Figure 22 – Fully implemented player wins screen.	27
Figure 23 – Unreal blueprint concept design.....	28
Figure 24 – Screenshot of unlocking door blueprint code.....	28
Figure 25– Screenshot of picking up a battery blueprint (within level blueprint).....	29
Figure 26 – Screenshot of blueprints within the battery blueprint.	29
Figure 27 – Screenshot of battery power decreasing within the Lewis blueprint.....	30
Figure 28 – Screenshot of torch’s blueprints within the Lewis blueprint.....	31
Figure 29 – Screenshot of progress bar widget blueprint.	31
Figure 30 – Screenshot inside the matinee editor.....	32
Figure 31 – Screenshot referencing matinee within blueprint.....	32
Figure 32 – Matinee Icon	32
Figure 33 – Results from “Did the game have a good atmosphere?” question	34
Figure 34 – Results from “Did the game make you feel scared at any point?” question	34
Figure 35 - Results from “Were there any predictable scares?” question	35
Figure 36 – Results from “Was there a constant build-up of tension?” question	35
Figure 37 – Screenshot of stat unit console command.	36

Chapter 1: Introduction

In this chapter, I will discuss what aims and objectives this project will meet and the steps in order to achieve them. I layout of the reports structure providing information of each chapter and a brief description of the background of the project.

1.1: Background

Horror games have become very popular over the past few years. This project will consider why there is popularity in this genre if the idea of the game is meant to scare the player. I will see what it takes to create a horror game and how other games use certain techniques to be the best in the horror genre and what techniques is important to avoid.

1.2: Report layout

Chapter 1: Introduction - This contains all the aims and objectives the project will try and complete.

Chapter 2: Literature review – This chapter will go into detail about what we can learn about horror games and what design methods are used.

Chapter 3: Design – This chapter talks about the overall design process for the prototype. This shows how the game should look and work as a game.

Chapter 4: Implementation – In this section is where all the design ideas have been used to create the game.

Chapter 5: Testing – The testing stage is where the prototype is play tested by participants and data is gathered.

Chapter 6: Evaluation - This chapter evaluates the testing data and how the prototype worked and how it matched up to the aims and objectives.

Chapter 7: Conclusion – Summarises the whole project, any bugs that were found and overall thoughts about future improvements.

1.3: Aims and objectives

1.3.1: Aims

The aim of the project is to create a horror game that takes into consideration the methods used on popular games and to avoid any mistakes commonly used when designing a game in this genre.

1.3.2: Objectives

To achieve the aim, I must:

- Gather information on what makes a horror game scary
- Design a game that will meet these techniques
- Implement the layout and designs using a game engine
- Create the functionality of the game using code.
- Test the prototype, compare results

Chapter 2: Literature review

2.1: What makes a horror game scary?

In this section I will be conducting research on how horror games differ from normal video games and what factors are required for it to be classified in the horror genre.

To start, we must understand what the definition of horror actually is. The definition of 'Horror' in the oxford dictionary is: *"An intense feeling of fear, shock, or disgust"*. Following this definition, it is understood that certain emotions such as fear, revolt, panic and shock need to be explored, to allow the game to be classified as scary and therefore fit into the horror genre.

This definition also addresses the three different types of horror and is the basis for many subgenres in the horror category. For example, disgust can be seen to influence the survival horror category which features games such as 'Left 4 dead.' This game focuses on the outbreak of zombies and the extreme body horror that leaves the player feeling revolted and disgusted. Games that fit into the survival horror category usually avoid trying to create fear throughout disempowerment as many survival games give the player a weapon, therefore horror has to be created in a different way throughout body horror.

"Horror games often seek to disempower rather than empower players in order to make them feel vulnerable and uneasy" (Donovan, 2011). Donovan's statement is reinforced throughout various horror games with one example being "Amnesia: The Dark Decent." Disempowerment is used throughout this game to make the player feel vulnerable and defenceless against the creatures lurking in the castle. At many points in the game your only option is to run and hide as you are exposed to the creatures lurking in the castle with no means of protection. Your only useful piece of equipment in the game is a lamp which helps to guide you along, whilst also keeping you sane. This game doesn't work on a health bar like many other horror games, but instead your mental stability deteriorates, especially if you spend lots of time in the dark.



Figure 1 - a screen shot from the horror game- " Amnesia: The Dark Decent".

As the definition states that shock is an emotion associated with horror, this is often played upon in many horror games. This feeling of shock is often created throughout mystery and uncertainty where the player feels on edge, not knowing what to expect. This will allow the player to be caught off guard, build the tension, and the unnerving feelings of what's around the next corner. Horror games often play on this to scare the player, creating a tense atmosphere even if there is nothing really to be 'scared of.' This causes the player to fear, fear itself and feel on edge throughout the whole of the game. Some horror films often use jump scares to break the tension and cause shock to the player. As stated in JumpScares, *"The "jump scare or "jolt scare," the horror films most basic trick, is a moment of surprise or terror that seemingly comes out of nowhere but is actually a synthesis of specific sights and sounds"* (Muir (2013)). These effectively cause a sudden and abrupt moment of shock to the player, as the tension has been building throughout. However, in certain games such as "Five Nights at Freddy's" they use a jump scare right before the game is over every time as an indicator the game is over. While this is a very effective way to scare the player, jump scares that are repeated in the exact same way often lose their impact causing the player to become desensitised to it.

Fear in horror games can also be evoked throughout the setting and the environment. This is one of the most important features in the horror genre as it has a huge impact on the player's belief and experience of the game. The correct lighting and assets in a horror game are crucial to creating a believable atmosphere that the player can be immersed into. A good example of this is "Slender: The eight pages," this game requires the player to walk around the forest collecting pages that are scattered around, each one displays a message and the goal is to collect all eight. However, as the player collects more pages 'Slender man' is closely following behind you and the longer the player spends looking for the pages the closer 'Slender man' becomes. With the game being set in a dark forest with only a torch as your only source of light, this creates a very ominous and tense atmosphere as the player can only see in the direction the torch is pointing. This makes finding the pages very difficult, but also tense as the player is always on edge as 'Slender man' can be lurking around any corner.

2.2: Why do we play horror games?

If the purpose of a horror game is to scare the player, then why do we continue to play them? This all depends on what the player is looking for when playing the game, whether that's adrenaline, wanting to indulge in experiences they will never have in their human life, or whether that's the after high from playing the game. For the latter, experiencing an after high from playing a horror game occurs as the player experiences tension and fear throughout the game, causing the player to feel on edge and therefore increasing their heart rate. As Madigan states *"People become physiologically aroused due to the fear they experience during the media event—and then when the media event ends, that arousal transfers to the experience of relief and intensifies it. They don't so much enjoy the experience of being afraid –rather, they enjoy the intense positive emotion that may directly follow."* (Madigan, 2015). Therefore, once the game has finished, the player will experience a huge sense of relief, creating a high that makes the whole experience worthwhile.

2.3: Existing game engines

2.3.1: Unreal Engine 4

Unreal engine 4 is a powerful game development software created in 1998 by Epic Games. This software allows the user to design beautiful 3D worlds using landscape design and being able to apply textures. Unreal allows the designer to remodel simple shapes by changing certain angles or extruding faces of that shape allowing total flexibility and creativity. The design can then assemble these shapes together and rearrange them to create amazing looking models and designs. The software not only allows you to apply textures to the shapes to create character and authenticity but also allows you to create and apply your own bespoke materials. They can design how rough or shiny the texture will be along with multiple other factors that allows the designer to be as creative and imaginative as they require. Unreal engine has a multitude of different features which makes it such a popular tool when it comes to design AAA games.

Unreal has other features such as landscape design, this allows the designer to create different types of mountains and raised platforms, which means there is massive flexibility when creating 3D worlds. Along with the landscape design, there is a foliage feature, this allows the designer to place different types of foliage in certain areas, where it is needed. This feature is also very flexible as it can be scaled to very little foliage or in high quantities which

helps speed up the process.



Figure 2 – A screenshot from unreal engine 4.

When developing my game, I will need to think of what software will be most compatible with creating the game. I will need to weigh up the pros and cons of a few software's to see what is the right fit for me. Unreal has a lot of positive features, some of which I've mentioned above. Other factors that make it a very good choice, is that the software is completely free to download. *"Unreal Engine 4 is completely free (including all future updates!) to download and use. You get all the Unreal engine tools, free sample contents, complete C++ source code which includes code for the entire editor, and all of its tools".* (PV, 2016) There is a 5% royalty cost that unreal will take if the game is published and creates revenue, however this seems insignificant. Other positive features that really set it above other software's, is the fact that you can use 'Blueprint Visual Scripting'. *"Blueprint Visual Scripting in unreal Engine 4 is an extremely powerful and flexible node-based interface to create gameplay elements and provides artists and designers with the ability to program their game and to quickly iterate gameplay within the editor without writing a single line of code!"* (PV, 2016) This is very useful as the design doesn't need to be completely familiar with code and can visually see the blueprint elements.

The negative aspects of using unreal, is that due to the number of extensive features, the platform can seem quite complex and take a while to properly understand. This means the user must become educated and efficient with the platform before use and the software also requires a computer with quite high specifications in order to make the most out of the system. Unreal 4 is a very powerful and useful software to use due to the blueprint system and use of multiple different features.

2.3.2: Unity Engine

Unity, developed by Unity Technologies in 2005 is a game development software. This software is very powerful when it comes to making games for less powerful systems. This means that unity is able to create games that can be used on mobile phones, as they don't have as much processing power as a desktop PC, they can even produce games that can be played on the internet. Unity can develop games in 2D and 3D, so therefore, when creating a 2D game unity is the better choice than unreal. However unreal is making changes in order to add more 2D features. Unity's features work in the same way as unreal the designer can still add assets and resize and position objects in order to create the game. When programming in unity the designer is able to code the game using C# and JavaScript. These are very popular coding languages and can offer really flexible scripts. In comparison with unreal the price of unity is also free, however, there is a unity pro option that allows access to more features that the basic version does not have.

Unity seems very good at being able to produce fast and flexible games that can run on nearly any device and with the option to create 2D games makes it a very good choice when making smaller type games. However, since my game will be a PC game and will need to resemble a hotel, graphics and 3D modelling will be very important in order to make the player feel immersed within the game, Unity might not have the features I require.

2.3.3: CryEngine

CryEngine is a game development software that was designed by Crytek in 2002. CryEngine works very similar to Unreal Engine 4, it is used to create fantastic looking games by using the material editor features. The features in CryEngine are also very similar to Unreal, CryEngine allows the user to create and edit geometry so the designer is able to create levels and structures, the engine also allows the user to randomly generate landscape by giving it a few values such as max height and the overall area measurement and it will generate a land scape and the user can keep generating a landscape till they have one that they are happy with. The landscape look very realistic and are excellent for making open world games. However important materials seems a lot more difficult as it requires a .tiff file compared to unreal which will expect most common image files such as .jpg and .png. However, the cost of CryEngine is free and there is no royalty fee like there is in Unreal.

Programming wise, CryEngine seems to be very dependent on programming therefore maybe very difficult to learn everything to get the maximum features out of the programme, compared to unreal which has blueprint.

CryEngine is a very good piece of software it has the potential to create beautiful looking games and environments. However, the user interface is not the most simple to learn as it is in unreal, it also seems like the engine relies on having a team of people to produce a game rather than the ability of one person as there are elements that require extensive knowledge in that certain area.

2.4: Game development

2.4.1: 3d Models

Creating 3D models is very important in game development as this allows the developer to create unique models that are relevant for that game. This could be anything from a door to a character. To do this, it is important to get an understanding of how it works. Using software's such as Maya allows the users to create polygons, extrude and mould these objects into shapes that represent something. The less polygons or faces the object has is often referred to as 'low poly' and will only look very basic. However, introducing more polygons creates finer detail and a better-looking model. This will take more computing power to produce high poly models. Therefore, the developer should understand what kind of game they are trying to create, as some computers or game consoles won't be able to play models with too many polygons. Within the game engine Unreal there is a feature that allows the user to model shapes, not to the detail of what Maya can produce but it is still very possible to model in Unreal and create that model to a static mesh which will allow the user the ability of animating the mesh.

2.4.2: Textures

Textures are also a very important feature when developing games. These give plain assets the right design to create the feel of the game. For example, horror games will require dark and spooky textures to give it an ominous atmosphere. The unreal software allows the user to import images such as a wallpaper image and create a material from that texture, a material is a texture that can be assigned to a mesh. A material uses the texture and alters the way that texture will

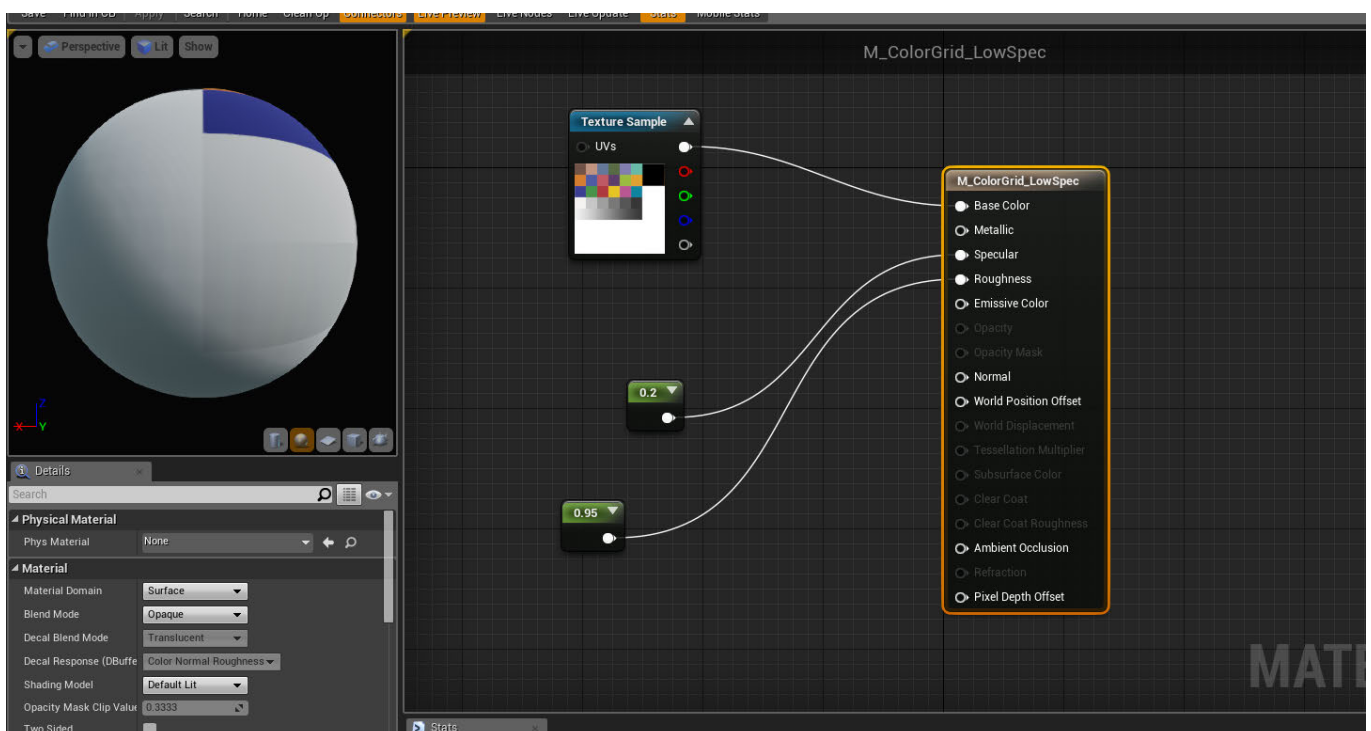


Figure 3 – A screenshot of the Unreal material editor.

appear on a mesh, the way this is possible is by connecting the main texture to the base colour and allying values to the options such as roughness. Depending on the value, the texture will appear to have a more of a rough and realistic look, however to make it look even more realistic you can add a normal mapping and Specular Mapping. Normal mapping is a way of faking or creating bumps within the mesh by faking the lighting around it. Specular maps are the maps you use to define a surface's shininess and highlight colour.

Specular mapping helps to define the shininess of a particular mesh. These types of mapping are used to highlight areas within the texture making parts of the texture lighter than others. It does this by organising pixels from black to white, wherever there is more white the shinier the object will be in those areas.

2.4.3: Lighting

The next thing to consider is lighting, I should achieve certain lighting to again set the atmosphere. There are different kinds of lights that can be placed in the game that are already implemented in the game engine software. Point lights allow the designer to create a light source from a point, this could be a candle or a light bulb. These can be customized so you can decide how bright or how much of a radius these lights will have in your game. Spot lights are another light source that can be used, these beam light on just one spot and can also be adapted to how strong or weak you require them to be. This could be used to light up a necessary item or could be used on a torch asset so it can move and act the same way a torch would. Directional light also allows the designer to determine what direction the light is coming from. This can be useful as it will allow light to come in from certain windows and can brighten areas up within the game as it can represent natural sunlight.

2.5: Existing Games:

Games such as 'P. T' is a very good example of how horror games can be made. P.T (short for playable teaser) was released as a playable teaser for the upcoming game 'Silent Hill'. Players were able to download this from the PlayStation store, until unfortunately the game (Silent Hill) was cancelled. However, the game P.T demonstrated how a short horror game, can be just as powerful as a AAA title (AAA are games with the highest budget). The game consisted of the player walking down a corridor in someone's house and when the player reached the last door, then the player would then start back walking down the same corridor. Each time the player looped around there would be subtle changes within the corridor, which helped to explain the story behind the game environment. Eventually after a few loops around the corridor, things would start to get more intense, with violent banging and graphic images and scenes. This playable teaser was very good at building tension, as the subtle changes made the player expect a jump scare, however with it constantly being the same environment, they didn't know when to expect it. The corridor also consisted of one other door that was locked, as the player loops around, a baby can be heard screaming, this already makes the player very curious and cautious of what is in that room. Curiosity is important in the game as you want the player to feel fear but also have the urge to explore, so you can eventually catch them when they are off guard to create the biggest scare.

Another example of an existing horror game is 'Five nights at Freddie's.' In this game, the player is working on a nightshift and must watch the CCTV camera's in a diner called 'Freddie's' the idea of the game is to watch the cameras to see if anything changes and to use your limited amount of power to lock the doors, to stop the creepy looking teddies getting in and essentially killing you. The game is average as a horror game with 5 levels hence the title 5 nights at Freddie's, however the game becomes less scary once the player has lost once as the jump scare of when the teddy kills is always the same and becomes predictable. This takes away any tension and makes the player desensitised as it is very predictable. Having a repetitive scare for every level takes away the fear and tension the player has and therefore also reduce the enjoyment.

It is important that I make sure I use these techniques like P.T as this game was very well received and created a lot of support and excitement surrounding the game 'Silent Hill' that the company was promoting. When designing my own horror game, I want to avoid making the same mistakes that Five nights at Freddie's have previously made, as even though the game was popular it did not have the same effect or impact as many other horror games.

Chapter 3: Design

In this part of the documentation I will talk about the design process of how the game will look, including the level design, the structure of the game and the Menu's and GUI's. I will also be designing the HCI section of the game, such as the graphical user interface and the menus. The menu system will need to look appealing and offer the features that the user will need to use before they start.

3.1: Belial Hotel - Treatment Document

3.1.1: Abstract

Belial is a survival horror game, where the player will start off on the top floor of a hotel. Once the player wakes up, disturbed by a hysterical scream in the middle of the night they realise something has changed and the hotel is now derelict. The game will require the player to search around the abandoned rooms trying to find some notes and clues to help them escape the horror hotel.

3.1.2: Background

The game is featured in present day and the main protagonist is a character called Lewis.

Lewis is an ex police man, who was made redundant from the force which encouraged him to then pursue his goals of creating his own successful business. This has results in him traveling all around the world trying to settle various business deals.

Lewis's business has sent him to Mexico and his personal assistant has booked him into the Belial hotel. After searching the streets of Mexico to find the Belial hotel, he stumbles upon the hotel down a little side street. Upon arrival, everything seems perfectly fine, there are other characters checking into the hotel, everything seems perfectly normal. Lewis is given the key to his room and told he is on the top floor. He enters the room and thinks the building on this level seems a little outdated and feels there is an eerie vibe.

On the first night of Lewis's stay in the middle of the night he hears a scream, he sits bolt upright in bed and tries to decipher what direction it's coming from. After a while of silence he lays back down to be awoken half an hour later by another continuous and now hysterical scream. Lewis decides to go and see if any other guests can hear it in the corridor, and try give aid to the person screaming. He leaves his room only to hear his door slam shut behind him, realising he's left his key in his room, he knows he needs to get down stairs to the concierge. Lewis heads for the stairs, soon to find it is blocked off by the debris from a collapsed roof.

3.1.3: Game Genre

Belial hotel is mainly a horror genre game however it can also be considered to be a puzzle / mystery game. The player will have to search around looking for clues or items that will help to him to escape as he learns more about the hotels history. The game will work on building tension to create suspense and unpredictability to create more of a horror factor in the game.

3.1.4: Target Audience

Belial Hotel is targeted towards 16 years old+, the reason for this being that the game contains a lot of horror elements that are intended to scare the player. Therefore, the age restriction needed to be higher as it may contain unsuitable sights and images for a younger audience. Within this age group, the game will specifically target itself towards players that seek thrill and adrenaline throughout playing games. Therefore, this game will need to largely focus on creating a tense, on edge ambience for the player, so their thrill-seeking expectations can be met.

It could also be considered that the game may attract another target audience such as the players that enjoy solving puzzles and mysteries. Since the game will prove challenging to complete and has a sense of mystery that the player will uncover, this may influence other people to play.

3.1.5: Look and feel

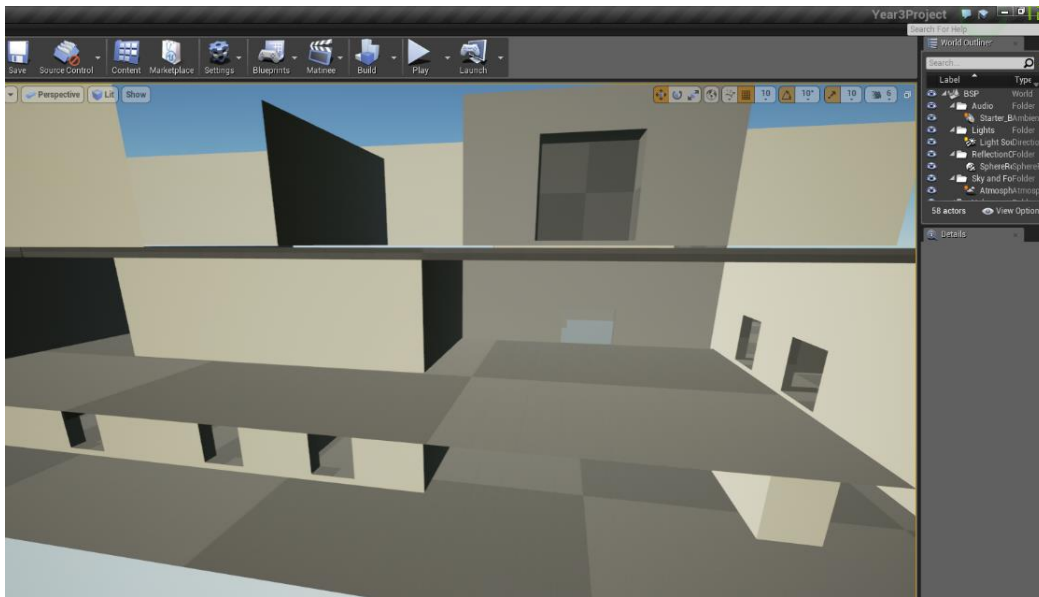


Figure 4 – A screenshot of the Unreal Engine Prototype BSP Stage.

The aesthetics of the game is very important. With the game being set in a hotel, it is important that it resembles a hotel very well. To achieve this, I will do extensive research into the designs of hotels and their structures until I find one that I can take inspiration from for my horror game. I will then create a BSP level of the map, this is basic geometry that will give the overall shape of the building and each floor the player will play through. Once the BSP level has been made I will then model assets in Maya, these can be anything from chandeliers to doors, that I will specifically make to create the desired effect. Once I have the BSP and assets, the next thing I need to consider that could be considered the most important is texturing. This allows the designer to apply texture to shapes / models to create a more realistic looking environment, this can be things that help create the authenticity of the hotel such as wallpaper or carpet, or things that add to the creepy ambience such as blood.

Texturing also helps to improve the feel of the game as well as the look. The look of the game is important as the player will need to recognise where they are, however the 'feel' of the game is what brings it to life. To do this, I will be adding various lighting. Lighting is important as the player needs to see where they are going, too much lighting will take away the fear and tension of the game, whereas too much darkness will make the game hard to play. To create a good balance, I will start off with a good amount of lighting so the user feels that they have a safe zone. However, as the game continues and the player ventures further into the game, the lighting will slowly start to reduce and diminish.

3.1.6: Game play

The gameplay will consist of a lot of interaction with objects, such as doors, keys and other useful items. The player will spend time searching for clues or specific items that will help them survive. This could be finding a key and then trying it on several doors until one opens. With the game being based on PC the normal keys 'WASD' will move Lewis around, the mouse will be used to look around and the 'E' key will be used to interact with objects.

The player will be in 1st person, this is important as it allows the player to look through Lewis's eyes. This is very important in a horror game as this will not only create a sense of involvement as the player feels as though they are Lewis experiencing all of the fear first hand but also restricts the players vision to tunnel vision camera angle, which creates further fear as the player won't be able to tell if anything is behind them.

3.1.7: Goals and Challenges

The main goal of the game is to escape the hotel safely and uncover some of the mysterious activity that has been going on. To achieve this the player will need to get passed some of the debris that may be blocking the way or find the key for the stair doors or even a switch to get the elevator working, every floor will hold a separate challenge that Lewis will need to conquer.

On the top floor the player will have to find the key to open the door to the stairway, the key will be located within one of the rooms hidden out of sight. This is designed to take the player a while to find, and the sound effects will be used to create tension and fear as Lewis looks around.

The next floor will involve using the elevator. To activate the elevator, Lewis needs to turn on a switch to start it up, this again will be in a difficult place to find. Once the switch has been found the player will take the elevator to the ground floor, but instead it will go to a basement room where an unusual figure, will see you in the distance. This figure will begin running towards you, the lift doors will then close right before they reach you and the lift will then take you up to the third floor. The figure will be aware of your location and begin to hunt for you.

Once the player has got to the next floor, there will be debris to prevent the player from just walking down the steps. To get to the next floor, the player will find a trap door that leads to a room beneath them. This room will be filled with dead corpses and can only be opened from the inside or with a key.

3.1.8: Audio

Audio is a very important aspect to the game, it helps to build tension, sets the mood and makes the game feel more realistic. It is important to add the right sound effects when developing a horror game, the sound effects helps to add to the eeriness. I will be implementing a footstep sound effect with a slight echo, this makes the character feel alone and vulnerable as well as doors creaking. I will also add a couple of scream effects as it will make the player believe that there is someone else in the building with them.

3.2: Flow diagram

This diagram shows the flow of the GUI; the user will start at the main menu screen and will have 3 options to choose from. The first being to start the game, the next for help and instructions and the

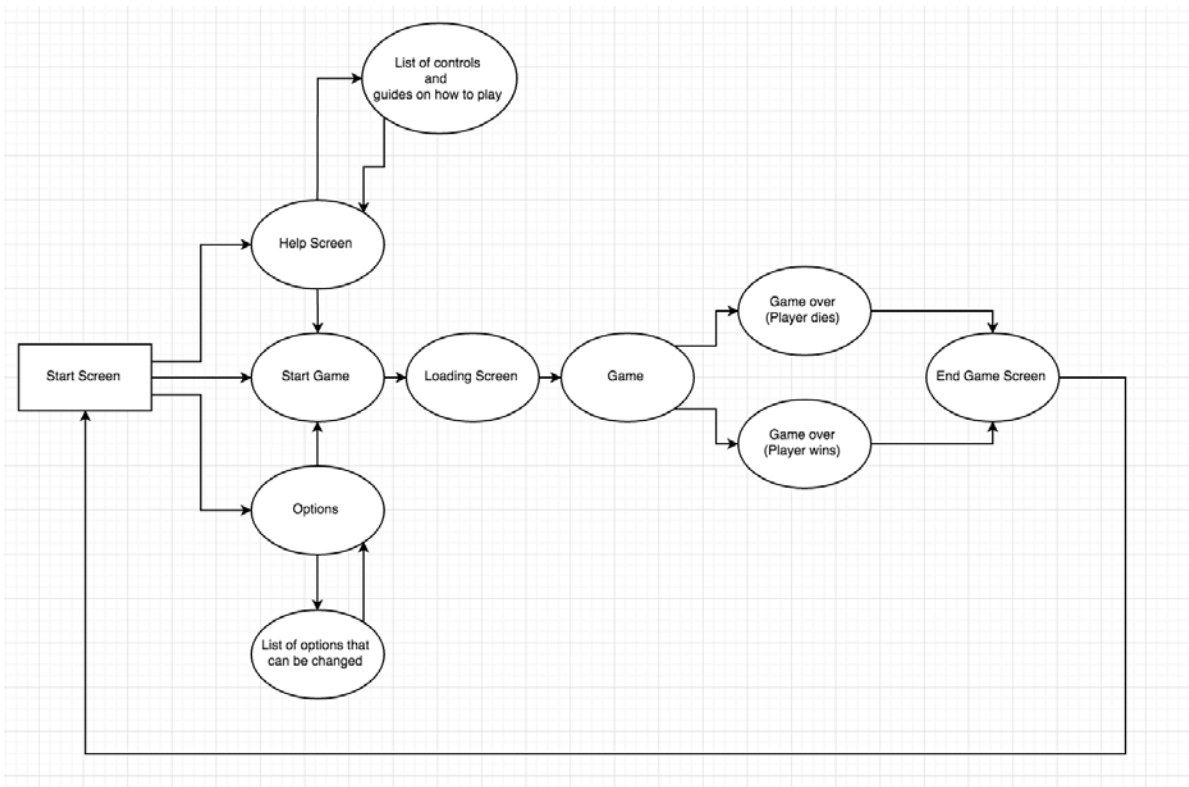


Figure 5 – A flow diagram of the structure of the game in development

last to change options about the game. Once in the game there is only 2 outcomes, either the player wins (escapes) or loses (dies) then menu will display a summary screen and then loop back to the beginning.

3.3: Main Menu (Start Screen) GUI

The main menu screen will have the hotel in the background, with a looping sound effect and will have the title of the game. It will also have the Play button, options and controls button as well as a quit button.

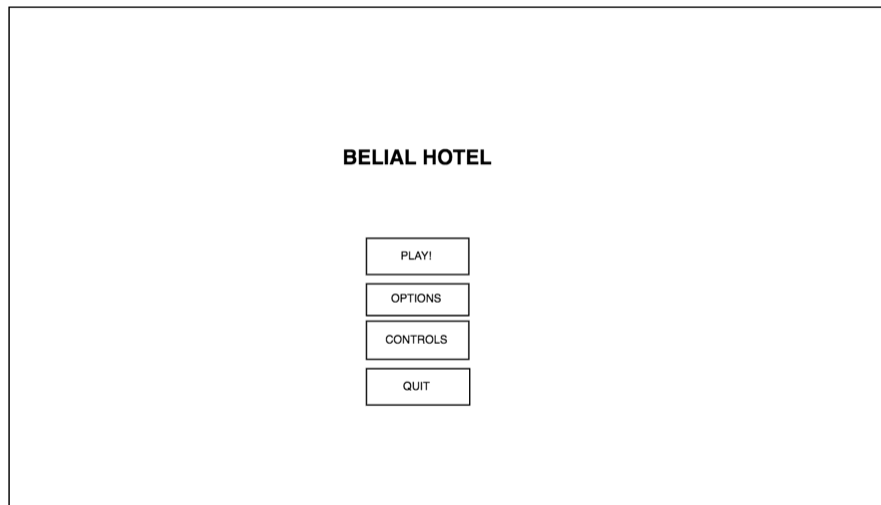


Figure 6 – Main Menu GUI (Design).

3.4: Options Menu GUI

The options menu will allow the user to change from a selection of different resolutions so the user can experience the game depending on the specification of their computer.

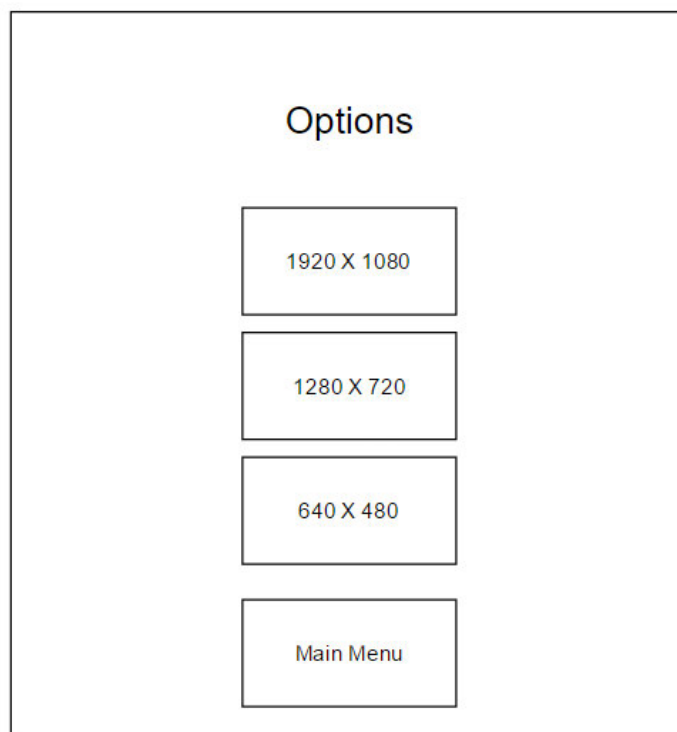


Figure 7 – Options Menu GUI (Design).

3.5: Help Screen GUI

The help screen again is almost identical to the options screen, however this screen will give a list of all the controls the player will need to know in order to play the game.

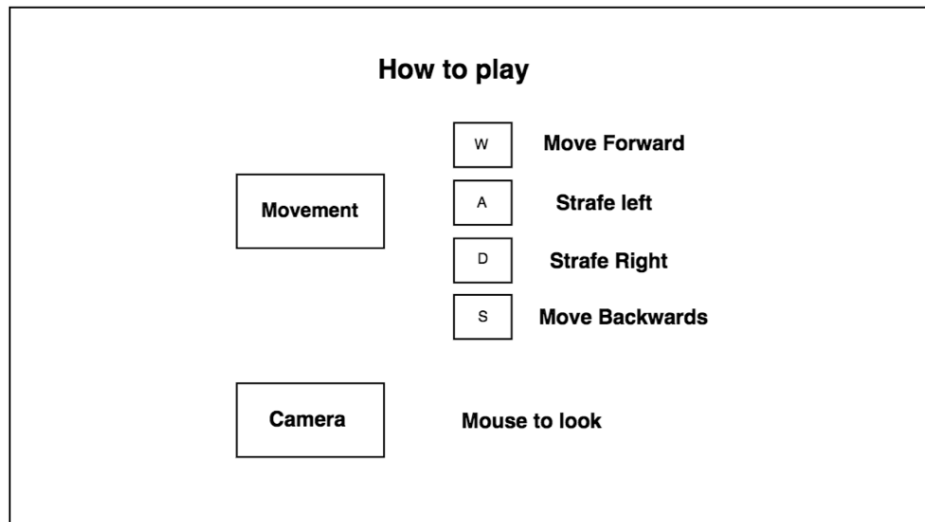


Figure 8– Controls Menu GUI (Design).

3.6: Gameplay Screen

This screen will be applied while in gameplay, this will overlap the current screen and will display the battery level, what items the player is holding and any clues they may have picked up.

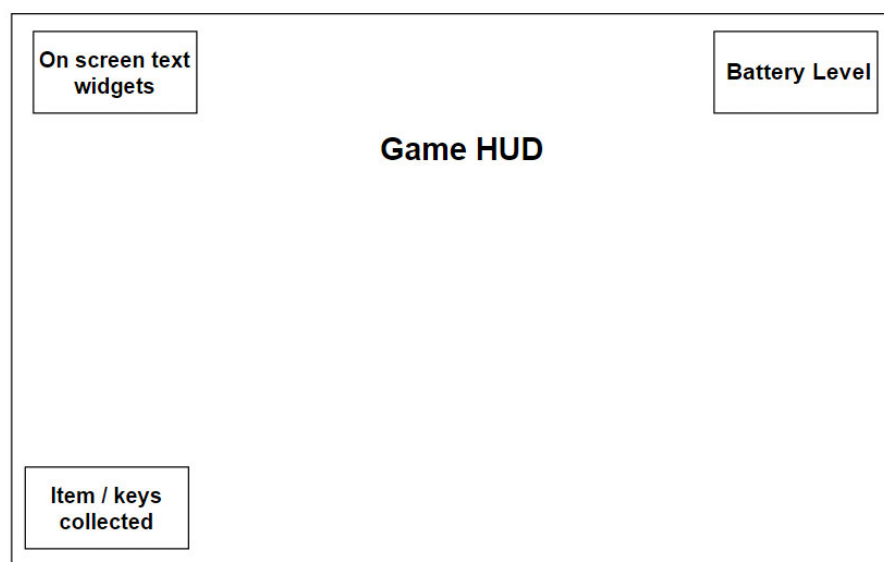


Figure 9 – In game display (Design).

3.7: Level design

When designing the game, I must think what the level will look like. This will be the basic structure off the hotel, with the rooms staying in roughly the same place for each floor but with the stairs and elevator not moving. As there will be a lot of rooms the events will take place in the hotel rooms.

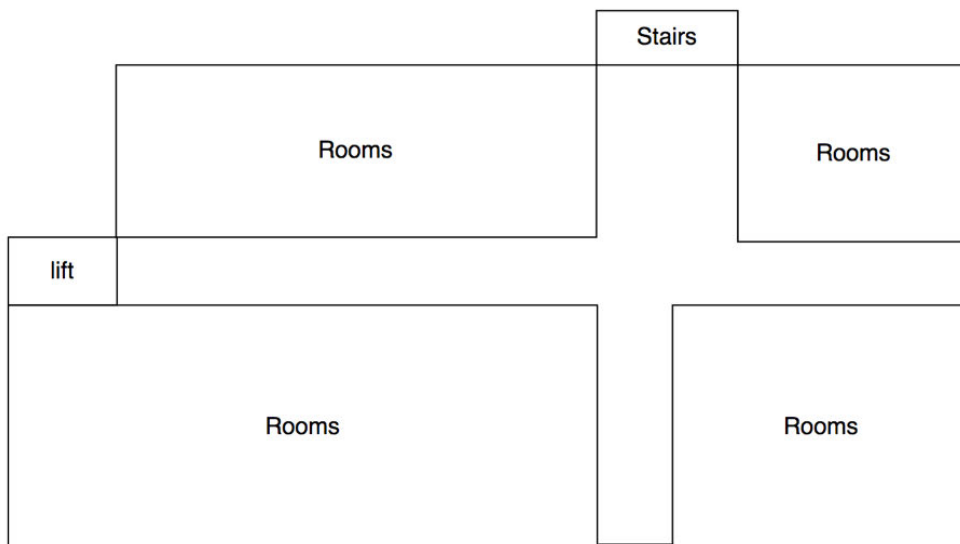


Figure 10 – General layout of hotel floor / Level design.

The scares that will be created within the game will be mostly sound effects these effects will require play throughs of the game, so I can execute the sound when the tension is high. There will be 4 different floors. For each floor the layout of floor will be slightly different and will be designed in a way to have sharp corners. This is because it will be easier to create a jump scare when a player walks around a corner, if there are multiple corners the player won't know when it's coming.

A room on each floor will require extensive searching as these will be the rooms that have key information in the form of notes and a way to escape. The notes will let the player know they are on the right track to escape.

3.8: UML

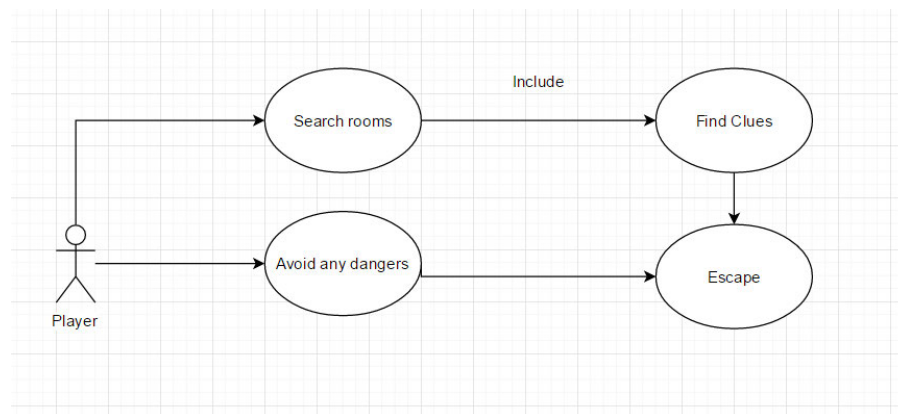


Figure 11 – UML design.

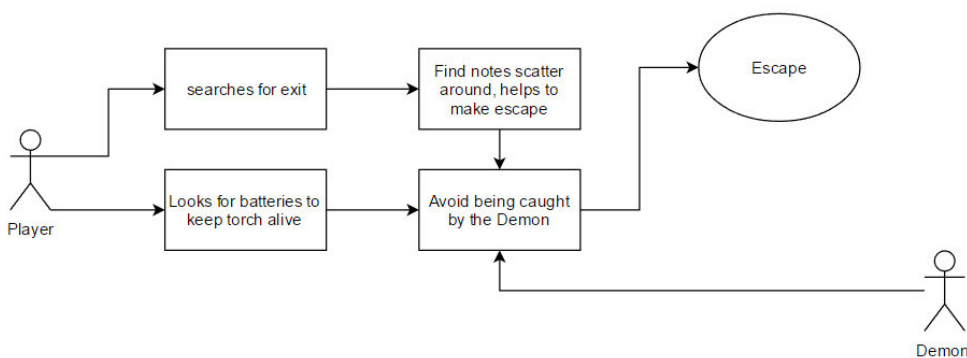


Figure 12 – UML design (2).

Chapter 4: Implementation

4.1: Unreal

The software I will be using to create the game will be Unreal Engine. First I will need to build the basic structure. To do this I will be using the Unreal Engine 4 software, this software allows the designer to place BSP (Binary Space Partitioning) shapes into the 3d world. The software allows the user to create basic shapes such as a square, circle, cylinder and a basic set of stairs. These can be scaled, positioned and rotated in any way, so these shapes are very useful to design the outline of the hotel. These objects are normally set as 'add' however, they can also set to 'subtract' this allows the user to place a transparent polygon into the 3d world that subtracts that shaped object out of another polygon. For example, a square polygon could be placed down then a subtract sphere can be placed inside it to create a whole.

To create models that will represent certain objects, there is an option in Unreal Engine this allows the designer to model the object further than just resizing the shape. The object can be manipulated by the edges and by the faces so they can be replaced or dragged out to completely reshape the object. Extrusion works in a similar way to just dragging the face expect when a face is selected the object then can be extruded this allows the designer to create an entirely new cube from that face which can be fully edited also. This allows for a very wide creation of objects and will be my main way for designing specific objects.

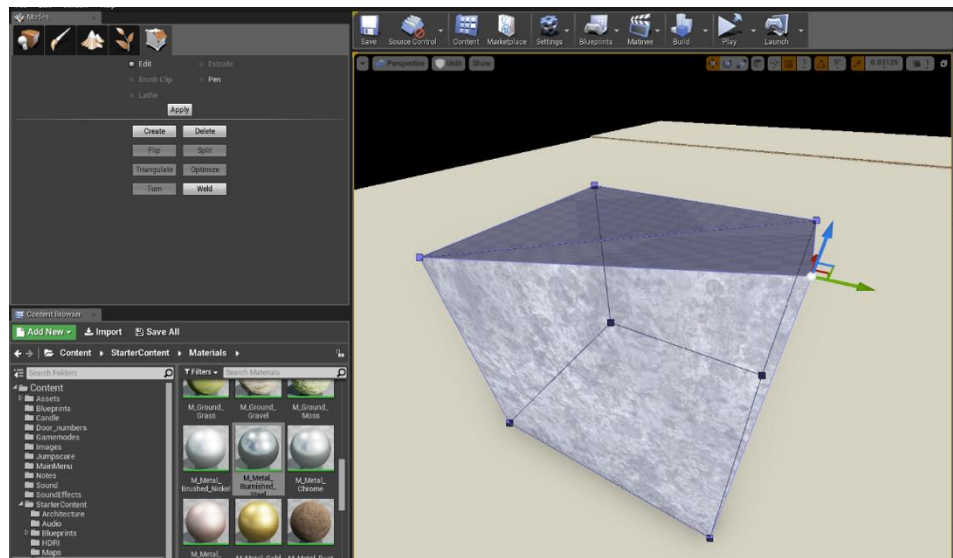


Figure 13 – Screenshot from Unreal Engine BSP Editing

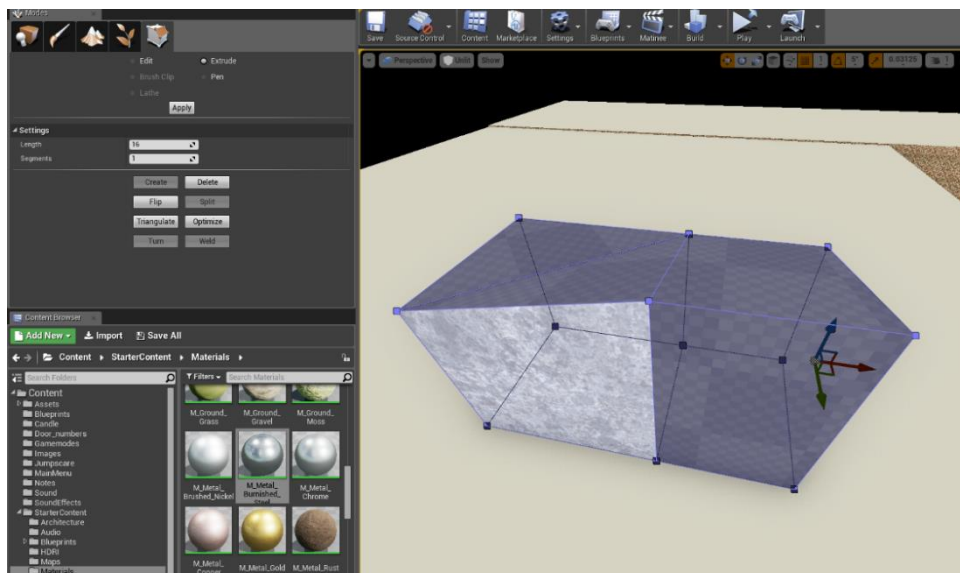


Figure 14 – Screenshot from Unreal Engine BSP Editing (2).

4.2: Importing / Exporting and Asset creation

When designing Belial Hotel, it is important to fill the levels with assets. Assets are just components that you add to a project in order to give it characteristic this could be anything from a door to a sound cue and allow the player to interact with objects. Using Unreal Assets can be created within the engine, for example you would use basic geometry to form a shape then convert it to a static mesh, then a blueprint if you want to give it its own set of instructions. However, creating assets can be a lot more effective when using modelling software such as Maya, when using this software, it is also important to learn what formats can be exported by the modelling software and what formats the Games engine will support. Unreal supports the file type .FBX, this means I can only import assets by this file type. This means when creating assets, I will have to use software that is able to export into this format. Maya can export in this format and is very good as a modelling software, this will be my go to choice when creating any assets, I need.

When creating assets for the game, I was mainly using the Unreal Engine to design most of the models. However, I realised that creating something such as a door would mean using another type of programme. The reason for this is because when animating a door, it needs to rotate around the bottom left or right pivot. While it still is possible to move the pivot in unreal it does not stick therefore the pivot is always centred in the middle of the object. This meant it was important to create these kinds of assets within Maya. When exporting from Maya it is the centre (0,0,0) point when the pivot will be placed this meant that I could create a door, position the corner of the door on the (0,0,0) point and when exporting the model the pivot would be in the place I needed it to be.

4.3: GUI

When designing the Menus, the first procedure was to create a new empty level. The reason for this is so that the game is not already running when the menus load. I then created a main menu widget that is displayed once this level first starts. The Play button loads the main level when clicked. The other buttons set the main buttons to invisible and then selected the new buttons they are referring to, to visible. For the colours and theme of the buttons, the main menu had to match the horror theme therefore a dark red for the title allows it to stand out against the background, the background is meant to represent the hotel itself.

Main menu / Title screen.

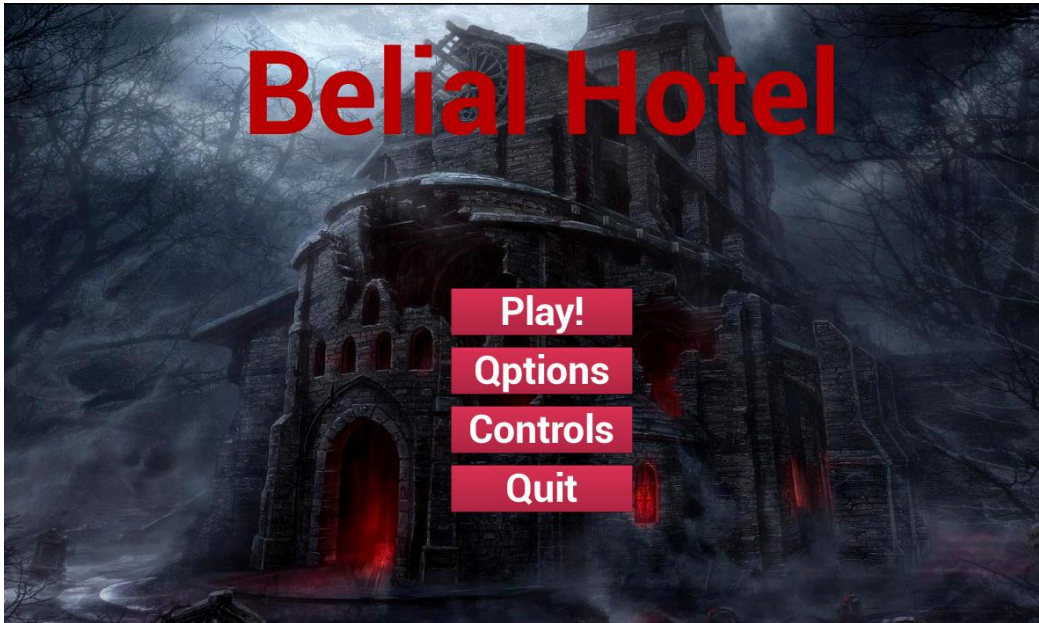


Figure 15 – Fully implemented Main menu.

This is the Options menu, this menu allows the player to change the resolution of the game, this will allow the user to have a better experience playing the game depending on their computer specifications.



Figure 16 – Fully implemented Options menu.

This is the controls menu, this menu has all the the user may need when playing the game and has a link back to the main menu.

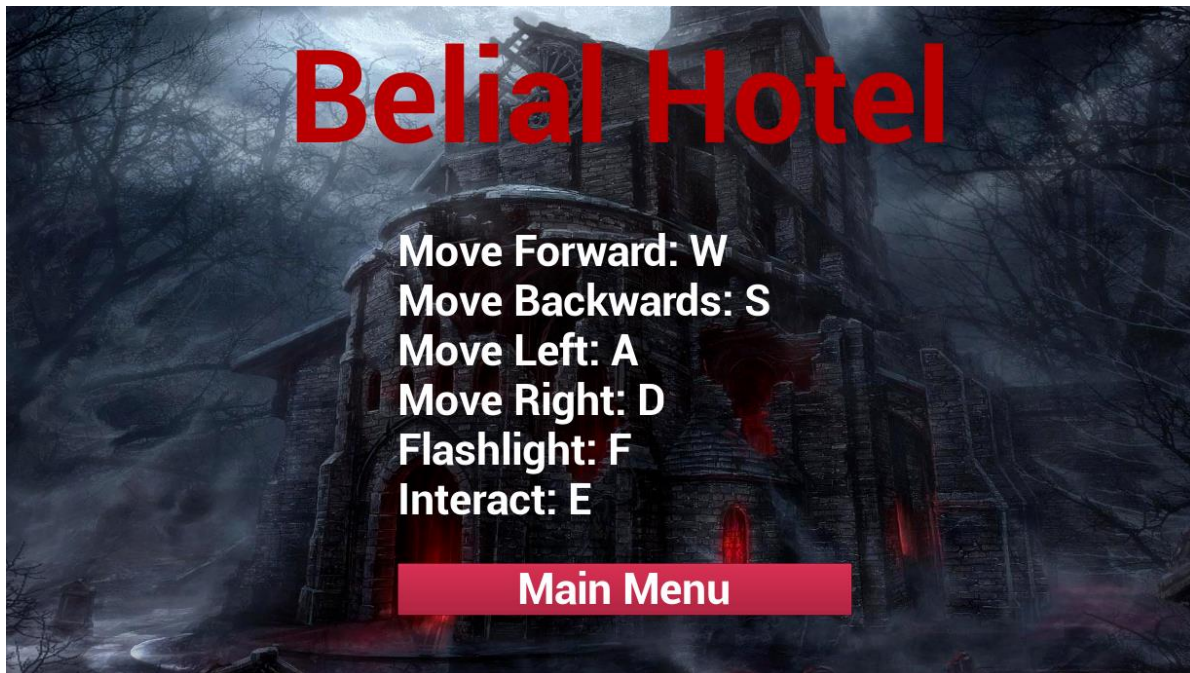


Figure 17 – Fully implemented Controls menu.

Default display the player will see when playing, the battery in the top right is a progress bar with the value of 100, it will decrease by 1 every second until the player finds a new battery that will refill the progress bar back to 100. If the player runs out of battery they have 8 seconds to find a new battery else the it will be game over.



Figure 18 – A gameplay Screenshot within the prototype.

Notes GUI.



Figure 19 – A gameplay screenshot of note widget.

When the user sees a note, they will be able to interact with it by pressing the E key. It will then display a texture on screen with text printed on it. These are used as clues to help the player find their way around and can help to add tension by so the player does not think they are alone.

Key GUI



Figure 20 – A gameplay screenshot of widget when key is collected.

When the player finds, a key text is printed on the screen for a second stating where this key will unlock. It also displays a key icon in the bottom left in case the user forgets they have collected a key or did not see the text displayed. It will also be removed when the player opens the corresponding door.

Game over screen

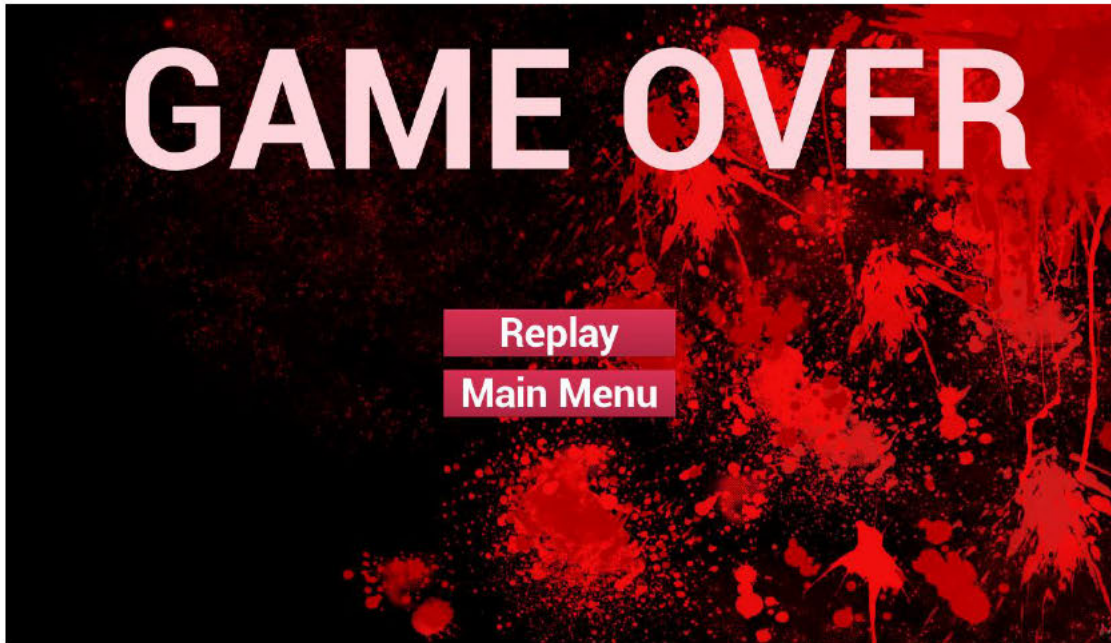


Figure 21– Fully implemented game over end screen

This will be displayed if the player runs out of battery or is caught by the demon within the hotel. It allows the player to return to the main menu or start the game again.

The victory screen. This will be displayed if the player managed to find the exit.

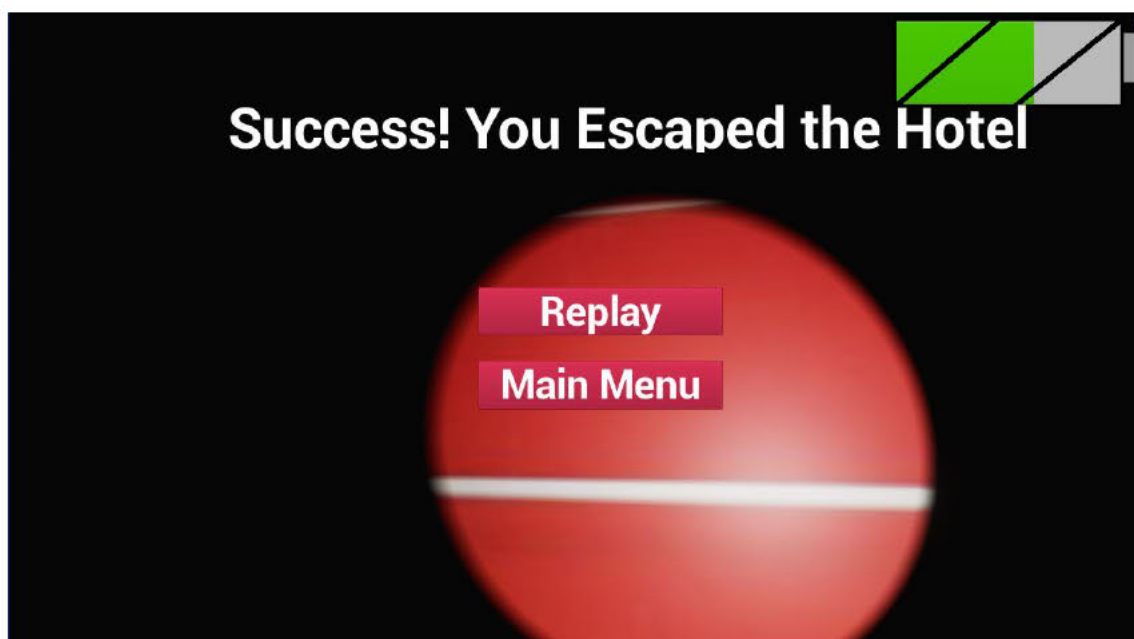


Figure 22 – Fully implemented player wins screen.

4.4: Blueprints

The unreal engine blueprints is a way of scripting but made a bit simpler as they are visible and easier to work out what node is connected to what, they are representing the code C++. Blueprints work by starting an event, this could be a collision with a box trigger of a key being pressed. That

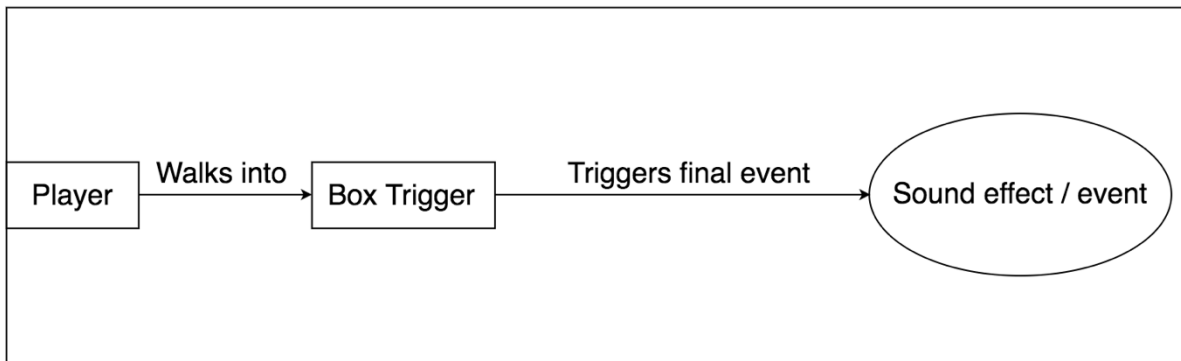


Figure 23 – Unreal blueprint concept design.

event will then be attached to a specific outcome. For Example, if a character was to pick up a key there would be a box trigger attached to that key mesh. Once the actor walks into that box and presses the key for the interaction it would destroy the key mesh and add the key to a variable stating the player has picked up the key. For this to work a gate will need to be created a gate is entered if the certain key is pressed however will not open unless the actor is overlapping that keys box trigger. Once the gate is exited we can set the variable to true for the key and then reference that key actor to be destroyed. We can then refer to that key variable if there was a lock on a certain door that the key could open. The same process would be made but this time a branch would be added to act as an IF statement if the key is true the door will open.

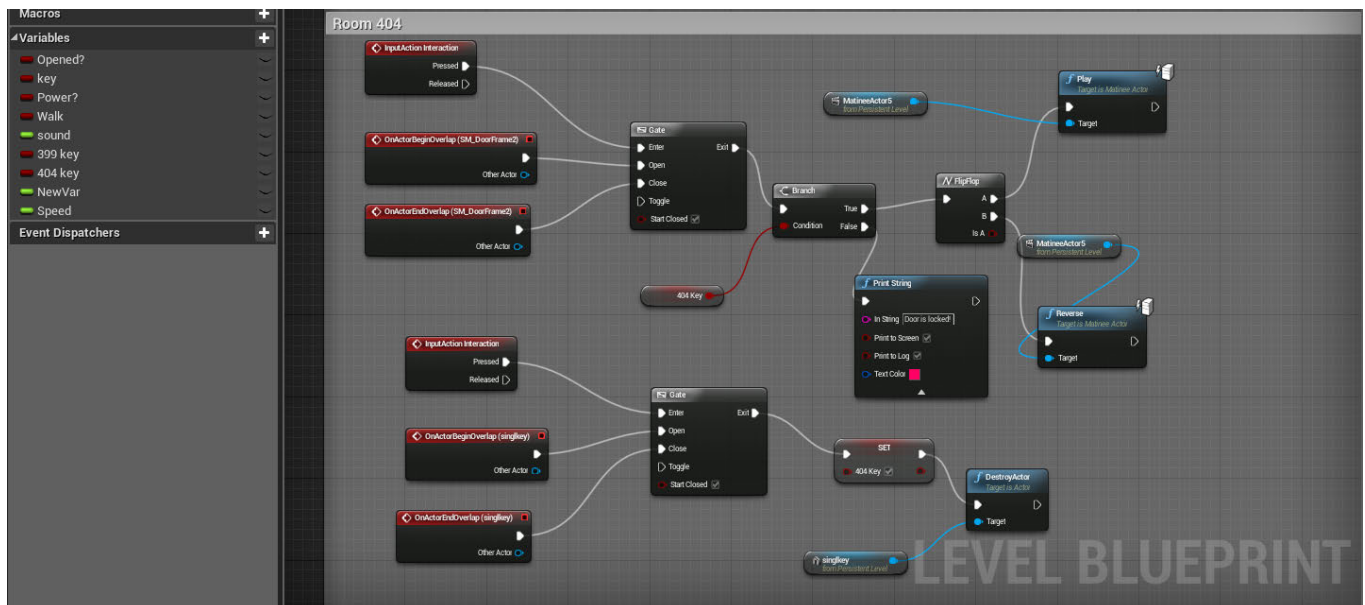


Figure 24 – Screenshot of unlocking door blueprint code.

This is the blueprint for collecting the batteries within the game. In order for the player to collect the batteries, it will require the player to press the interaction button when they are close to the a battery. The battery within the level and the one the blueprint is referring to, has its own blueprint this means you can attach different objects to it, such as a box collider and give it its own set of commands. Once the player enters the box collider attached to the battery mesh and presses the interaction button, it will then enter and open the gate. On exiting the gate it will destroy the actor and then I refence the battery blueprint to destroy.

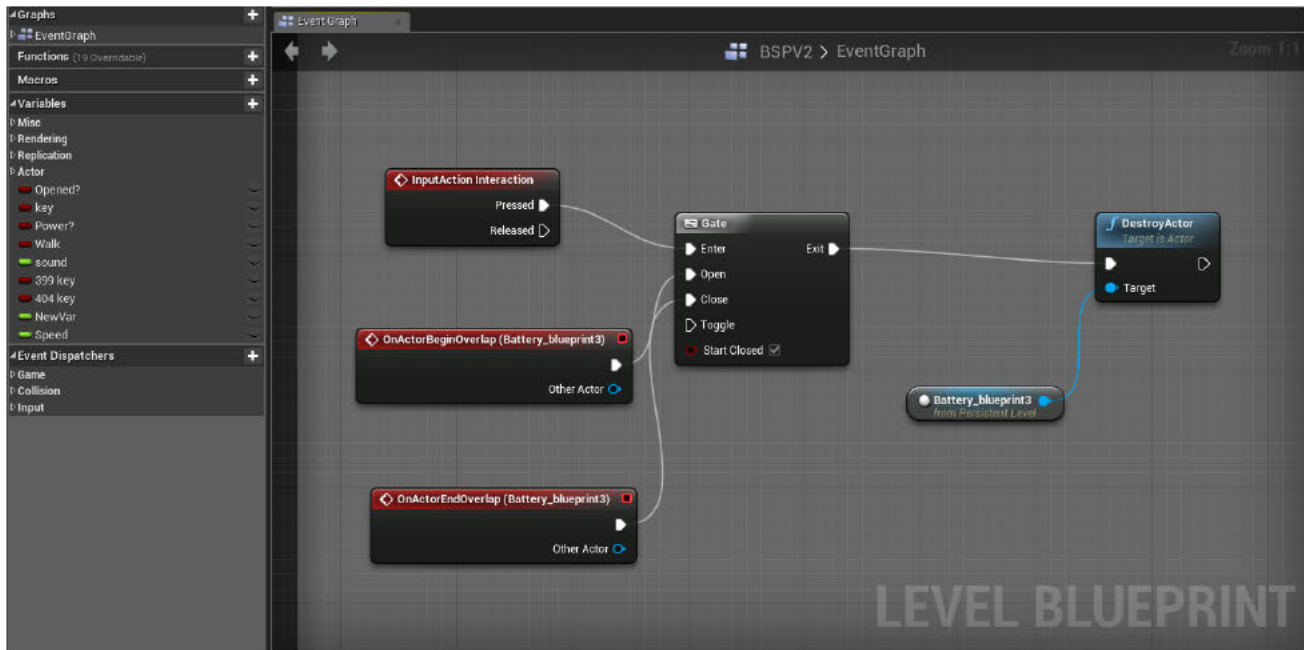


Figure 25– Screenshot of picking up a battery blueprint (within level blueprint).

In the battery blueprint I also had a set of instructions that in the event the actor is destroyed, it will set the battery power back to 100. In order to do this I have to cast to the 'Lewis' blueprint. This allows me to talk to the variables inside the Lewis blueprint and change their values.

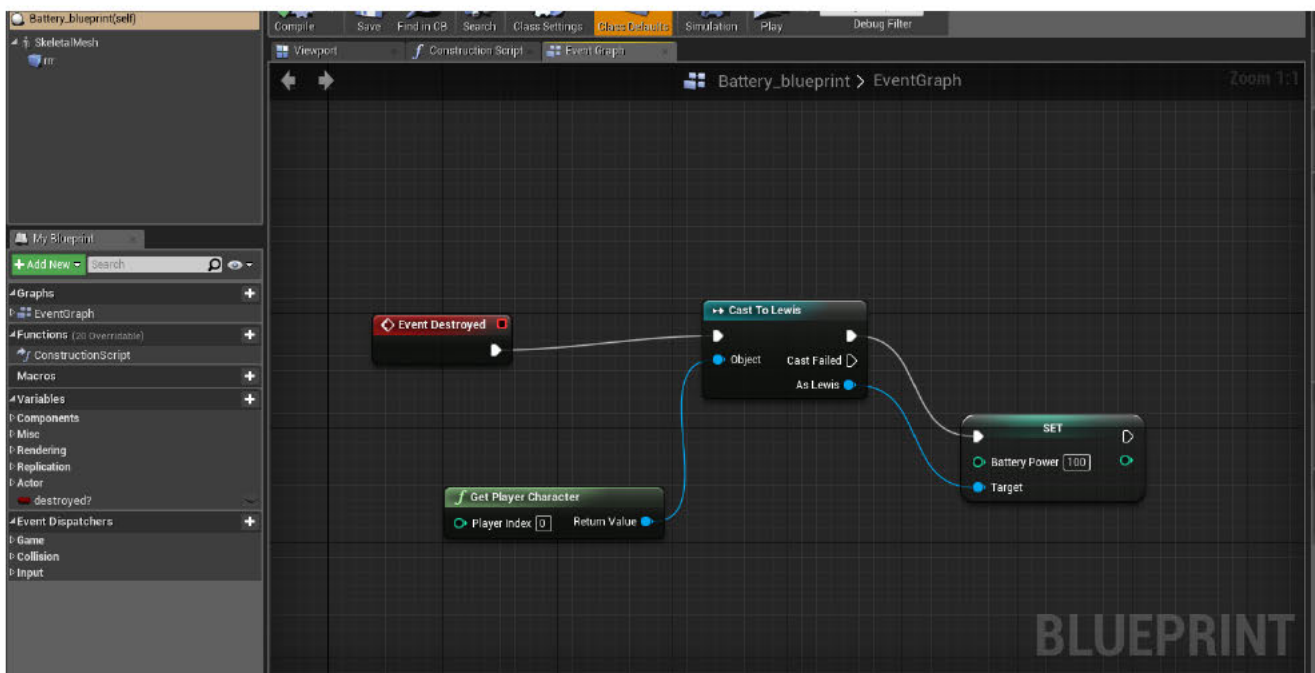


Figure 26 – Screenshot of blueprints within the battery blueprint.

Within the Lewis blueprint I start by using an event tick, which checks a condition continuously every second. Attached to the event tick is a branch. A branch is like an IF statement, if a certain condition is true it will do one thing if not do another. In this situation, the first branch checks whether the player is dead or not, if true it displays the game over screen by creating a widget and adding it to the view port. If the variable dead is false, then leads to then next branch. This branch checks if the torch is on, if it is false it will lead to a delay which is set to 8 seconds, this allows the player chance

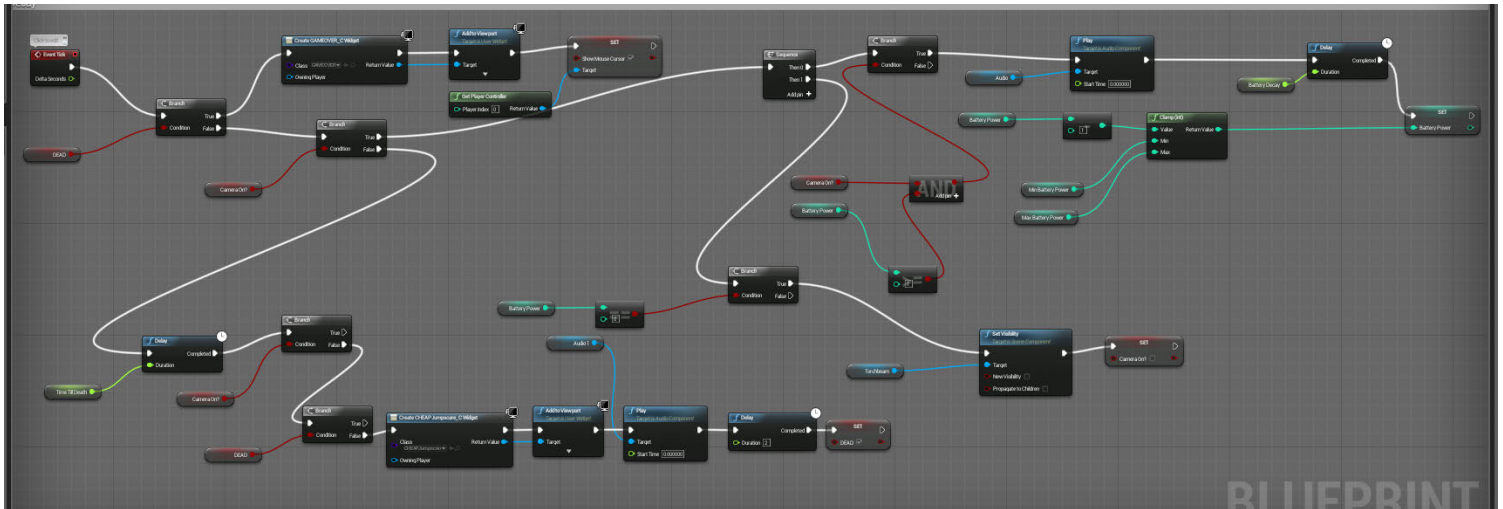


Figure 27 – Screenshot of battery power decreasing within the Lewis blueprint.

to find a battery, if after the 8 seconds the torch is still off it sets the dead variable to true.

However, if the torch is on then this leads to a sequence. A sequence does one thing first then runs the event in that sequence. The first sequence leads to a branch, this branch checks whether the torch is on and have the value greater than one. If this is true, then a delay is started with the value battery decay. The battery decay is how fast the battery power value will go from 100 to 0. From the decay leads to the battery power variable but this time I set the value, to set the value, I will use the clamp node which allows me to enter 3 values the first being the battery power minus 1, in the value section of the node and two other values, the battery max and min values, I make these 0 and 100. In the next part of the sequence it will check using a branch node if the camera is equal to 0 if that is true I will reference the torch beam light and set the visibility to off and then set the torch variable to off also.

Also, within the Lewis blueprint, the F key will toggle the flashlight on/off. I use the event begin play

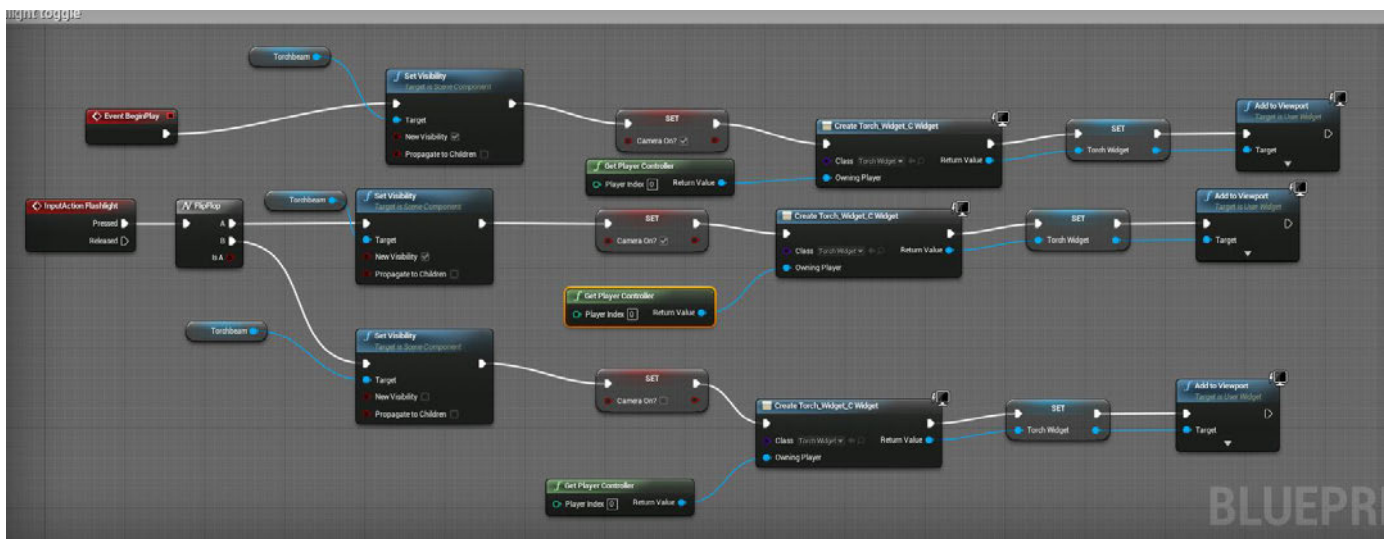


Figure 28 – Screenshot of torch's blueprints within the Lewis blueprint.

function so that when the level first starts it sets the visibility of the torch on and sets the torch variable to on. It then creates the widget of the progress bar to the viewport. I can then create an event for when the flashlight button is pressed and create a flipflop so that when the button is pressed it will alternate between the two functions which are the same as the event begin play but, one is sets the torch variable to off.

In order to get the progress bar within the torch widget to communicate with the battery values I will have to set up a blueprint within the widget. The first thing to do is to bind the progress bar this create a node which is used to get a value for how full the progress bar will be full. I will then get the Lewis node which will allow me to grab the battery power and max battery power. From this I will need to create a divide node and convert them values from intger to a float to make it easier to divide. I will then attach the divide node to return value in the progress bar nodes.

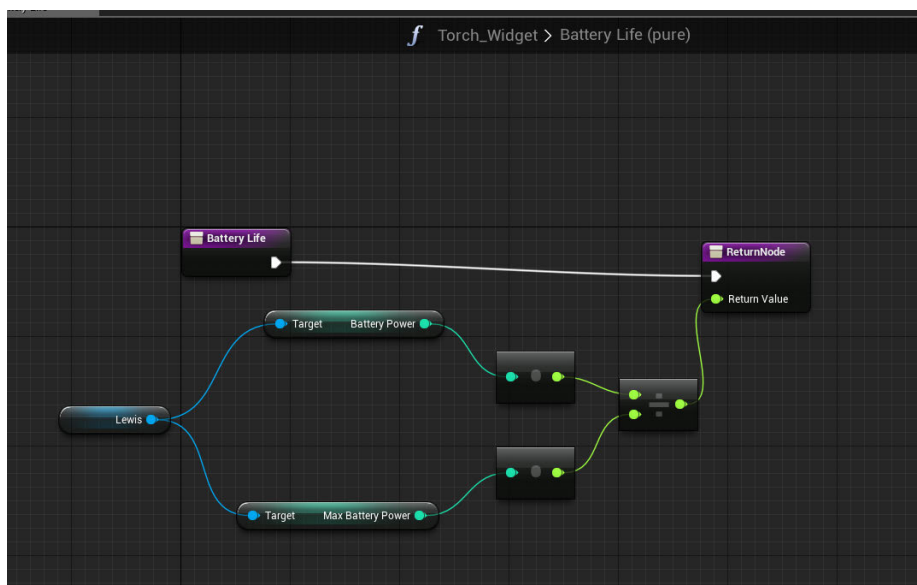


Figure 29 – Screenshot of progress bar widget blueprint.

4.5: Characters

During the implementation stage I needed to design the main protagonist 'Lewis'. Since the game is first person I could use an empty collider and attach a torch to it. By positioning the first-person camera just behind the torch, I could create the illusion you were looking through the person's eyes whilst holding the torch. To make it even more realistic I created a footstep sound cue, so I could create a footstep sound every time the player moved forward.

4.6: Matinee

For the horror game, I am designing, I will animate meshes so the player can interact with objects, the best way to achieve this is to create a matinee. A matinee allows the designer to create

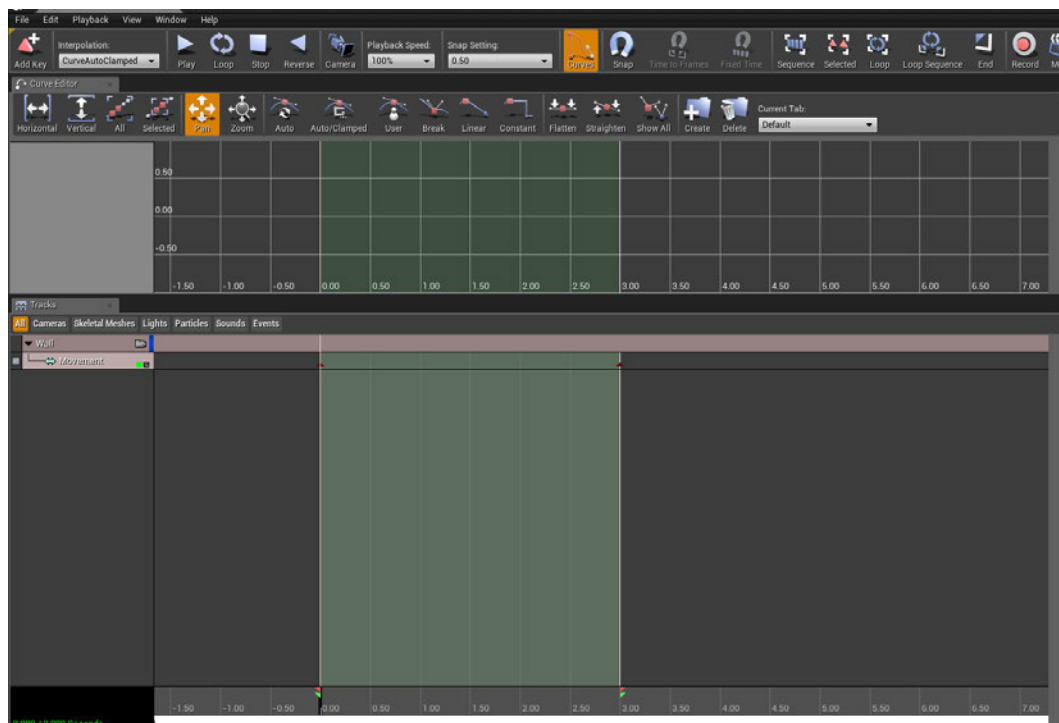


Figure 30 – Screenshot inside the matinee editor.



Figure 32 – Matinee Icon

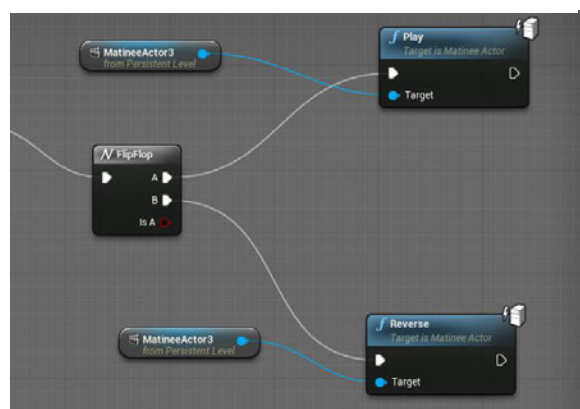


Figure 31 – Screenshot referencing matinee within blueprint

an empty group, in this group you can add the meshes that you wish to animate from there you can add tracks. Tracks can affect the mesh in different ways for example, I will add a movement track which tells the engine that the mesh will be moving. I will then take a key frame of the start position and then a key frame of the object in a different position further down the timeline depending on how fast the object will move. Once the Matinee has been created it will have an icon within the editor. This icon can now be referenced in the level blueprint, once referenced it can be attached to the play node, which will play the matinee referenced to it, it can also be attached to a reverse node which will play the animation backwards. This is very useful as it allows for opening and closing door or drawers and only needing one animation.

Chapter 5: Testing

In this section I will be testing the prototype that I have created and get feedback from play testers, by which I will be able to find out if the game is scary, but also meets the requirements on what makes a good horror game and avoids any mistakes. The game will also be tested through the engine to check for the frame rate and overall performance the game is outputting. This is important as the frame rate will need to be over at least 30 FPS (Frames per second) to make sure the game is playable and there is no lag or delay.

I will conduct a survey that will ask the participants questions that will help give me the required feedback that's needed to improve the necessary parts of the game. It is important to learn what could be improved from the feedback as it is important the game functions and plays well however, the game must make sure it builds tension and sound cues and animations are used at the correct times so they are not predictable and make for a good horror game.

5.1: Protocol

I will start by letting the participants play the game, once they have finished playing I will ask them several questions about their experience and then gather feedback from the results. Once I have gathered the feedback I will check to see if there is a common correlation between answers.

5.2: Questions

- Did the game have a good atmosphere for the horror genre?
- Were there any predictable scares?
- Did the game make you feel disempowered and vulnerable?
- Was there a constant build-up of tension?

The tests will be carried out all on the same computer so the tests are all fair and each participant will be observed whilst playing, to gain information on the way they play and the reactions they had to any scares.

5.3: Results

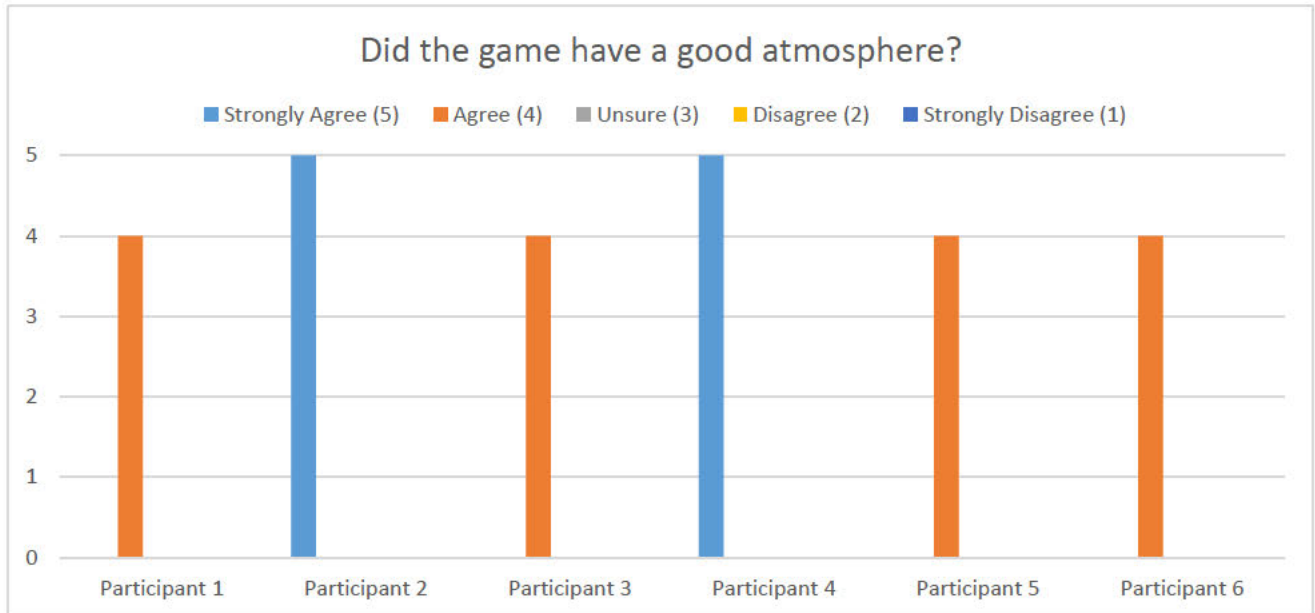


Figure 33 – Results from “Did the game have a good atmosphere?” question

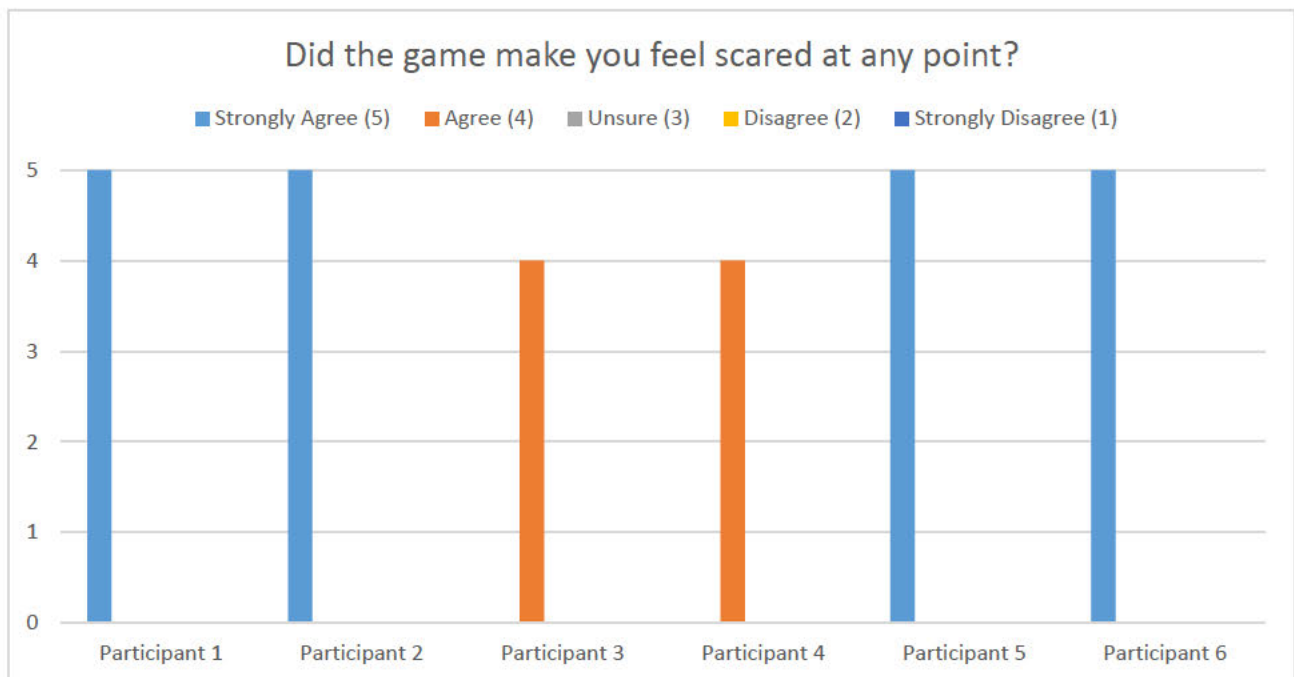


Figure 34 – Results from “Did the game make you feel scared at any point?” question

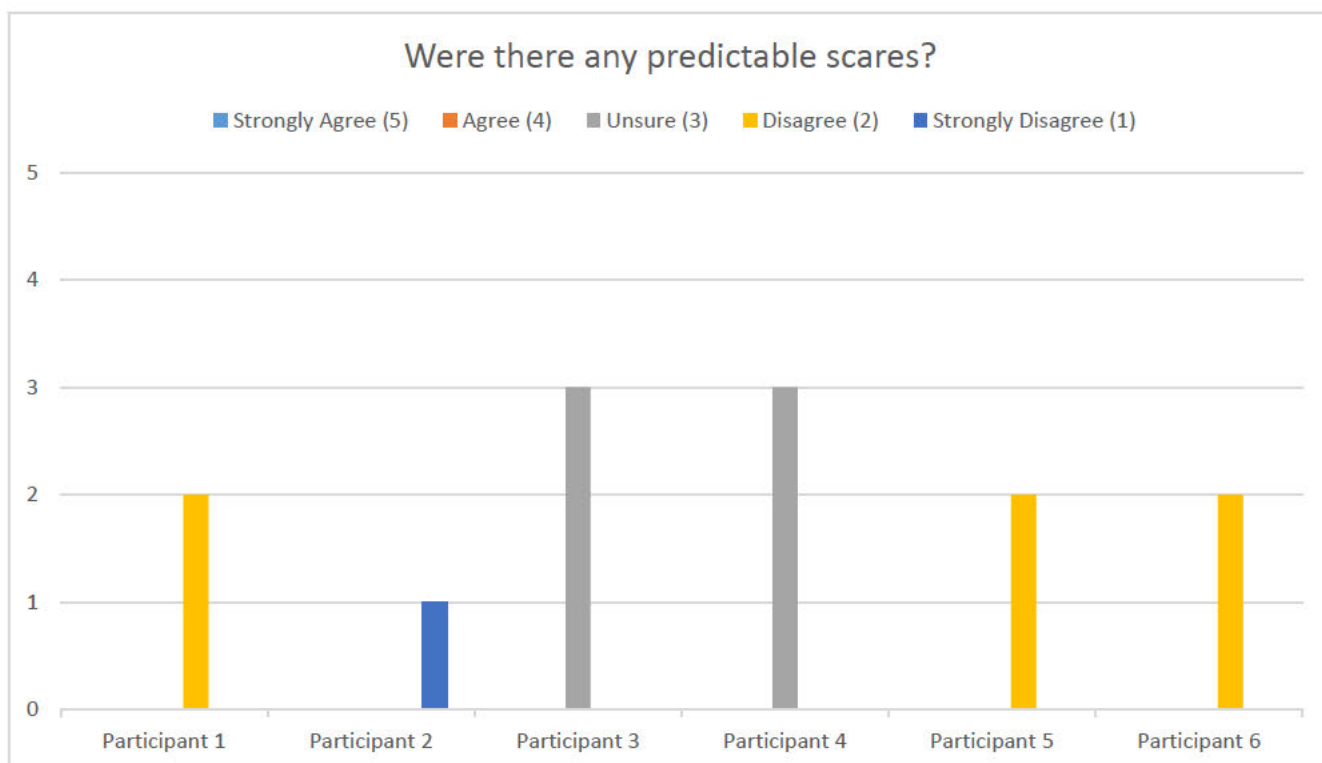


Figure 35 - Results from "Were there any predictable scares?" question

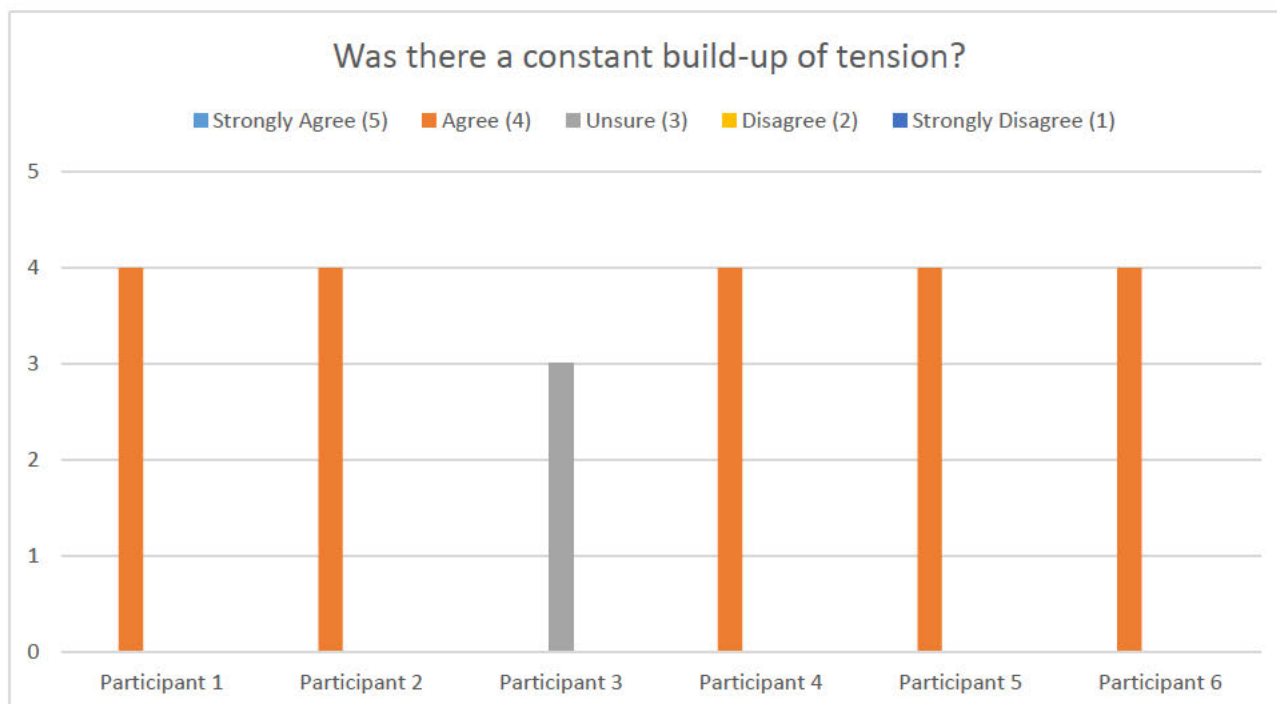


Figure 36 – Results from "Was there a constant build-up of tension?" question

5.4: Feedback question

Participant 1:

"The game was good; however, I would have liked to have seen more variety in each room."

Participant 2:

"I enjoyed the game I loved the fact you needed to collect batteries in order to survive, I would have liked to have seen more history about the hotel and why you are there..."

Participant 3:

"Every room was nearly identical; I think I would have really improved the game if there was a bit of a difference and more items to interact with."

Participant 4:

"I really enjoyed Belial hotel the game had the right level of eeriness and tension, when the scares happened they caught me really off guard!"

Participant 5:

"The game was a little short, however I liked the fact the corridors were so dark that you couldn't see till the very end, it had me on edge every time I was walking around."

Participant 6:

"I would have liked to have seen more mystery in more rooms of each floor, each room could of unravelled more of the story and the Belial demon"

On each test, I would also run the build in console command STAT UNIT and STAT FPS. This would display a list of information such as the frame rate which would allow me to take an average reading of 62 FPS which is more than enough frames to play a lag free game.

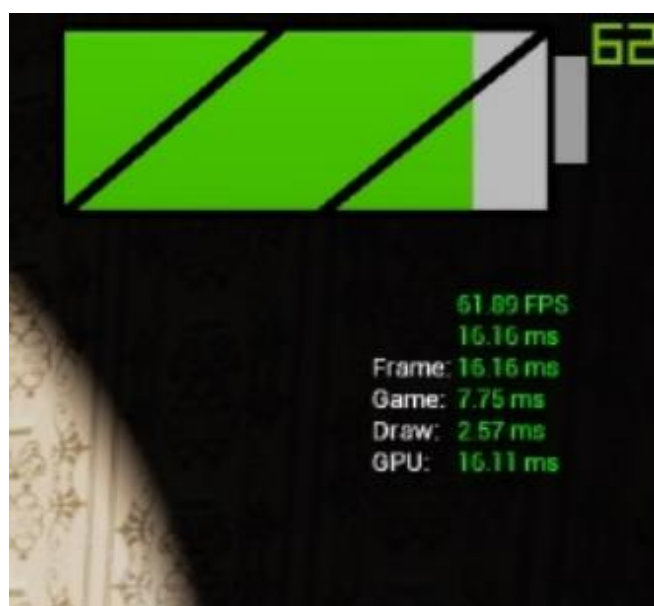


Figure 37 – Screenshot of stat unit console command.

Chapter 6: Evaluation

From the testing results, I have managed to gather a good amount of data. From the testing, I have been able to understand which parts of my game were strong and the places where there could have been room for improvement. The first question I asked was if the game has a good atmosphere, from my research I have found that creating the right atmosphere is crucial when designing a horror game as it's vital in setting the mood for the player. All the participants scored this question with very good and positive results. This is very good news as it means I have reached and succeeded in achieving one of the aims I set myself at the beginning of the project.

The second question asked if any of the scares they experienced such as sound effects or animations were predictable. From the bar chart, I can learn that 4 out of the 6 participants disagreed with this question meaning that the scares were not predictable with is another very important feature that horror games need to avoid, for the game not to be boring and not scary. However, 2 of the 6 also stated they were unsure about whether they were predictable, I feel as though this could be because when the player runs out of battery a scare will appear, this may seem repetitive, however I believe this was necessary as it gave the player more of an incentive to collect more batteries and search every room.

The third question asked if the game scared them at any point, this is one of the most important questions I had as the ultimate goal for a horror game is to scare its players. The results were all extremely positive for this question as almost all the participants gave 'strongly agree' with only 2 giving this question 'Agree'.

The fourth question asked whether there was a constant build-up of tension. Tension was another key factor I found to be important when developing horror games. I knew how important it was to have lots of tension so the player could be startled by the effective jump scares. This question also scored very high with 5 out of 6 agreeing with this question. I believe this question scored high because of the need to find batteries for the torch. Trying to find them before the battery dies out add to a lot of tension.

At the end of each questionnaire there was a feedback question so the participant could state what they enjoyed the most or if there were any improvements that could be made. From the feedback collected it seemed that the only suggestions that were made were to add more interactive objects within the rooms. This was pointed out as the testers found that it could potentially help to tell the story of this mysterious Belial hotel. It was also stated that each room was very similar, so by adding different assets and materials this would have created a more unpredictable feel. I believe that this feedback is very useful as it seems that even though the game is scary, there are also improvements I could make to further advance my game, giving the hotel more character and therefore creating more tension and mystery.

6.1: Evaluation of aims and objectives

In the beginning, I set a few aims and objectives that this prototype should meet. I believe that the prototype has met these aims. From the information gathered in the testing stage it is proven that

the game does create a scare when playing it for the first time. The game also builds tension with having to collect batteries, this was one of my most important aims.

It was also important that the game had good audio as audio is key to create all the scares and dramatic effects. Again, from the testing data we can see that everyone who tested the game stated that there was good atmosphere when playing the game. This was achieved by creating a good looping sound track that plays in the background. Alongside sound, lighting was another big factor that I had to research and make sure the lighting was perfect to set the scene. Using nearly complete darkness with just a torch to see where the player was going was perfect to create the horror atmosphere, this allowed the player only to see what was in front of them, this made the player feel a lot more vulnerable when checking certain areas of a room and behind themselves.

Chapter 7: Conclusion

To conclude, I feel very happy with how the prototype combined with my research has turned out. I believe the functionality of the game works well as well as the lighting sound and atmosphere. The goal was to create a horror game that build up tension and did not have any predictable scares, which from my testing research I have been successful.

However, I did face some problems when creating and designing the horror game. When programming using blueprints, it was very difficult to make sure everything behaved how I wanted it to, I had problems when coding the battery as the battery life would drain even if the camera was off, however I could print the values onto the screen to visually seen when the problem was. I also had a problem adding the demon into the seen, using its animations and trying to make it able to follow the player proved difficult, however which a lot of research I could overcome this problem.

Appendix

1.1: Terms of Reference

1.2: Course specific learning outcomes

My course has its own set of specific learning outcomes that is expected of me to meet when completing my project and passing my degree:

- To study the history of computer games, game genres, game structures and game design principles and to use the skills acquired to specify and evaluate new game applications;
- To gain an understanding of the creative practices involved within the field of games design and development;
- To become knowledgeable in identifying and implementing emergent communication and entertainment technologies and to be able to apply this to the design and development of interactive systems;
- To learn structured approaches to computer programming and multimedia and web computing;
- To provide and understanding of the relationships between concept, design and development within the context of theoretical approaches to games design;
- To gain an understanding of the multidisciplinary environment in which commercial games are designed and produced and to acquire skills in project management and team working.

My project will meet a few of these aims. such as the first point, I will have to study the history of computer games and game genres as I will need to get a complete understanding of what classes as a horror game, what is expected from a horror game and to see how other developers design and programme their games.

I will also make the game as creative and original as possible as well as taking inspirations form games I have researched and played. This is important as there have been so many horror games it will be important to stand out and offer something different. By doing this I will also meet the second learning objective.

1.3: Project Background

Horror games have always been my favourite genre type game. They rely heavily on building tensions and a mysterious plot so that the user is fully engaged within the game in order to get the maximum scare. I have played many horror games in the past, 'Outlast' a horror game based on exploring a mental asylum and filming everything inside for evidence, the only means of defence is to hide from whatever is inside. I have taken inspiration from this game, as I believe the idea of just being able to hide really adds to the scare feel. Other games such as 'Amnesia the dark decent' also match this type of horror game but adding puzzles that the user needs to solve in order to progress further in the game. My game will be based in a hotel where the player which wake up to screams and will leave there room in the hotel and will have to work their way out, without getting caught by whatever is lurking inside, by solving solutions to puzzles on each floor until they escape.

1.4: Aim

The aim of the project is to create a horror game on unreal engine.

1.5: Objectives

- Firstly, I will have to conduct some research of other horror games to see how they build tension how they operate and what kind of sound effects they use in order to create an atmosphere.
- Secondly I will have create the BSP model of the game on unreal so I can get a feel for how the game will look and the layout of the game. As well as this I will have to create a couple models on Maya in order to really personalise my game and also to animate the character and models.
- Then I will have to apply textures to the BSP level so that it starts to look a lot more realistic and to add the sound effects for the game.
- I will then have to programme the game so that the A.I will roam around and once they make eye contact with the player, they will chase the player until the player is hidden or even killed if the player doesn't make it away fast enough.
- I will then have to animate and programme the interactive features of the game so the player can interact with certain objects.

1.6: Problems

Problems I could face when creating the game, could be that the difficulty is too easy or too hard. If the game is too easy then it won't prove to be very scary as the player will be able to walk through the game too fast and won't create a 'Scary' feel. If the game is too hard then the user will get fed up of trying to complete it and thus never complete it. Other problems I could face could be that if the game loses the tension then it will lose the horror feel to it, so I will have to make sure I use the right sound effects when necessary and to create weird/scary assets to make the user really confused and to create a whole mysterious feel to the game.

1.7: Deliverables

The deliverable for this project will be a PC horror game, that is set within a hotel, when the player wakes up everything has changed and there will be an eerie atmosphere as the player solves puzzles and tries to uncover what happened to have changed the hotel.

1.8: Required Resources

Resources that will be required:

- Desktop PC
- Unreal Engine 4 software
- Maya Software

References

Donovan, T. (2011, May 12) *Beginner's Guide: Survival Horror*, Available at: <http://www.eurogamer.net/articles/2011-05-11-beginners-guide-survival-horror-article> (Accessed: 24 April 2017).

■ Muir (2013). *Horror Films FAQ*. Hal Leonard Corporation. ISBN 1-4803-6681-1. Retrieved 1 January 2015.

Madigan, J (2015, October 29) *The Psychology of Horror Games*, Available at: <http://www.psychologyofgames.com/2015/10/the-psychology-of-horror-games/> (Accessed: 24 April 2017).

Satheesh PV (2016) *Unreal Engine 4 Game Development Essentials*, : Packt Publishing.

Bibliography

<https://en.oxforddictionaries.com/definition/horror>

<https://www.gamespot.com/reviews/five-nights-at-freddys-review/1900-6415896/>

<http://www.gamesradar.com/why-pt-first-real-horror-game-years-and-best-game-ps4/>

<http://thefederalist.com/2015/03/03/six-elements-that-make-for-a-quality-survival-horror-game/>

http://dispatches.cheatcc.com/1233?utm_source=zergnet.com&utm_medium=referral&utm_campaign=zergnet_292369