Forgotten Fears

Hannah Farrelly-Spain

N00211013

Supervisor: Joachim Pietsch

Second Reader: Timm Jeschawitz

Year 4

DL836 BSc (Hons) in Creative Computing

Abstract

For this project I have decided to make a single-player thriller horror game using Unity as the main engine along with assets from the Unity asset store. The game will be loosely based off Little Nightmares and My Friendly Neighbourhood where the players are thrown into the thick of it with little or nothing to defend themselves where one hit will be the end of their journey. For the design I wanted to do it in the style of the VR game Job Simulator, the game is a low poly style which I think would work to my advantage when trying to design / create my own assets. For my research I decided to investigate what makes horror media, specifically some of the top thrillers and psychological horror video games, so captivating to the masses in terms of gameplay and mechanics. As mentioned, at the start of this project, I had planned to create some of the assets for the environment along with my own character models using Blender. However, as time went on with development, I realized that I would not have had ample time to make character and enemy models along with trying to rig animations for their walk cycles, but I like to try and add them in the future.

Acknowledgements

I would like to thank Joachim Pietsch for his supervision on this project. I would like to particularly thank my friends I have made on my journey in IADT for the help and support they have provided me throughout the years of collaborating with them.

The incorporation of material without formal and proper acknowledgement (even with no deliberate intent to cheat) can constitute plagiarism.

If you have received significant help with a solution from one or more colleagues, you should document this in your submitted work and if you have any doubt as to what level of discussion/collaboration is acceptable, you should consult your lecturer or the Course Director.

WARNING: Take care when discarding program listings lest they be copied by someone else, which may well bring you under suspicion. Do not to leave copies of your own files on a hard disk where they can be accessed by other. Be aware that removable media, used to transfer work, may also be removed and/or copied by others if left unattended.

Plagiarism is considered to be an act of fraudulence and an offence against Institute discipline.

Alleged plagiarism will be investigated and dealt with appropriately by the Institute. Please refer to the Institute Handbook for further details of penalties.

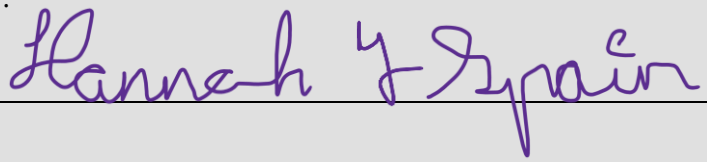
The following is an extract from the B.Sc. in Creative Computing (Hons) course handbook. Please read carefully and sign the declaration below.

Collusion may be defined as more than one person working on an individual assessment. This would include jointly developed solutions as well as one individual giving a solution to another who then makes some changes and hands it up as their own work.

DECLARATION:

I am aware of the Institute's policy on plagiarism and certify that this thesis is my own work.

Student :

Signed 

Failure to complete and submit this form may lead to an investigation into your work.

Table of Contents

1 Introduction (1½ pages)

My main objective for this project is to create a presentable and fully functioning thriller horror game loosely based around the atmosphere of Little Nightmares with influence from My Friendly Neighbourhood. The games story was to be based around how most streamers tend to mod older technology with newer part, such as old CRT TVs, to be used for display or aesthetic purposes, even though most of them cannot handle the power needed to output this and often result in them getting screen burn or blowing out and breaking down completely. I thought that this fad could lend itself to becoming a horror game based on you playing as a retro computer trying to escape the faith that fell before their friends before them, gathering useful key items to unlock the way forward while avoiding enemies to get to the Amalgamation, a corrupt being located in The Streamers Workshop made of a collection of failed and broken projects hellbent on adding you to the mix, with shutting them down for good being the only way to survive.

I wanted to challenge myself with trying to create some of my own character models for the player, enemies, and boss along with their attack and walking animations using Blender, having had a little experience with it in previous projects. I also wanted to try make some of my own assets to be used in the levels as well as attempting to create a death animation for them player should they be killed during their journey. Since I'm more interested in the designing side of game dev, I thought it'd be best to keep the coding simple with having the movements and camera being basic but not too much and to not go too far with the enemy AI so that I could give this project my best effort, having never worked with pathfinding before as well as not being too confident in my programming abilities.

2 Research

For this project I wanted to create a Thriller game based around the idea of machine corruption with the influence of modern technology and influencers, the aesthetics of divine machinery and the distortion of reality would make for an interesting horror game. I was also heavily influenced by some Indie horror games such as Little Nightmares 2 and FAITH, both of which touch on the themes of corruption and technology.

2.1 Psychology of Horror media

One of the main questions people keep asking is why do people enjoy horror? Why would anyone enjoy watching or in this case playing horrifying games where you are constantly being put into uncomfortable situations or having to fight for your life? When looking into why people enjoy playing horror games they found that most people use them to gain a sort of natural high from the fear they are able to replicate, this is because whenever we are in danger or in stressful situations a part of our brain called the amygdala releases triggers our fight or flight reflexes, causing our adrenal glands to produce 2 types of stress hormones: adrenaline, which triggers the release of dopamine and endorphins, and cortisol. This mixture of hormones creates a natural high to us especially when

mixed with the wave of relief we often get after overcoming or escaping the threat, this is why most games have so much anticipation built into them "*There are several studies on the science of fear that suggest the sensation gives humans somewhat of a "natural high." Our adrenaline rises, endorphins and dopamine are released, and we get hyped due to these short bursts of fright*." One study showed that sound in horror is crucial for viewer experience, making the point that delayed visual cues accompanied with loud and sudden audio cues, leave the viewers imagination to run wild, filling them with dread and anxiety "*Horror films also use evolutionarily resonant auditory and visual techniques to augment our anxiousness. One is by reducing the time we have to interpret visual cues. Thus, anxiousness and fear are inspired by what we see, as well as by how long we get to see it*." Another study had made the point that people with low empathy and fearfulness were more likely to enjoy and seek out horror media along with those who displayed higher empathy "*Individuals high in empathy will express more negative affect regardless of a successful resolution to the threat in the film*" and surprisingly that the outcome of the protagonist also influenced viewers and that "*a positive outcome for the protagonist and a poor one for the antagonist are the key predictors of satisfaction with a film. If neither occurs but a threat is removed, this would also lead to a satisfactory experience, but the experience would be diluted*."

2.1.1 Psychological Impact on Players

As previously mentioned, when looking into why people enjoy playing horror games "*There are several studies on the science of fear that suggest the sensation gives humans somewhat of a "natural high." Our adrenaline rises, endorphins and dopamine are released, and we get hyped due to these short bursts of fright*."

Another finding was that consuming horror media often can help viewers develop or further develop their empathy, that when viewers or players take into consideration the protagonist's perspective they show signs of personal distress and empathic concern for the characters "*Research has shown that horror movies can elicit people to feel empathy...Additionally, the authors found a tendency for those who enjoyed horror to show greater empathy and compassion levels…Exploring the benefits of psychological horror video games is crucial as it offers players a broader perspective on mental health when confronted with these themes*."

Playing horror games or consuming horror media has shown to not only impact players mentally, as it has been proven that playing video games in general helps reduce peoples stress and provides them with an escape from the mundane and taxing of real life, but physically as well. They not only improve your immune system and elevate your mood, but it has also been seen to develop your fear response "*When you watch something that frightens you, you simulate a fear experience. Your brain thinks it's real, and it learns from the experience*." But many studies have shown that playing horror games helps us to produce more white blood cells and while the intensity of horror will cause your heart to race, this helps to strengthen the heart as it provides the heart with a workout "*Video games tend to raise your heart rate in proportion to the amount of fear you experience. Over time, this can help your heart work harder for you*."

2.2 Principals of Horror Game Design

While horror games tend to follow some of the fundamentals of game design with both level design and gameplay mechanics, they often deviate to follow their own set of rules. Depending on the type of horror you play they will have different standards to uphold to ensure the players get the most out of their experience. During this research they share a similarity with both design and mechanics, but they also have 3 - 4 fundamental principles to follow. The first 2 being similar to each other are to **Disempower the player** and to **Relinquish control** from the players - some games achieve this by not supplying the players a weapon to defend themselves with or by taking away their ability to fight entirely leaving them completely helpless, but narrative designer Michel Sabbagh has argued that limiting a players resources is the best way to induce and increase that feeling of panic while also encouraging problem-solving "*While giving the player tools for means of survival isn't a bad design choice in and of itself, making said tools extremely powerful will not only make the player potentially reliant on them, but it will also shatter the sense of vulnerability that's essential in making the experience palpably frightful. The tools that the player is given should have limitations of their own, which will compel the player to approach parlous situations in a more methodical and careful manner*" the third technique is to **Subvert the player's most treasured memories** - it has been proven that subversion is an essential aspect of horror and that the most popular method is to destroy the players sense of safety and nostalgia by reimagining their core childhood memories as developer Gavin Eisenbeisz as gone on record to say "*Doing this is really enthralling to people for a number of reasons, all of which make a game more readable and intriguing. It mainly adds a sense of familiarity which gives people an instant connection to your game, as well as a sense of irony, which people find amusing and memorable. This makes the game stand out and is huge for branding*." The last technique is to **Earn the player's trust, then turn it against them** - most gamers are able to find a sense of comfort and safety when they are finally able to find either a save station or a safe room, both of these are universally known as the only safe haven from any enemies and being the one place most players retreat to in order to catch their breath or to take a moment from the stress, but what happens if that rule is broken? Most of Mason Smith's GCD 2021 talk MORTIS 101: 'FAITH's' Horror design toolkit could be used as reference but one of the most intriguing arguments that he made comes from when the relationship between developer and player is disrupted and when the developer is no longer the players advocate "*What parts of the game does the player think does a player think are safe? And how do you invade that?... What parts of the game does a player think is safe? Is it safe room? Is it a particular corridor or hallway? Is it a particular part of the game? It can be tempting to never let the player feel safe like they're always in danger…But I believe in that playful relationship between designer and player, especially in horror. But you have to find those moments, right those vignettes*."

2.3 Principals of Horror Sound Design

For a horror game to be considered hugely successful they must have decent sound design, whether that be in the form of ambience, score, or sound effects. The purpose of adding effective sound effects to a horror game is to elicit either a sudden burst of fear to the players in the form of a jump scare, to launch the player into a panic or to signal to the player impending danger, players imaginations often try to fill in the gaps with whatever horrifying beast or source they can conjure, which feeds into our innate fear of the unknown. A few studies have been done in relation to the

effects that proper sound design has both on the game and the players experience, it was found that monitoring players heart rates that when horror games were played without sound or music showed a significantly lower heart rate compared to those who played the game with sound and music "*in this paper we have specifically focused on the sounds that are supposed to enhance our experience of a game... Our observations made it clearer that it is possible that some parts of the game world might be ignored by players as a result of poorly integrated background music or sound effects. The data analysis shows that sound in horror-games clearly affects players' experience. Due to the fact that most participants' heart rate increased or that their playstyle changed when sound was involved, is a clear sign that sound affected the player experience to some extent*." It was also found in that study that players who played the game with no sound often missed crucial cues in the story or environmental cues that would help them progress the story.

2.4 Top Rated Psychological and Thriller Horror Games and what makes them so successful?

For this segment there were no studies done that I could find on these games for me to research into what makes them successful as individual horror games, however I decided to go by the highest rated games in each genre and investigate some of their reviews, using this and some of my own knowledge from my own gameplay of some of the examples used to make my arguments.

2.4.1 Silent Hill 2 (Original + Remake)

Silent Hill 2, and with the release of its remake, is one of the top-rated psychological horror games to be released in recent years and is seen as a pillar of the horror game community, with what looks to be a simple premise. A man returning to a town he and his deceased wife visited in search for her after receiving a crypted letter from her telling him she'll be there waiting. However, as you unravel the layers this game has, you come to realise there is much more beneath the surface. The narrative masterfully escalates the tension and devises an emotionally intense journey you experience alongside James. It boldly explores the darkest corners of the human psyche, presenting topics that most games steer clear from. The symbolism in the town's environments, reflecting its inhabitants' inner turmoil, is both masterfully done and disturbingly relatable in its bleakness. The depressing and dark hallways of Silent Hill resonates with personal struggles "*Much of Silent Hill 2 is themed around a descent, and there's a clear gradient in the visual texture of each area that communicates that downward spiral. Locations initially feel unkempt and abandoned, such as apartment blocks with peeling wallpaper and empty cupboards*."

If the story and characters were not enough, the game also utilises its music and sound effects, that were so carefully inserted into this world, to lure us into a hypnotic trance and blow us away with its use. The unpredictable yet deliberate timing of the sound effects, and the lengths that were went through to ensure that each random sound would have the greatest impact / effect, lures us the players in with each playthrough, while making each iteration we play feel uniquely unsettling and captivating. The soundtrack is an amazing interplay of atmospheric soundscapes and pregnant silences, a combination that sets this game in a league of its own. "*but the sound alone creates an atmosphere of NOPE-level fear more effectively than visuals ever could. Whatever the hooves are attached to sounds so heavy too, and this conjures up all kinds of monstrous images in your head about what it could be*."

Lastly the puzzles and unsettling atmosphere of this game is unrivalled to any other. Everything about the world feel wrong, presented a feeling akin to the uncanny valley that oddly fits into the game's narrative, with familiar elements being twisted and distorted into something deeply unsettling. Details such as finding multiple James slumped dead either over a table or lying motionless on the ground as you are on your way to confront Eddie or stumbling across a mannequin wearing your late wife's clothing with a flashlight in it, these elements of psychological horror set this game apart from any other trying to replicate it. "*Solving puzzles to further the plot dates back to the original Silent Hill and has since been the series' signature mechanic. Players will need to brainstorm and come up with solutions to various puzzles that keep the gameplay feeling fresh even after hours into it*." Tying into the game's weirdness, it will present the player with 1 out of 8 canon endings depending on choices the players make along their journey as well as how you the player play as James throughout the course of the game, including his health status and item interactions, will significantly change the outcome of the ending. These endings range from canon 'True' endings such as the "In Water" and "Leave" , resulting in James either driving his car into a lake as a form of self-punishment or leaving Silent Hill accepting his role in his wife Marys death with a little girl called Laura, to the bazar "UFO" and "Dog" endings, where James gets abducted by a UFO as the name suggests and James breaking the forth wall of the game and turns into a dog. With the potential to experience all these alternative endings it gives the players an inventive to do multiple playthroughs of the game.

2.4.2 P.T

P.T originally released on Steam , despite having been cancelled in 2015, is still considered by veteran gamers as a hallmark of horror. It was one of the first horror games to break every cliche of survival horror game, you immediately get thrusted into the game without knowing who you are, what you are doing here and how to control. The game is unpredictable. Doors open and close randomly. The light could be off or turn to blood red at any moment. You could walk through the hallway safely while nothing happened, and then later you are trapped in the closet with a foetus while someone is watching you through a hole to name a few instances. While still a horror game, P.T. usually avoid the use of jump scare. This is where its amazing use of sound design comes into play, while it is scary and unpredictable, the crying baby sound is the most captivating. It created a sense of dread and left the detail of the grisly murder to the player's imagination "*this in itself a powerful means of disempowering you, stripping you right down, removing what you know, and forcing you approach its horrors honest and functionally naked*."

2.4.3 Outlast

Outlast is another game that is a pillar of the horror game community, having been made from two former AAA horror game developers whose main objective was to create a horror game that was the embodiment of grotesque, terror, and depravity. You play as a journalist who enters an asylum belonging to Murkoff Corp to search for evidence of shady things happening there only to realise it would turn into your worst nightmare ever. As you explore the asylum in search for evidence of Murkoff's illegal activities , you encounter multiple tortured and seemingly mutilated patients who are bloodthirsty to kill and all you can do is to either run or hide , or you die "*That's the motto of Outlast games , " run , hide or die !"*."

The eerie setup, dark corridors , haunting background music , sudden jump scares and terrifying chase sequences all made up for a real heart stopping terrifying experience. These encounters / moments often discouraged many from taking steps ahead especially when anything could jump out for you from the darkness "*Outlast is filled with nearly every type of scare imaginable – shapes dash out the corner of your eye, lights conveniently shut off when you enter certain rooms, and bodies that seem dead suddenly burst to life*." The suddenness of tensed music playing will make you sweat thinking what exactly is ahead that is resulting in this music.

What makes the chases especially difficult is that players won't know the layout of the area / floor when playing 1st time so not knowing where to run and hide while being chased often results in panicked and frenzied decisions all the while you are reliant on your night vision to help you evade danger, pair this with the locked door or dead end. One wrong turn and you might end up at a dead end while the enemy corners you and beats you to death and all you can do is stare at your screen and watch your death or try your best to make your escape. The only solution to avoid any chase, is to use stealth to your advantage in getting past the enemies. Your camera's night vision is your lifeline in the game "*There are no weapons to be found in Outlast, so you're forced to endure the darkness armed only with a night vision camera and the ability to run*" especially when players must navigate through prolonged periods of darkness sometimes while evading enemies or while in chase. However, this tool is not free and comes at the cost of draining out your camera's battery but luckily players are able to scavenge for more with some being obvious, either at the beginning of a dark section or puzzle, and others being hidden extras that players often must really search for.

2.4.4 Five Night's at Freddy's

Five nights at Freddy's, or FNAF, is seen as a pillar of the Indie game community, with its launch success being the immense recognition it got from early YouTubers "*The release of the first game coincided with the rise of streaming culture on YouTube, and creators like PewDiePie, Jacksepticeye and Markiplier earned millions of views by reacting to Five Nights at Freddy's, screaming into the camera and hyping up the experience…Even if fans weren't playing the game themselves, they were sharing the experience with their favourite creators; the wave of YouTube "Let's Play" videos proved more effective than any marketing campaign.*" The game has you playing the role of a security guard on the night shift at Freddy Fazbears Pizza, a nod to franchises like Chucky Cheese, where your sole role is to make sure no one enters. However, you find out on your first night that your job is not as advertised, the animatronics move freely around the pizzeria, and you must try and survive 5 nights.

While the game doesn't have much in regards to controls , consisting of you using your mouse to look from left to right, to close the doors, check the lights and flip between your cameras, it makes up for it in gameplay and suspense, providing the players with a truly horrifying experience as the players are unable to move from their office "*By making a simple change to the typical horror game mechanics and forcing players to remain stationary while the monsters came to them, the game created a type of anticipation and fear that players had never experienced before*." You are given limited power that will deplete with the use of any of the cameras, doors, and lights so players must be strategic with their usage, because if the power goes out before 6AM then it is game over. Every night the player is greeted with a voice recording from the previous night guard, who we hear at night 5 met his untimely demise from the animatronics, giving us not only an insight to the bigger story of the games but advice as to how to defend ourselves against the animatronics.

The game incorporates eerily disturbing sound throughout your nights, ranging from simple footsteps outside your doors to groans of pain and death rattles coming from the animatronics themselves, as if there is something living within them desperate to get out. This is hinted to be the cause of evil spirits haunting the animatronics, as the former security guard tell us of the horrific disappearance of 5 children in the 80's by an unknown serial killer along with "The Bite of 87" where an employee had their frontal lobe bitten into by one of the main 4 animatronics, newspaper snippets of which can be seen throughout the halls of the pizzeria. This along with the constant hum of the office fan puts the player on edge, having their nerves on edge and keeping them in suspense.

2.5 Conclusion

Looking across all the research that has been conducted, the popularity of horror media derives from the use of quality sound design and effects, intense and palpable atmosphere along with enriched level designs. A game must be able to keep players engaged while also keeping them challenged whether that be with the gameplay itself or with the idea of completing the game to its entirety. These points have been proven when looking at the most popular games in each category and what mechanics they offer, as they adhere to the rules of creating horror games. Even some of the popular Indie games have shown to follow the core rules of making a horror game, while twisting them slightly, which could lead to Indie games becoming the future of gaming or at least competing on the same level as AAA games.

3 Requirements

3.1 Introduction

When looking into the possible requirements for making a thriller horror game, research was done on a range of existing titles ranging from AAA to Indie such as *Little Nightmares, Little Nightmares 2, and Faith: The Unholy Trinity.* All games lend themselves to similar mechanics where the player has either little to no way of defending themselves or a limited resource to be used with the upmost care and calculated risk. *Little Nightmares* being one of the a shining example of a game that is a triple threat in regards to its eerie atmosphere and sound design, its smart gameplay mechanics from its puzzles to using a seemingly unfair power-imbalance when it comes to the enemies and environments to its advantage and finally it beautiful use of limited lighting in its environments in order to, quite literally, keep its players in the dark about what is to come next.

3.2 Requirements gathering

3.2.1 Little Nightmares

In the unsettling world of Little Nightmares, a puzzle-platformer horror adventure in a 2.5D style, you take on the role as a girl named Six, a small, hungry child trapped within the confines of the mysterious Maw. An iron vessel that is home to terrifying, distorted beings, with your only objective being to escape. The game challenges you with platforming and environmental puzzles that block your progress along with carefully navigating through hostile environments. Stripped of any means to fight back, the game forces you to rely entirely on stealth and clever use of the environment around you to evade the nightmarish inhabitants, while trying to find tools and keys to help you advance your way out of The Maw. Traverse eerie landscapes, overcome platforming obstacles, solve mind-bending puzzles, and exploit every shadow to outwit the Maw's terrifying denizens.

3.2.2 Little Nightmares 2

Little Nightmares II is a puzzle-platform horror adventure game following the story of Mono, who must work together with Six, the protagonist from the previous game, to survive the horrors of the Pale City and discover its dark secrets. *Little Nightmares II*, like its predecessor; it takes place in a 2.5D world. The player must explore the world, occasionally encountering platformer-like situations or being blocked by puzzles that must be solved to proceed. Unlike the first game, the player is not completely helpless; Mono can grab certain items and swing them to break objects or to fight back against smaller foes, although he, like Six, must rely on stealth and the environment to evade larger foes. The player must explore the world, occasionally encountering platformer-like situations or being blocked by puzzles that must be solved to proceed. The player must also use the environment to their advantage to evade the various monsters and beings inhabiting the Pale City.

3.2.3 Faith: The Unholy Trinity

Faith: The Unholy Trinity is a retro styled 8-Bit survival horror game, meant to take place in the early 80's, where the player takes on the role of a priest known as "John Ward" through the games 3 chapters, where he attempts to complete a failed exorcism to prevent the summoning of a dangerous and powerful demon. The games first act follow's Ward's return to the Martin house in Sterling, Connecticut, where one year before he and his superior, Father Allred failed to exorcize a demon from a young girl named Amy. Father Allred was killed along with Amy's parents, and the girl was placed in a psychiatric hospital from which she has now escaped. The second act introduces another priest, the shotgun-wielding, duty-driven, fan favourite Father Garcia, who is attempting to remove a demonic entity that appeared in Chapter 1 from a young boy. The climax of the game is a third act that follows Ward in the days leading up to an event called the Profane Sabbath that will summon a demon named MALPHAS, and another demonic entity known only as 'Gary' will open a portal to hell.

The game forces players to rely on the exceedingly simple one-button + cross pad controls. Ward's only action, apart from rare occasions when he handles a story item, is to raise his cross, slowly damaging whatever demonic entity or spirit is proximity to him. This lifeline, while being severely limited, manages to fell tremendously powerful at the same time being the only safeguard you have, as all but one enemy in the game will kill him with a single hit.

3.3 Requirements modelling

3.3.1 Functional requirements

The following is a list of requirements that are necessary for the game to function as intended.

- **MOVEMENT** - The player will be able to move throughout the levels using WASD and will be able to jump around using SPACE. The player must also be able to look around their environment using the mouse.

- **COLLECTION SYSTEM** – The player will be able to traverse the level and collect key items to advance to the next level.

- **HEALTH SYSTEM** – The player will have no health system in the game. I plan to have it be a one shot one kill to create a more suspenseful and tense atmosphere to make it harder for the player to advance.

- **ENEMIES** – The enemies will be made to be easily avoidable by slowing down their movements and creating the levels more spacious. This will give the player some breathing room when running around.

- **MENUS** – The game must have a main screen menu for the player to have a starting point along with a pause and settings menu, so that the player has the option to exit to the main screen or to adjust the volume of the game and / or it effects.

3.3.2 Non-functional requirements

These non-functional requirements were to make the game more atmospheric while also adding some pleasing visuals to the game.

- The menu screens must be consistent with their designs and must be east to read.
- The games visuals and designs must be consistent with each other.
- The levels are to have different background music to add atmosphere to the game.
- Enemies will emit noises to alert the player.
-

3.4 Feasibility

For this project I used Unity to build the entirety of the game, minus the scripts themselves being done through VSCode. I created some basic scripts (camera, movement, and collection) along with some enemy AI, hit boxes and trigger scripts while using Unity's built-in tools to create the games UI and some minor animations.

Initially I had planned to use Blender to design and rig some of my own characters models and their movement animations, along with attempting to create my own assets to use in my games environments and game animations. Here I fear I was a bit too ambitious with my ideas because, while I had a little experience with Blender and modelling assets and had done extensive research in how to rig up models and creating my own animations along with finding tutorials, I found that in order to relearn the basics for myself alone would have taken up too much of the development time for me to then produce a functioning game so the animations, both game and characters, had to be scrapped.

Considering this I had already been looking into alternative assets, models and animations from the Unity Asset Store and Mixamo, with the Asset Store having 1000+ ready-made assets for you to use varying from one that you can buy or ones to use for free where all you need to access them is to create an account. Mixamo acting in a comparable way allows you to select a pre-made character model, or upload your own, and place a variety of pre-rigged animations ranging from walk cycles to fighting patterns onto the selected character to be downloaded.

With all these readily available tools and software licenses, making the project within the allotted time that would be presentable would be achievable even after some setbacks.

3.5 Conclusion

This chapter introduces the initial requirements for developing a thriller horror game by referencing existing successful horror games like Little Nightmares and Faith: The Unholy Trinity, highlighting their effective mechanics such as player vulnerability, environmental interaction, eerie atmosphere, and sound design. This chapter then delves into Requirements Gathering, providing summaries of the three analysed games along with Requirements Modelling, outlining both functional and non-functional requirements for the proposed game. Finally, the chapter addresses this projects Feasibility. It states that Unity was used for development, with scripting in VSCode such as basic camera and movements, a collection system, and enemy AI.

4 Design

4.1 Introduction

From the start of this project, I had already decided that I would lean towards stylising this game in a low-poly format having based it off the design of another low-poly game *Job Simulator*, not only would this be easier to design but also to acquire assets for should the need come. Originally, I had planned for the game to have three levels, a starter / tutorial level, a normal level, and finally a boss level. However, the game had to be scaled down to two for it to be functional. The main concept was based on the aesthetic of *Devine Machinery* – *"a contemporary aesthetic which blends themes of technology with religion, focusing on the artificial and associating it with the divine. Commonly utilized imagery includes wires, computer monitors, transmission towers, hardware components, religious iconography, and angels.*" My idea was to create a game based on the corruption of retro technology, having been infused or blessed, with newer parts thus morphing them into twisted beings worshiping a false god, while the player would have no form of health and no way of defending themselves, having to rely on wits in order to stay alive. For the UI I decided to keep it simple with only having bare-bones menu selections, being the Pause and Settings menu along with the Main menu.

4.2 Program Design

For most of the scripts I used, I tried to keep them as concise and clear as possible by not overloading them with too many commands. I kept basic movement scripts as just that with extras such as damage or pathfinding being managed outside either with a different script or by using Unity's built-in tools instead. All the scripts for this project were written and edited in Visual Studio.

4.2.1 Technologies

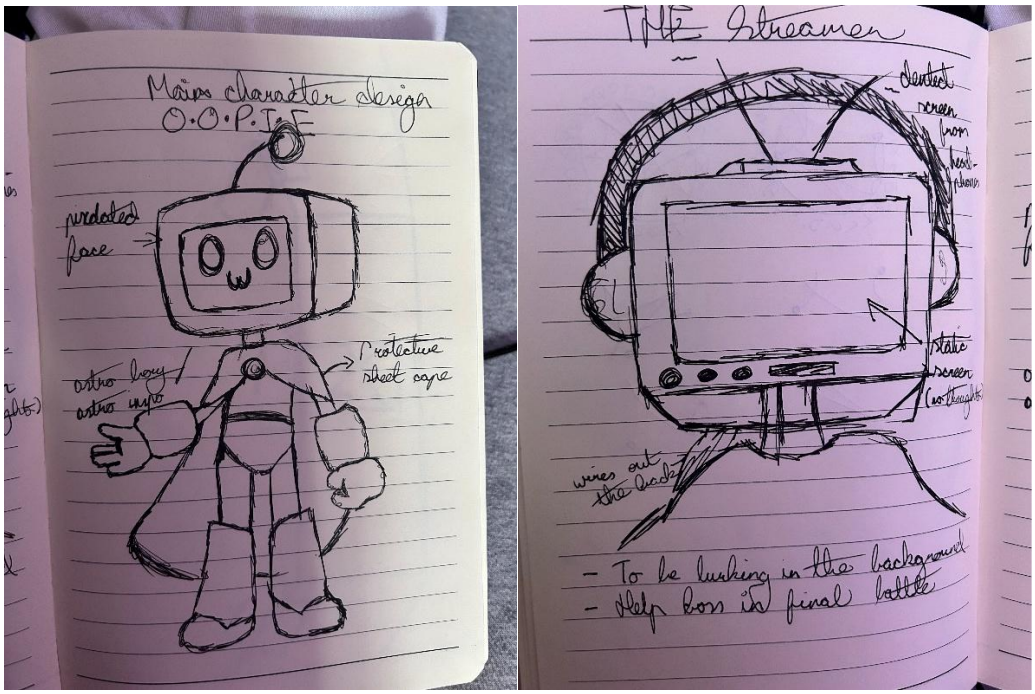
For the guts of this project, I decided to go with Unity as opposed to Unreal Engine or Godot Engine, along with using Visual Studio for all the C# scripts. My reason being that I have had experience with Unity in past projects and was confident that I could produce a playable result, but also because Unity has great support for developing 3D games with a wide range of tools such as Cinemachine to help development.

I had originally planned to use Blender as well to create 3D models for characters and assets, animations and cutscenes for this game along with trying to rid my own character walk-cycles, however I had seriously overestimated my knowledge of Blender and relearning how to use it properly would have taken up too much precious time, so it had to be sacrificed in order for progress to be made.

4.3 User interface design

4.3.1 Storyboard

The story is that you play as a retro 80's computer monitor called the Obsolete, Operating, Personal, Information, Entity (or O.O.P.I.E for short) who has woken up to find his friends to have been corrupted by new age technology implanted in them by The Streamer, worshiping them as a god for giving them new life and a new purpose. Your goal is to escape with your sanity intact, while trying to avoid being turned into a twisted version of yourself.

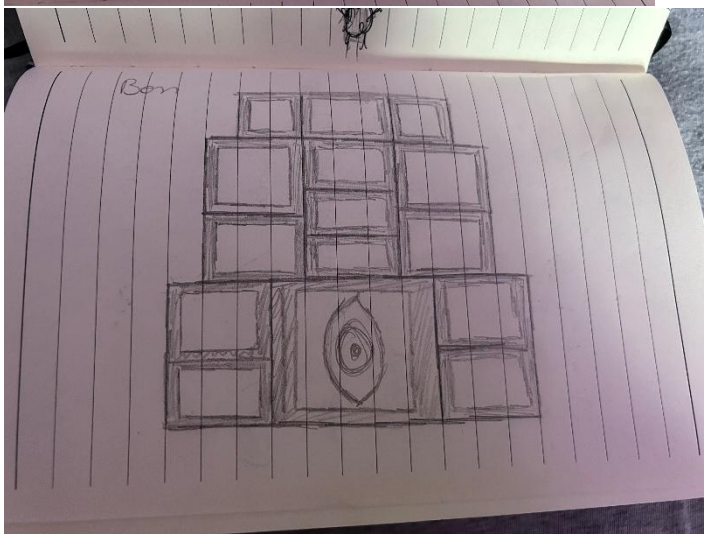


I was inspired by the fact that some streamers like to incorporate retro or older technology, such as old CRT tv's, into their setups and backgrounds for aesthetic purposes as they often display still images or video clips, where they are often fitted or modded with newer parts to be able to display these images with better sound and quality. Often these changes, if not done right, can cause considerable damage to the screens causing them to get screen burn or to burn out completely for simply not being able to keep up with what they are being used for. I thought this would be an amazing idea for a horror game story.

4.3.2 Level Design

The player traverses through two levels, being The Wardrobe and The Bedroom where they must avoid enemies and obstacles while trying to parkour their way around the environment to collect

items that will help them advance. Each level is designed with giving the player the main task of collecting a distinct number of items to advance to the next area, this being either some keys like in the Wardrobe level, or some floppy discs like in the Bedroom level. There was meant to be a third and finals Boss level where the player would have had to fight a giant amalgamation of wires and screens, successfully shutting down four core power modules while avoiding enemies and attacks, to finally escape the house. This did not make it into the final version due to time restraints.



4.3.3 Environment

I decided on having three levels, then reduced to two, to serve as an introduction / tutorial level, a main level, and a final boss level.

The Wardrobe level is where the player starts out in and serves as The Streamers storage, here is where most of their older or worn-down projects would reside. The Wardrobe is filled with old boxes, shelves and furniture that is no longer in use along with other computers, sitting unused and powered off on shelves, which could be used for future projects. I wanted the start to be similar to that of Little Nightmares, where the player is thrown into an unknown location with no real context of what is happening, only to be shown through environmental story telling.

The Bedroom is to serve as the main level, where The Streamer has all their latest gear and setup located. Here the enemies would be newer and faster models of what was left to rot in the

wardrobe, the level has the a bathroom connected to the main bedroom where the player has to search around the Streamers setup trying to find floppy discs in order to unlock the lock on the Workshops door to advance.

The Workshop was to be the final level, but did not make it into the final version. Here the player would be faced with a twisted amalgamation of all the Streamers failed and broken projects melded into one being of broken monitors and bundled wires. It was initially to have the player power-off the amalgamation by flipping four switches around the Workshop while avoiding attacks. It would have had old monitors and screens dangling from the ceiling to function as platforms for the player to avoid attacks or to reach switched, with the main boss being perched on top of a workbench with wires sprawled across the floor and twisting around any furniture in the area.

4.4 Conclusion

This chapter looks at how the game's core concept revolves around the "Devine Machinery" aesthetic, blending technology and religion with the narrative involving corrupted retro technology, personified as twisted beings worshipping a false god ("The Streamer"). it also talks about why Unity was chosen compared to other gaming engines along with how originally, Blender was intended for 3D modelling and animation but had to be excluded to maintain the project timeline. Lastly, this chapter looks at the environment and level design decisions that were made to be in keeping with the games original storyboard, while touching on ideas that had to be cut.

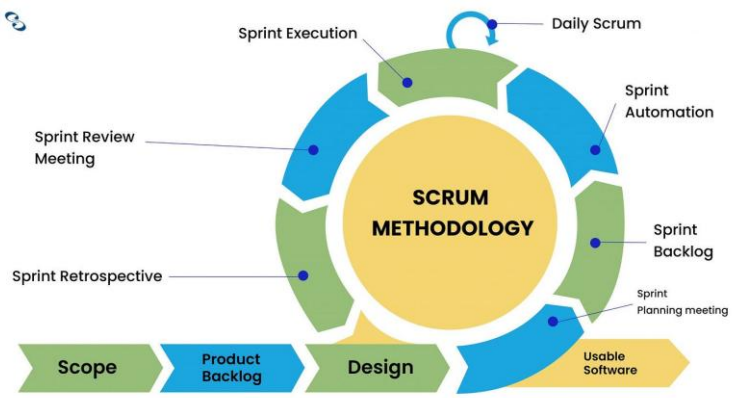
5 Implementation

5.1 Introduction

This section will outline the technologies and tool and were used to build the game within the Unity system. It goes into detail of the assets, UI and systems that were integrated to make a functioning thriller game, such as using some of Unity's build-in tools to create the third-person swing camera and all of the UI, as well as touching on Blender – an application that would have been used to create original models for the character and enemies in the game while also being used to make level assets.

5.2 Scrum Methodology

By using SCRUM methodology for the duration of the implementation phase it would function as a blueprint for the development of this project while also being a way to manage tasks that were to be completed or changed to have the game be functional at the end of the deadline. With each sprint being reviewed on what was and was not completed, monitoring any backlog and what were the next steps.



I found that the used of this framework helped manage fallouts in pervious plans in this project while also making building of the project over the allotted time more manageable. An example being that early sprints were to be focused on core gameplay mechanics, such as movement and camera work to be done and out of the way early on, while later sprints were to be used primarily for design of menu UI, implementation of background music and polishing and refining level designs along some core mechanics. However, while some plans did not end up being implemented into the final product, the use of SCRUM helped manage these changes and made sure that the rest of the game would be in a playable.

It would also allow the chance to revise the progress that has been made, to be able to fix any significant issues that were found or to adjust any plans for future development. By breaking the project into more manageable sprints, it made it easier to monitor the results so that the original goals could be met even if adjustments were needed.

5.3 Development environment

Unity was primarily used to build this game, along with Visual Studio as the IDE (Integrated development environment) for writing and debugging the C# scripts. I chose Unity not only because I had worked on it before and was confident that I could produce a playable result, but also because Unity has great support for developing 3D games with a wide range of tools such as Cinemachine to help development.

All assets, while some were initially planned to be made using Blender, scenes and scripts were managed by Unity's editor interface. This along with Unity's Console were frequently used for testing whether it was for error checks, integration or log output and debugging.

For the duration of this project, I used Miro to organize tasks in a "To Do" list format, gather research and inspiration for the design and development of this game along with storing tutorials that may have been useful. The list was split up into three segments labelled "To Do," "In Progress" and "Done" with each task being moved or added into their corresponding segment. This helped alleviate some of the stress of remembering what was and yet to be done within each sprint.

Finally, I used Unity for the primarily to manage version control and used GitHub to commit the finished product. In the past I found that GitHub would bugout when trying to commit changes to Unity projects and would often have previous commits disappear or not save at all, so I found it easier to just keep the work-in-progress on my laptop and in Unity, with all being saved and backed up onto a OneDrive account.

5.4 Sprint 1

5.4.1 Goal

The goal for this sprint was to create a test level that I could experiment with movement scripts and actions such as jumping, running, and crouching. I did this by creating a Whitebox level with a loose layout of the level, with basic cubes as obstacles and having a capsule be a temporary player model, to be able to assess and develop the players movement and camera.

5.4.2 PlayerMovement + First person / Third person camera

I first made a basic first-person camera script to assess the players movement with WASD and once that was completed, I rigged the main camera to Cinemachine and edited the script to swing around the player. I experimented with the idea of having the player being able to build up speed, to be able to clear larger jumps, and having the ability to double jump on top of the basic movements.

I did run into issues with the player having momentum, being that it was difficult to control and would result in the player running uncontrollably around levels along with the double jump causing

the player to either glitch out in-between jumps and the crouch causing the players camera to glitch or for it to not work entirely, so these features had to be removed which just left the player being able to move and jump.

5.5 Sprint 2

5.5.1 Goal

The goal for this sprint was to fix any outstanding errors that were to be fixed with the player movement and third-person camera scripts, as well as to start mapping and building the levels themselves and assets using a mix of Blender and the Unity Asset Store. I also wanted to gather sound effects and music to be used for both the levels and menus.

5.5.2 PlayerMovement script

To get the player movement and camera script to work I had to completely re-write them, so I started by removing the ability to crouch, sprint and double jump from the script. Apart from the basic movements I added a ground check to help fix the initial issues with both the players momentum and jump by checking if the player is touching a layer called “isGround” set in Unity.

```
    isGrounded = Physics.CheckSphere(groundCheck.position, groundDist
    groundMask);

    if(isGrounded && velocity.y < 0){
        velocity.y = -2f;
    }

    float x = Input.GetAxis("Horizontal");
    float z = Input.GetAxis("Vertical");

    Vector3 move = transform.right * x + transform.forward * z;

    controller.Move(move * speed * Time.deltaTime);

    if(Input.GetButtonDown("Jump") && isGrounded)
    {
        velocity.y = Mathf.Sqrt(jumpHeight * -2f * gravity);
    }

    velocity.y += gravity * Time.deltaTime;

    controller.Move(velocity * Time.deltaTime);
}
```

5.5.3 Blender assets + Unity Store

Initially for this project I wanted to try my skills in blender to create my own assets to be used in my levels, specifically some of the more horror elements. However, the more I investigated doing this I quickly realized that I had been that I did not have both the skillset and the time to be able to successfully complete this, even with the tutorials I found since I had to relearn how to use Blender. But I had anticipated this, so I had selected some free assets from the Unity Asset Store to be used instead.

Unity Assets:

https://assetstore.unity.com/packages/3d/props/furniture/low-poly-simple-furniture-free-240197

https://assetstore.unity.com/packages/3d/props/retro-psx-horror-puzzle-item-pack-icon-lowpoly-250188

https://assetstore.unity.com/packages/3d/props/simple-stylized-cardboard-boxes-308830

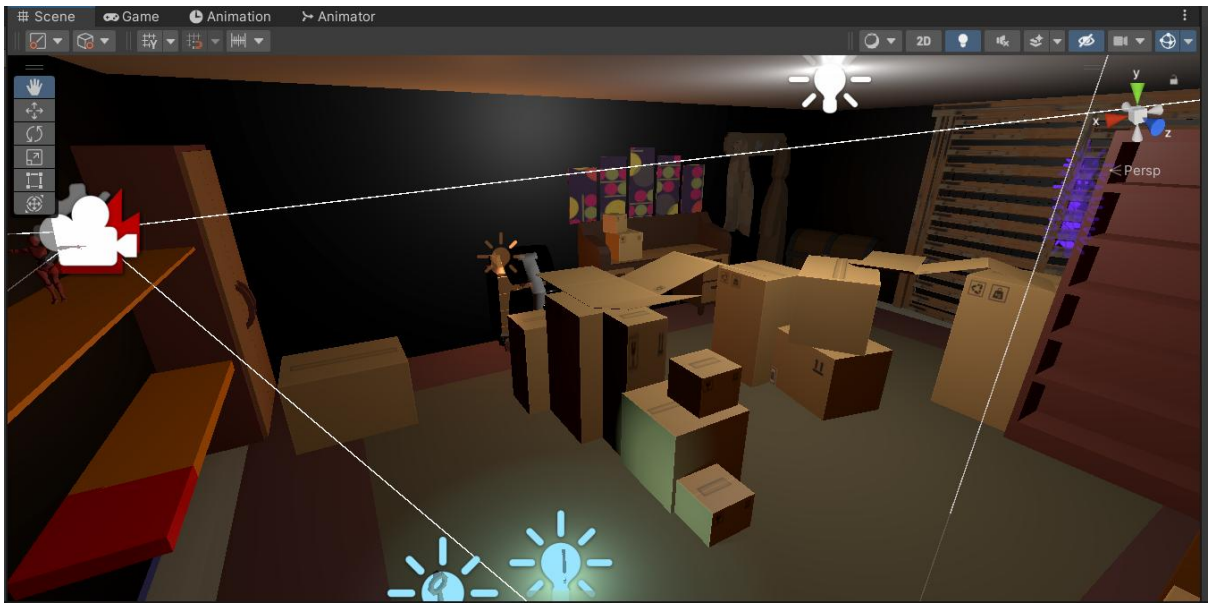
https://assetstore.unity.com/packages/3d/environments/apartment-kit-124055

Blender Tutorials:

https://youtu.be/4haAdmHqGOw?si=kQQ55_3OSisa2WQD

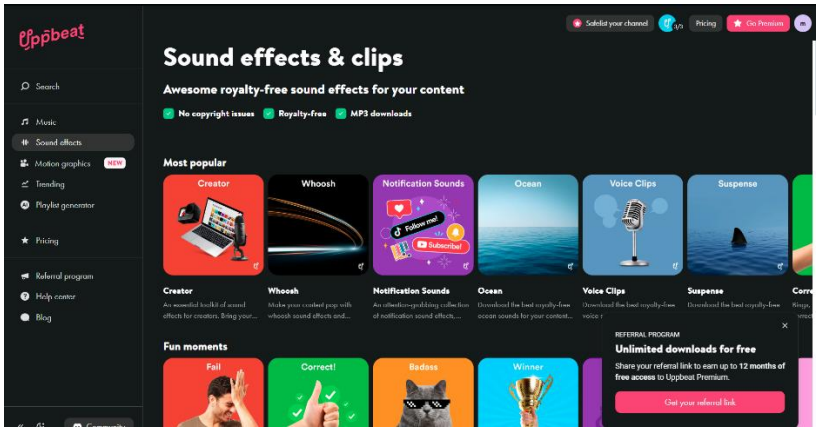
https://youtu.be/vaiucoQwQjM?si=Sya6pjtH2VTEH1DG

I was able to finish creating the first level named “The Wardrobe” that acts as a sort of tutorial level to get players introduced to the mechanics even while it has some enemies in it.

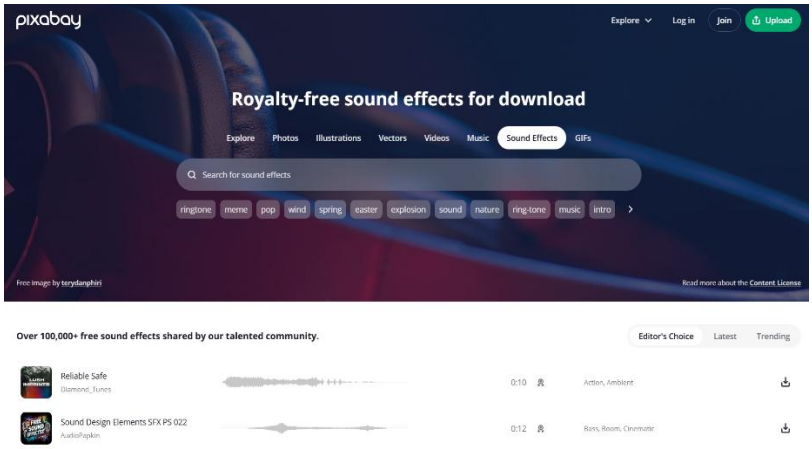


5.5.4 Sound effects + Background music

Lastly for this sprint I gathered some sound effects and background music I thought would make the game more immersive by making free accounts on Pixabay.com, Mixkit.co and UppBeat.com.



I have used all these sites in the past for the same reason and found them to be very useful, the only drawback is with UppBeat.com, because with the free account you only have limited access to some music and are only allowed to download three every month. So, to bypass this I set up three alternative accounts to be able to download all that I would need or those I thought might fit.



5.6 Sprint 3

5.6.1 Goal

During this sprint, the initial plan was to design my own character to be used in the game along with rigging some basic walk animations for the player and the enemies. I had researched some tutorials on how to do this along with drawing up some of my own designs for the models I wanted to create. However, I underestimate how out of practice I was with using Blender and as to how much time it would take not only to relearn how to use Blender properly again but to also make the models actually and to learn how to rig animations. So, unfortunately most of this sprint was wasted on unused assets.

5.7 Sprint 4

5.7.1 Goal

The goal for this sprint was to create code for the players health and the generic enemies' behaviours that would be around the levels, along with making variants with some being faster or slower and having different detection ranges for the player, along with programming the boss levels behaviour and attacks.

5.7.2 Player Health State

Since I did not want the player to have a proper health state where they would be able to take multiple hits without dying, I had to program it so that when the player would touch a certain surface it would do 100% damage.

```
void Start()
{
    currentHealth = maxHealth;
    if(OnHealthChanged != null){
        OnHealthChanged(currentHealth, maxHealth);
    }
}

public void TakeDamage(int damage)
{
    currentHealth -= damage;
    currentHealth = Mathf.Clamp(currentHealth, 0, maxHealth); // Ensure
    health stays within bounds

    if(OnHealthChanged != null){
        OnHealthChanged(currentHealth, maxHealth);
    }

    if (currentHealth <= 0)
    {
        Die();
    }
}

private void Die()
{
    if(OnPlayerDeath != null){
        OnPlayerDeath();
    }
    Debug.Log("Player Died!");
    // Add death effects, respawn logic, etc.
    // For example, you might disable the player GameObject:
    gameObject.SetActive(false);

}
```

The code above allows the players health to be set in Unity and checks for there to be any change. When the player does sustain damage the TakeDamage function calculates how much damage was dealt and subtracts it from the players current / remaining health while making sure the health stays within its bounds and, since enemies and obstacles deal 100% damage, when the player runs out of health it runs the Die function, causing the player to respawn.

I also created a DamageSurface script which could be added to an object directly or to a layer that work similarly to enemy's attack function.

```
public class DamageSurface : MonoBehaviour
{
    public int damageAmount = 10; // Amount of damage to inflict
    public float damageInterval = 1.0f; // Time between damage ticks
    private float nextDamageTime;

    private void OnTriggerStay(Collider other)
    {
        if (other.CompareTag("Player") && Time.time >= nextDamageTime)
        {
            PlayerHealth playerHealth = other.GetComponent<PlayerHealth>();
            if (playerHealth != null)
            {
                playerHealth.TakeDamage(damageAmount);
                nextDamageTime = Time.time + damageInterval;
            }
        }
    }
}
```

5.7.3 Enemy AI

The enemy script I found to be a lot harder since I had never worked with pathfinding or enemy AI before. I stared with setting the public variables that would be needed for player detection, chasing and attacking along with the enemy's health and walk speed.

```
    public NavMeshAgent navAgent;
    public Transform player;
    public LayerMask groundLayer, playerLayer;
    public float health;
    public float walkPointRange;
    public float timeBetweenAttacks;
    public float sightRange;
    public float attackRange;
    public int damage;
    public Animator animator;
    public ParticleSystem hitEffect;

    private Vector3 walkPoint;
    private bool walkPointSet;
    private bool alreadyAttacked;
    private bool takeDamage;
```

Next, I have the Update method checking for the players proximity and determining the enemy's behaviour. If the player is not in either attack or sight range the enemy will patrol around the level, if

the player is in sight range the enemy will chase the player, if the player is in both attack and sight range then the enemy will attack the player.

```
private void Update()
{
    bool playerInSightRange = Physics.CheckSphere(transform.position,
    sightRange, playerLayer);
    bool playerInAttackRange = Physics.CheckSphere(transform.position,
    attackRange, playerLayer);

    if (!playerInSightRange && !playerInAttackRange)
    {
        Patroling();
    }
    else if (playerInSightRange && !playerInAttackRange)
    {
        ChasePlayer();
    }
    else if (playerInAttackRange && playerInSightRange)
    {
        AttackPlayer();
    }
    else if (!playerInSightRange && takeDamage)
    {
        ChasePlayer();
    }
}
```

The Patrolling function has the enemy wonder between different waypoint set in its navMech, checking for the player, if there is no waypoints set then it generates a random Z and X offset for it to go to while checking to see if it is one the IsGrounded layer.

```
private void Patroling()
{
    if (!walkPointSet)
    {
        SearchWalkPoint();
    }

    if (walkPointSet)
    {
        navAgent.SetDestination(walkPoint);
    }

    Vector3 distanceToWalkPoint = transform.position - walkPoint;
    animator.SetFloat("Velocity", 0.2f);

    if (distanceToWalkPoint.magnitude < 1f)
    {
        walkPointSet = false;
    }
}

private void SearchWalkPoint()
{
    float randomZ = Random.Range(-walkPointRange, walkPointRange);
    float randomX = Random.Range(-walkPointRange, walkPointRange);
    walkPoint = new Vector3(transform.position.x + randomX, transform.
    position.y, transform.position.z + randomZ);

    if (Physics.Raycast(walkPoint, -transform.up, 2f, groundLayer))
    {
        walkPointSet = true;
    }
}
```

The ChasePlayer function simply speeds up the enemy's movement when the player is spotted while targeting the players location, causing the enemy to move towards it.

```
    private void ChasePlayer()
{
    navAgent.SetDestination(player.position);
    animator.SetFloat("Velocity", 0.6f);
    navAgent.isStopped = false; // Add this line
}
```

The AttackPlayer function checks if the enemy has already preformed an attack, if not it will move towards the players position by using Raycast and rotate to face the player, while checking to make sure that the player is still in front of them, then preform an attack. I had toyed with the idea of letting the player damage and destroy some of the lesser enemies, but since in the end there would only be one main enemy I decided against this.

```
    private void AttackPlayer()
{
    navAgent.SetDestination(transform.position);

    if (!alreadyAttacked)
    {
        transform.LookAt(player.position);
        alreadyAttacked = true;
        animator.SetBool("Attack", true);
        Invoke(nameof(ResetAttack), timeBetweenAttacks);

        RaycastHit hit;
        if (Physics.Raycast(transform.position, transform.forward, out hit,
        attackRange))
        {
            /*
                YOU CAN USE THIS TO GET THE PLAYER HUD AND CALL THE TAKE DAMAGE
                FUNCTION

            PlayerHUD playerHUD = hit.transform.GetComponent<PlayerHUD>();
            if (playerHUD != null)
            {
                playerHUD.takeDamage(damage);
            }
             */
        }
    }
}
```

I created a ResetAttack function so that there would be a slight delay between attacks to allow the player to get away, should the enemy miss while also having it so that the enemy's attack animations could have a chance to reset.

```
private void ResetAttack()
{
    alreadyAttacked = false;
    animator.SetBool("Attack", false);
}
```

5.7.4 Boss AI

While I had planned to create a boss level and AI, they did not make it into the last version, due to complications with previous enemy AI scripts that took way too much time to resolve. I decided that it was better to have functioning enemies compared to subpar enemies and an incomplete boss.

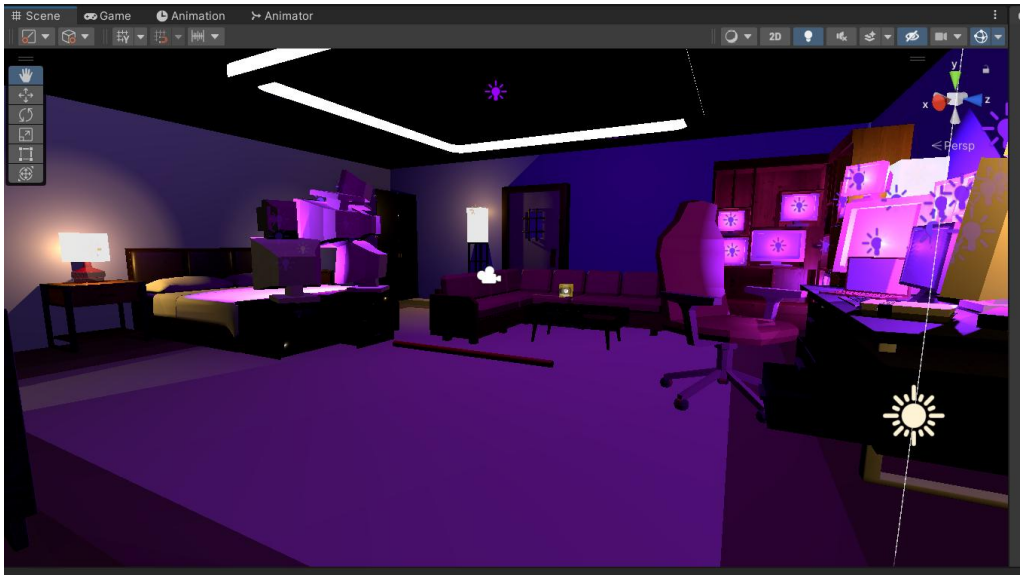
5.8 Sprint 5

5.8.1 Goal

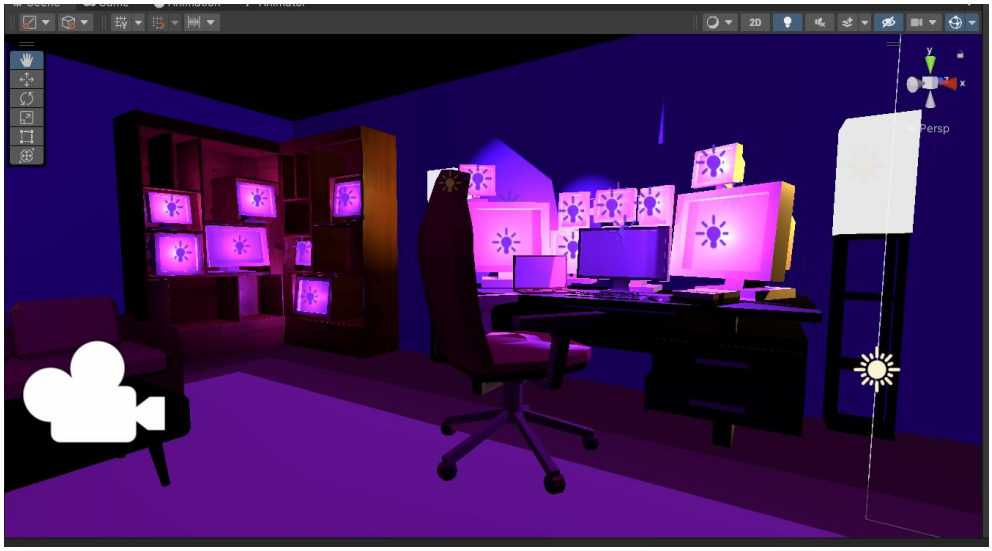
The goal for this sprint was to create a complete and functioning second level for the player to traverse around, while creating a trigger teleport system to send the player to the next level that can only be activated if the player gathers all the necessary collectables.

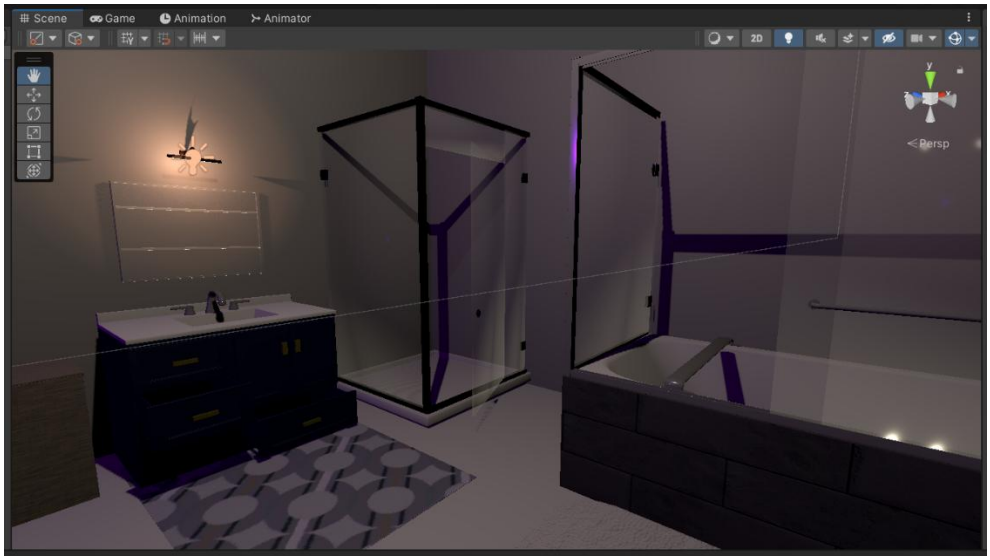
5.8.2 Second Level

I went about creating the second level using some previous Unity Store assets for the furniture, collectables, and platforms. The idea was to have the Bedroom acts as the main level while introducing newer enemies for the player to avoid but had to be scaled down.



The bedroom is made up of two areas with the other being a connected bathroom, both have several collectables for the player to find to escape the level. Instead of having normal lighting I wanted to have each screen emit their own light source to make the level more atmospheric.





5.8.3 Collectables + Script

For the collectables I chose a simple horror prop asset package from the Unity store that had some items like the floppy discs and keys that I could use. I thought that using these would be the better idea as they would be recognised as key items, plus they would have a glowing light around them to make them more recognizable.

For the collectables I made Three collaborative scripts that read from each other, on being the main Key script which checks to see if any keys in this instance have been collected, and if so, they

become inactive and are put into the players Inventory.

```
C: > Users > hanna > Major-Project > Assets > My Scripts > Key.cs > ...
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class Key : MonoBehaviour
6   {
7
8       private void OnTriggerEnter(Collider other)
9       {
10          Inventory inventory = other.GetComponent<Inventory>();
11
12          if (inventory != null)
13          {
14              inventory.KeyCollected();
15              gameObject.SetActive(false);
16
17              // if (inventory.NumberOfKeys == 3)
18              // {
19              //     gameObject.SetActive(true);
20              // }
21          }
22      }
23  }
24
```

The Inventory script keeps track of how many items the player has collected, adding one to the count each time, and sends this data to the inventory UI.

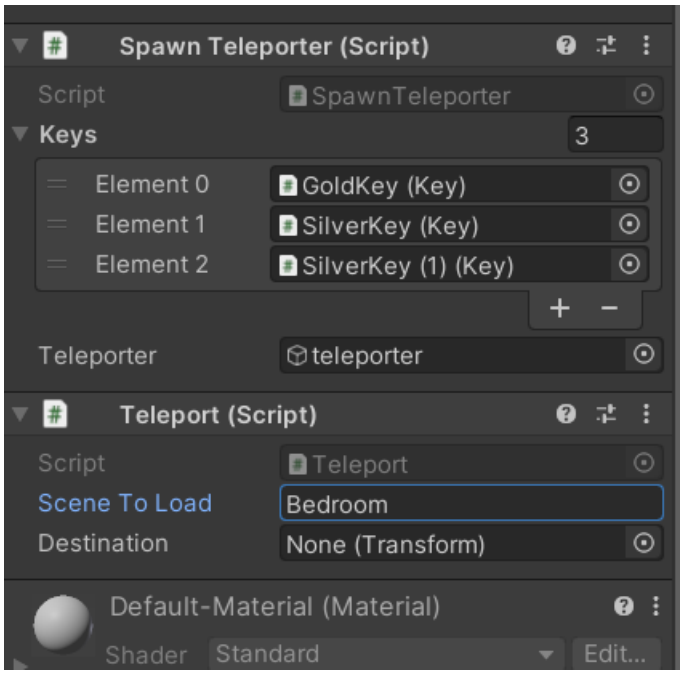
```
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Events;

public class Inventory : SpawnTeleporter
{
    public int NumberOfKeys { get; private set; }

    public UnityEvent<Inventory> OnKeyCollection;


    public void KeyCollected()
    {
        NumberOfKeys++;

        CheckKeys();

        OnKeyCollection.Invoke(this);
    }
}

```

Lastly, the InventoryUI script displays the number of keys collected to the player, by pulling the data from the Inventory script and converting it onto a string with Unity's TextMeshPro, in the top left on the screen.

```
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class InventoryUI : MonoBehaviour
{
    private TextMeshProUGUI keyText;
    void Start()
    {
        keyText = GetComponent<TextMeshProUGUI>();
    }

    public void UpdatedKeyText(Inventory inventory)
    {
        keyText.text = inventory.NumberOfKeys.ToString();
    }

}
```

5.8.4 Teleport + Script

I wanted the teleporter to only be activated when the allotted number of items were collected to allow the player to advance in the game. To do this I let the items as an array rather than individuals, this would make it much easier to set and change the number of items needed or the items themselves.



Next, once active, the script looks to see if the player has touched the teleporter and if so will teleport them to the next intended level. If the player has not yet interacted with the teleporter, then it will remain active and waiting until then, while simultaneously not allowing any others such as enemies to be able to activate it.

```
public class Teleport : MonoBehaviour
{
    public string sceneToLoad;
    public Transform destination; // Optional

    void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            SceneManager.LoadScene(sceneToLoad);

            // Move player to destination (optional)
            if (destination != null)
            {
                GameObject.FindGameObjectWithTag("Player").transform.position =
                destination.position;
            }
        }
```

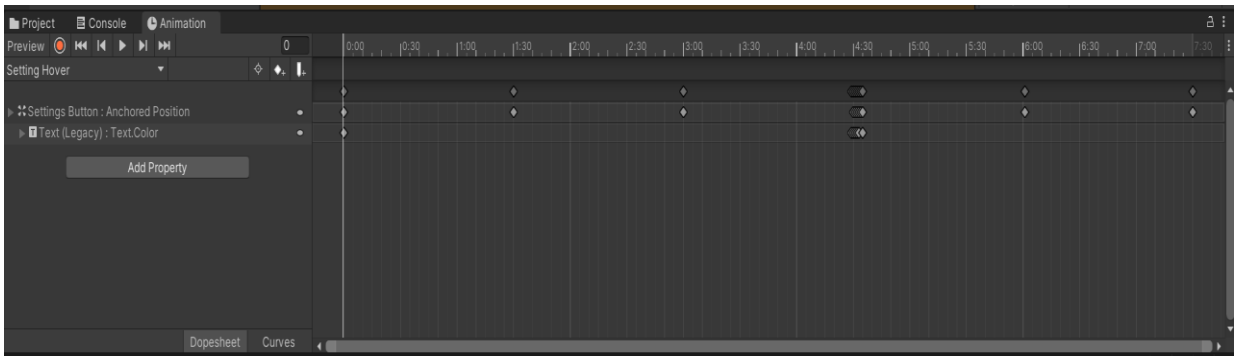
5.9 Sprint 6

5.9.1 Goal

The goal for this sprint was to design, program and animate the menu screens that would be used for the game. I decided to make three, being a Main start menu to introduce the game, a Pause menu where players could pause the level and have the option to quit the game or navigate to the Settings menu to adjust the music or sound effects volume.

5.9.2 Main Menu

The main menu displays the games title “Forgotten Fears” with buttons below allowing the player to start the game or navigate to the Settings menu. The background is an image of a cracked glitching screen with the text in the title to match. I animated some of the letters in the title to flash unusual colours while spelling out a secret message, a little detail I thought would be fun to add, along with animating the Start and Settings buttons glitch out on the screen, breaking away from their hovering pattern by using Unity’s Animator tool to create them.



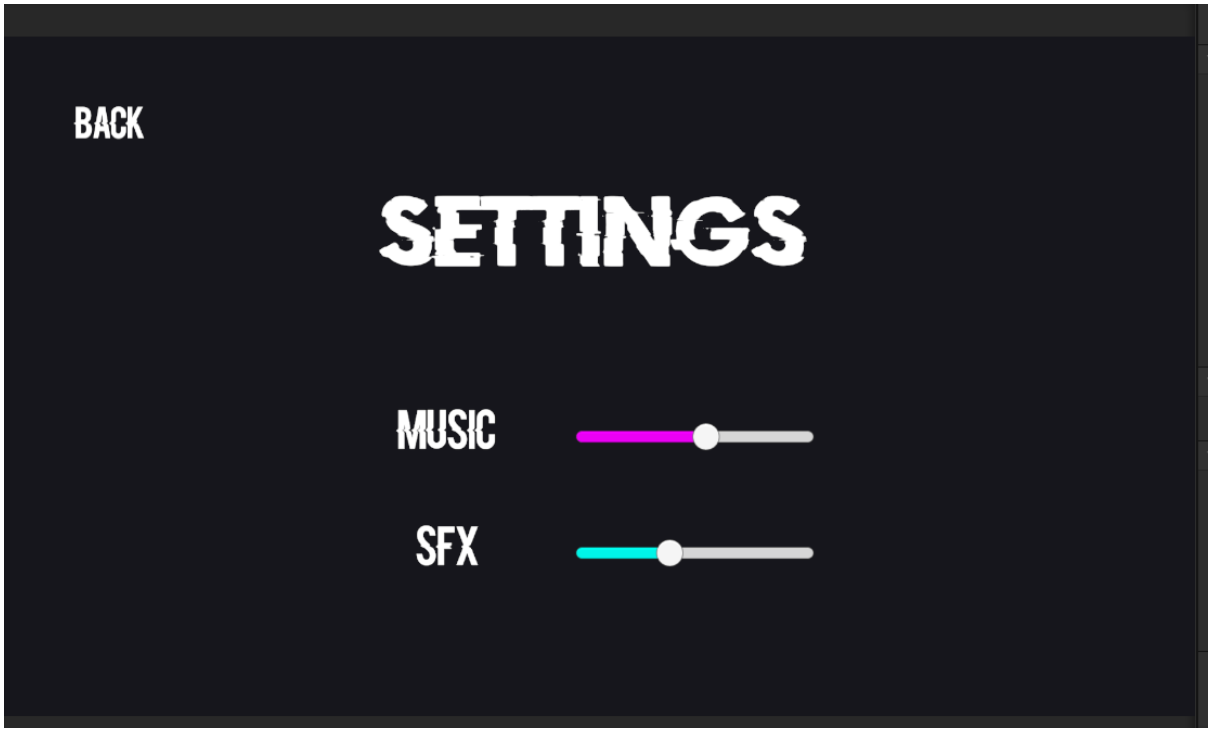
The MenuNav script simply allows the scene name to be set and changed to the desired level, using Unity's Scene Manager.

```
5
6   public class MenuNavScript : MonoBehaviour
7   {
8       public string levelName;
9
10      public void changeScene()
11      {
12          SceneManager.LoadScene(levelName);
13      }
14  }
```

5.9.3 Settings + Pause menu

The pause menu is kept simple, with the only options being to Return to the game, to navigate to the Settings menu or to Quit the game entirely, leaving the player at the main menu screen. The MenuNav script is used for both the Pause and Main menu along with the Back button in the Settings menu and was programmed to be interchangeable between menus to make it easier to switch between.

Similarly with the Settings menu, it only allows the player to adjust the volume of the background music and the sound effects, while letting them navigate back to the previous menu.



The VolumeSettings script is basic, with it setting the sliders to be interactive but is mostly overseen in Unity's Audio Manager.

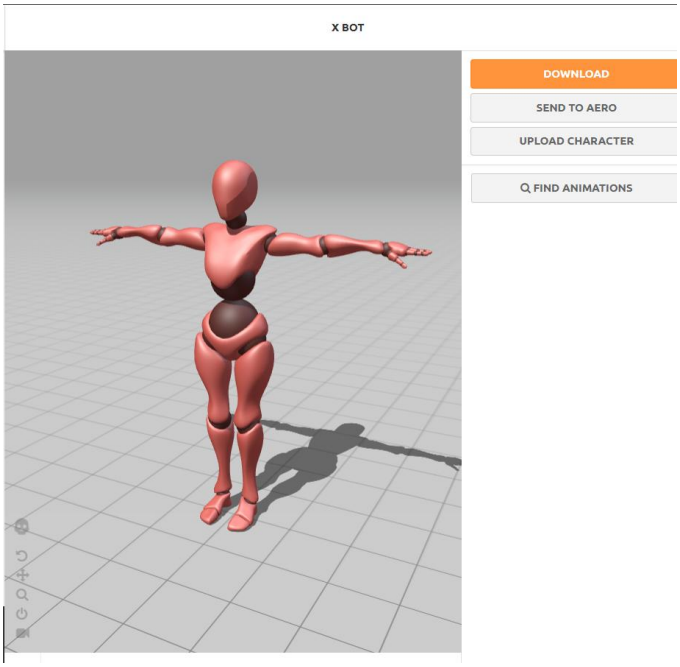
5.10 Sprint 7

5.10.1 Goal

The goal for this sprint is to gather suitable character models from Mixamo to be used for the player and enemies, rather than having them be cylinders as they previously were, and to rig up animated walk cycles for then again using pre-made Mixamo animations.

5.10.2 Mixamo Models + Animations

Since I had to scrap the idea of making my own character models and animations, I decided to use premade ones from Mixamo, a web-based service owned by Adobe that simplifies 3D character animation by offering pre-rigged characters, skeletal rigs, and a library of animations. I chose to have the player be depicted as female bot named Bot Y and for the enemies to be depicted as the male bots named Bot X.



Their animations were managed and rigged, using their premade ones from Mixamo, in Unity's Animator. Here, I tried to set up triggers between the animations so they would loop through a Jogging, Jumping, Landing and Idle animation for the player, I was unable to however get it to work properly. The way in which Mixamo's animations are packaged, for me at least, would not allow them to cycle through them at all, only displaying the players Idle animation. So, in the end I had to just set both the player and enemies to be in a constant running animation so that I could proceed with making the game.

5.11 Sprint 8

5.11.1 Goal

For this final sprint, apart from doing some minor fixes such as level layouts, enemy positions, and player inventory, I started to properly implement the sound effects and background music that I had picked out at the start of this project. I wanted to finish setting up the Settings menu so that it would interact with Unity's Audio Manger, adjust the volume music, and sound effects.

5.11.2 Error Fixing

I thought it best to try and fix any minor errors that while were not causing issues with the game but would have been better to resolve, such as the players inventory not correctly counting the items. I also wanted to try and get some different animations to work for the players walk and jump cycles before then trying to do the same with the enemies. Sadly, this proved to be a fruitless attempt, as I just could not figure out how to correctly rig them to trigger in Unity's animator.

5.11.3 Audio management + Sound Implementation

I stated by creating an AudioManger script that would hold all the sounds and music stored and control audio playback. I then set up two empty GameObjects "Music" and "SFX", each with a separate AudioSource component to oversee music and sound effects separately. The audio clips are then dragged and dropped into their respective slots labelled AudioClip and MusicClip in the Unity Inspector, under the "Music" and "SFX". I added a line of code that would automatically play the assigned background music when the game starts along with a Loop option. The sound effects are triggered by the player lands after jumping or when an enemy attacks by listening to the ParicalController.

```
using UnityEngine;

public class AudioManager : MonoBehaviour
{
    [Header("---------- Audio Source -----------")]
    [SerializeField] AudioSource musicSource;
    [SerializeField] AudioSource SFXSource;

    [Header("---------- Music Clips ----------")]
    public AudioClip MainMenu;
    public AudioClip Level1;
    public AudioClip Level2;
    public AudioClip Level3;
    public AudioClip DeathMusic;
    public AudioClip PauseMusic;

    [Header("--------- Audio Clips ----------")]
    public AudioClip Death;
    public AudioClip unlock;

}
```

5.12 Conclusion

This chapter accounts all the progress made in the eight sprints documented during this project with details about the various tasks that were done during those sprints. Each sprint lasted a week with the jobs that were completed or that were backlog recorded along with both screenshots and links

to tutorials used or to be used. Some of the necessary steps to making this game include programming movement, camera, and enemy AI scripts along with a player inventory and health system, to design and create various menu screens for the player use to adjust volume level or to exit the game. Level and environment design was crucial as they needed to be simple enough for the players to navigate without getting too lost on the objective while also having enough of a challenge to keep players engaged.

Sprint 1 details the initial creation of a test level that would be used to experiment and assess movement scripts and actions such as jumping, running, and crouching.

Sprint 2 details the fixing of outstanding errors with the player movement and third-person camera scripts, as well as the start of building the levels themselves. Here sound effects and background music was also gathered from sites like UppBeat.com and Pixabay.com to be implemented later in the project.

Sprint 3 details the initial plans to design and create original character designs to have been used in the game along with the rigging walk animations for the player and the enemies. It also details the researched into tutorials on how to do this along with the drawing up of designs for the models that unfortunately had to be abandoned.

Sprint 4 details creation of the players health and the generic enemies' behaviours, going into detail of how they operate and behave by detecting if the player is within range, along with the potential making of enemy variants and programming a boss level.

Sprint 5 details complete creation of the second level along with creating a trigger teleport system that can only be activated if the player gathers all the necessary collectables, as well as the collectables and player inventory system itself.

Sprint 6 details the designing, programming and animating of three menu screens that would be used for the game with there being a Main start menu to introduce the game, a Pause menu where players could pause the level and have the option to quit the game or navigate to the Settings menu to adjust the music or sound effects volume.

Sprint 7 details the gathering of suitable character models from Mixamo to be used for the player and enemies and to rig up animated walk cycles for then again using pre-made Mixamo animations.

Sprint 8 details the fixing of error ranging from minor like level layouts to attempting to fix major one such as the broken walk cycles of the player and enemies. This sprint also talks about how the audio management system was made and how the necessary sound effects were implemented and managed through the games Settings menu.

6 Testing

6.1 Introduction

The testing phase and user testing phase was entirely done by myself as I wanted testing more based on functionality to make sure the code and environment were coherent and functional. I spend some time making sure that the levels easy to understand, while fixing any layout issues and making sure that the collectables would have challenging placements, and to see what game features may need work after seeing them in action.

The users that performed the user testing consisted of friends and family outside of the college and a handful of classmates that were happy enough to give me some of their time to try find errors, bugs and just playtest the levels.

6.2 Functional Testing

Majority of the functionality testing was done by me as it was the most efficient way of making progress. This consisted mostly of me debugging code, fixing random or compiled errors I ran into or just reworking scripts to run more smoothly. However, when I was eventually stumped on some errors, I was able to get help form some of my classmates who were also working on games for their final project thankfully.

6.2.1 Discussion of Functional Testing Results

The results were as expected with most of the issues either being resolved within the first few tries, either by myself or with help from friends and classmates, or for some of the scripts / animators to be completely scrapped and reworked. An example being the constant issues I had with both the enemies AI and the players movement scripts, both scripts never working as intended resulting in constant errors that neither I nor friends could resolve so they were both re-written.

The original enemy AI script has the player being able to walk on top of and even phase through the enemy on occasion, along with the enemy getting stuck on random parts of the level. I tried debugging their pathfinding and hitbox to try and fix them getting stuck, but nothing seemed to work additionally, they would randomly speed up after touching the player so I thought it best to just start from nothing as I could not figure out the cause for it. The improved script wandered around the area fine and when the player was in range, they were able to follow them and had no issues colliding with the player.

As for the players movement, there was a persistent bug that would alter the players jump height and would make them jump at random intervals them holding down the spacebar along with the player having uncontrollable momentum when moving around causing them to run into everything, I ran into similar issues where I tried everything I could to try and fix the issues and thought best to start over.

6.3 User Testing

I found the user testing that I conducted was quite useful when it came to see how players would understand the control scheme, how they would react to the start level, how they would navigate the menus and how they experienced the game.

I got some good feedback when it came to the first level as it was the most played level due to its functionality, a lot of it was to do with the functionality of the game itself, that the enemies weren't colliding properly with the player, the teleport didn't trigger after all items were collected or even just going wild and trying to find faults or exploits in the levels and gameplay, feedback like that helped a lot.

I had positive feedback for the overall design and aesthetic of the game, which was nice, so it was good to know that people enjoyed the low poly, slightly quirky style of creepy I was going for.

6.4 Conclusion

This chapter details the testing process for the game through Functionality and User testing ,which was conducted with friends, family, and classmates to find errors, bugs, and playtest the levels.

Functionality testing involved debugging code and fixing errors, sometimes with help from classmates where significant issues were encountered with the enemy AI and player movement scripts, which were rewritten due to persistent and unresolvable bugs.

Additionally, User testing provided valuable feedback on player understanding of the controls, impressions to the initial level design, menu navigation, and overall game experience. Feedback highlighted issues like improper enemy collision and non-triggering teleports, which were resolved.

7 Project Management

7.1 Introduction

Looking back on the project, while facing many challenges from reworking and tweaking the initial idea for this project, failed attempts at modelling, to difficult and persistent scripting errors, I am both surprised and pleased with the outcome in terms of the sprints and result. Throughout the course of this project, I have had mixed emotions when it came to the workload that came with completing this project to a satisfactory standard along with its tasks, as I found myself struggling as time went on.

Due to planning ahead I was also able to keep mostly up to date with our sprints and goals that were set out for the duration of them, whether it was research, design or implementation, and if I felt that I wouldn’t be able to reach any of those set goals, I made it a priority to try make up for it in the next sprint while also working on work that was already set.

7.2 Project Phases

7.2.1 Proposal

The original idea for the project was to be a 3D survival horror video game with a dark atmosphere to build suspense. It was to be primarily built in Unity with the assets and most of the animations being done in Blender with some being used from the Unity Asset Store as well.

The game was to be loosely based off Little Nightmares and My Friendly Neighbourhood where the players are thrown into the thick of it with little or nothing to defend themselves where one hit will be the end of their journey. For the design I wanted to do it in the style of the VR game Job Simulator, a game with a low poly style which I thought would work to my advantage when trying to design / create my own assets.

Looking back on the initial proposal I regret not starting smaller and scrutinising my own abilities a bit more. While the idea was doable in theory, realistically I should have known my abilities in Blender we nowhere near the level they would have had to have been to make half of what was planned for this project possible.

7.2.2 Requirements

Figuring out the requirements that the game needed was a simple task when broken down into its Functional and Non-Functional requirements. The functional requirements were figured out by breaking down what my research and example games had in common, I tried to keep it simple as I could, bearing in mind my capabilities and the timeframe I had to work in, to make sure not to overcomplicate anything as I moved onto bigger things that I would have to change in the game.

The Non-Functional requirements took a little more research to fully realise what I would need, a lot of design research was done for understanding how the levels should be structured to give players a challenge while making it open enough to evade enemies and obstacles, I also did some research into menu designs to make sure they were very understandable and easy to navigate.

There were some things on the requirements list that we were not able to complete as the priority was to make sure that each requirement was fully completed to a standard that I was happy with, that means that stuff like the boss level + AI was not implemented but there were also plans that were simply just not feasible from the beginning.

7.2.3 Design

The initial research into the design was one of the most enjoyable parts of this project because it got me excited about producing it. Most of my inspiration for this project came from Little Nightmares 1 & 2, both thriller horror games about being thrown into a unknown and hostile environment, you must use your size and surroundings to your advantage while evading enemies and monsters out to kill you while your only weapon being your wits. I also investigated the idea of basing the theme of the game around *Divine Machinery* - "*a contemporary aesthetic which blends themes of technology with religion, focusing on the artificial and associating it with the divine. Commonly utilized imagery*

includes wires, computer monitors, transmission towers, hardware components, religious iconography, and angels."

When it came to the design process, I wanted to give the game a low-poly design to be in keep with an Indie look, I thought that this would lend itself to being easier to create / find assets for while being a slight challenge in trying to make it creepy. Still that my initial ideas for the look of the game are what caused me most of the issues, causing me to waste too much time in trying to relearn Blender when I had forgone the idea from the start I could have made much better progress.

7.2.4 Implementation

Overall, I found the implementation phase of the project to be simple. It taught me a lot about code structure and how to properly overhaul and or rework features of a game and how to move files from different unity projects properly.

7.2.5 Testing

I also feel that I could have tried to get more user testing done throughout the course of this project. However, given the number of changes the needed to be made and issues I ran into, it was a safe decision to focus on making the game completely functional rather than must constantly explain to the testers what should / will soon be working.

7.3 SCRUM Methodology

I feel like the SCRUM process worked quite well, I was trying to be careful not to get overwhelmed with tasks to make sure that each goal was achievable, along with prioritising a functional outcome for the project.

At the beginning, the sprints seemed very manageable but towards the middle of the project when the bulk of important work was to be done, all while dealing with major changes to plans and figuring out alternative solutions did feel quite overwhelming at points, I felt it important in having breaks from the project to make sure that mental and physical health wasn't being affected too heavily.

Overall, the idea of small and digestible tasks works well but when balancing this with backlog and other changes it can easily become overwhelming.

7.4 Project Management Tools

7.4.1 GitHub / OneDrive

While I should have used GitHub to properly keep track of my progress with this project, I felt it safer to use OneDrive to keep track of the project iterations as with previous experiences GitHub failed to save important assets and files because of the Git Ignore system, which I had no experience or time

in trying to modifying. So instead of risking losing valuable progress and time with this project, I opted to use OneDrive instead.

7.5 Reflection

7.5.1 Your views on the project

I feel that this project was average in comparison to what it could have been had I the knowledge, I feel that I was far too overzealous with my ideas and capabilities for much of this project and as a result it led to a lot of unnecessary stress and anxiety. However, despite all this I am pleased with the results none the less.

7.5.2 Completing a large software development project

I feel like I have learned a lot in terms of project management, when it came to this project, trying to keep on top of tasks while testing each implementation to make sure that everything was working smoothly was a tough task, but one that I was up for.

Time management was an extremely important part of the project, making sure you were not spending too much time on something was a very important part of making sure the project schedule was on track, I found myself taking breaks from big implementations or waiting until I had other systems finished before taking on big tasks as a new that it would take a very long time to fully implement.

I have learned a lot about how the Unity engine works and have developed my skills in Unity a lot more thanks to this project, I think running into problems and searching for ways to fix them were what led me to have a better understanding of how Unity works and how the scripting works within the engine.

7.5.3 Working with a supervisor

Working with my supervisor was fine as expected. We met regularly to discuss progress in the project as a whole and in the individual sprints during the project, along with giving plans for future work and goals.

I was also given advice about scaling down some ideas for the project when it was sensed that they would be putting additional pressure on myself and, given the time constraints, the projects result.

7.5.4 Technical skills

Thanks to the help of YouTube tutorials, Unity forum posts and the Unity documentation, I learned a bit more about the Unity Engine and how it operates. Coming into this project, I had very little experience and knowledge on the scripting side of Unity as I am more interested in the designing and animation side of video games, and this entire project was trial and error for me, doing as much research as I could into the ins and outs of the systems and methods that could potentially be used

for features to improve the game, scrolling through countless forum posts on how to fix problems to get it to a playable state.

I have further developed my understand of Unity's animator and feel that I could confidently try messing around with it to see if I could create my own animations.

Coming to the end of the project, I feel that while I was getting a better understanding of the language and the engine, I still had plenty left to learn especially if I wanted to try my hand at adding some of the more complex ideas I initially had in mind for this game.

7.5.5 Further competencies and skills

As previously mentioned, I feel that had I better knowledge and understanding of how to work with Blender and how to rig animations, I would have been able to produce a better-looking game along with elevating the game to a higher level. I feel that having a better understanding of enemy AI and pathfinding would be helpful in the future, as I would have loved to add the originally planned boss level at the end of the game to make it feel like more of a complete game and to add a bit of a challenge / buildup for the players.

7.6 Conclusion

This chapter reflects on the project management and progress of the game's development. Despite challenges like idea revisions, modelling difficulties, and scripting errors, planning helped maintain forward progress, with efforts made to catch up on missed goals in subsequent sprints.

This chapter investigates the creative decisions that were made in the initial Proposal and Design phase of this project, along with how the project was managed. Additionally, the Testing of Implementations to the project were monitored and the results found from user testing were recorded and corrected, if needed.

Lastly, this chapter gives my overall outlook and insight into my experience throughout this project along with what I have learned and would like to add in future development of this project.

8 Conclusion

8.1 Background

This chapter talks about the initial requirements for developing a thriller horror game by referencing existing successful horror games like Little Nightmares and Faith: The Unholy Trinity, highlighting their effective mechanics such as player vulnerability, environmental interaction, eerie atmosphere, and sound design. This chapter also delves into the initial project proposal and documents what changes were / needed to be made along the way.

8.2 Technologies

In this segment it goes into detail about what technologies or applications were used in the development of this project and why they were chosen. It also touches on how Unity and Visual Studio were used for the bulk of this games designing, development and coding as well as how blender was planned to be used for the designing and cinematic side of this project.

8.3 Research

This segment touches on how a game must be able to keep players engaged while also keeping them challenged whether that be with the gameplay itself or with the idea of completing the game to its entirety. These points have been proven when looking at the most popular games in each category and what mechanics they offer, as they adhere to the rules of creating horror games.

8.4 Design

This chapter looks at how the game's core concept revolves around the "Devine Machinery" aesthetic, blending technology and religion with the narrative involving corrupted retro technology, personified as twisted beings worshipping a false god ("The Streamer"). it also talks about why Unity was chosen compared to other gaming engines along with how originally, Blender was intended for 3D modelling and animation but had to be excluded to maintain the project timeline. Lastly, this chapter looks at the environment and level design decisions that were made to be in keeping with the games original storyboard, while touching on ideas that had to be cut.

8.5 Implementation

This chapter accounts all the progress made in the eight sprints documented during this project with details about the various tasks that were done during those sprints. Each sprint lasted a week with the jobs that were completed or that were backlog recorded along with both screenshots and links to tutorials used or to be used. Some of the necessary steps to making this game include programming movement, camera, and enemy AI scripts along with a player inventory and health system, to design and create various menu screens for the player use to adjust volume level or to exit the game. Level and environment design was crucial as they needed to be simple enough for the players to navigate without getting too lost on the objective while also having enough of a challenge to keep players engaged.

8.6 Testing

This chapter details the testing process for the game through Functionality and User testing ,which was conducted with friends, family, and classmates to find errors, bugs, and playtest the levels. I also gives insight to the feedback that was given and used to improve the overall experience for players.

8.7 Result

This project resulted in the creation of a 3D thriller platformer based around the aesthetic of Divine Machinery, while being inspired by other horror games such as Little Nightmares and My Friendly Neighbour. The game “Forgotten Fears” contains two levels for the player to explore tasked with collecting key items needed to escape while trying to about obstacles and enemies. Players are also able to Pause the game and adjust the audio settings through the Settings menu.

8.8 Project Management

This chapter looks at the creative decisions that were made in the initial Proposal and Design phase of this project, along with how the project was managed with addition of the Testing of Implementations to the project were monitored and the results found from user testing. Lastly, this chapter gives an overall outlook and insight into the experience throughout this project along with what was learned and would have been added to further development this project.

8.9 What I Learned

I have learned a lot about how the Unity engine works and have developed my skills in Unity a lot more thanks to this project, I think running into problems and searching for ways to fix them were what led me to have a better understanding of how Unity works and how the scripting works within the engine.

8.10 Future Development

As previously mentioned, I feel that had I better knowledge and understanding of how to work with Blender and how to rig animations, I would have been able to produce a better-looking game along with elevating the game to a higher level.

References

Aalborg Universitet, Grimshaw, M. N., University of Bolton, & Grimshaw, M. (2009). The audio Uncanny Valley: Sound, fear and the horror game. In *Audio Mostly 2009*. https://vbn.aau.dk/ws/portalfiles/portal/61573698/audioUncannyValley_MG.pdf

Agnello, A. (2019, September 20). *Body of Fear: How Your Body Keeps you Playing Horror Games - IGN*. IGN. https://www.ign.com/articles/2013/09/09/body-of-fear-how-your-body-keeps-you-playing-horror-games

Analysts: In 2024, indie games on Steam for the first time nearly matched the revenue of AA and AAA games. (2024, October 15). app2top.com. https://app2top.com/news/analysts-in-2024-indie-games-on-steam-for-the-first-time-nearly-matched-the-revenue-of-aa-and-aaa-games-272517.html#:~:text=2024%2013%3A02-,Analysts%3A%20In%202024%2C%20indie%20games%20on%20Steam%20for%20the%20first,of%20AA%20and%20AAA%20games

Boehm, A. (2024, September 23). *[Review] 'Mouthwashing' Is a Surreal Trip Into High Tension Horror*. Bloody Disgusting! https://bloody-disgusting.com/video-games/3832240/review-mouthwashing-is-a-surreal-trip-into-high-tension-horror/

Buglass, L. (2024, December 1). *Mouthwashing review: this low-poly horror masterpiece will haunt me for a long time*. TechRadar. https://www.techradar.com/gaming/mouthwashing-review

By, W. (n.d.). *5 timeless principles of good game design | Evercast Blog*. Evercast. https://www.evercast.us/blog/game-design-principles#:~:text=Game%20design%20principles%20focus%20on,them%20coming%20back%20for%20more.

Chen, R. (n.d.). *Investigating the Audio-Visual Psychological Effects in a Horror Game - ProQuest*. https://www.proquest.com/openview/1cfa02578b0300b9cb73a922f28aedab/1?pq-origsite=gscholar&cbl=18750&diss=y

Couture, J. (2024, October 19). *Exploring the subversive childhood horror of Choo Choo Charles*. https://www.gamedeveloper.com/design/exploring-the-subversive-childhood-horror-of-choo-choo-charles

Di Placido, D. (2023, November 6). How 'Five Nights At Freddy's' Conquered YouTube, And The Box Office. *Forbes*. https://www.forbes.com/sites/danidiplacido/2023/10/31/the-five-nights-at-freddys-phenomenon-explained/

Epps, D. (2023a, November 1). Why do we love horror games? Psychologists explain our morbid curiosity. *Digital Trends*. https://www.digitaltrends.com/gaming/why-do-we-love-horror-games-analysis/#:~:text=There%20are%20several%20studies%20on,much%20anticipation%20built%20into%20them.

Epps, D. (2023b, November 1). Why do we love horror games? Psychologists explain our morbid curiosity. *Digital Trends*. https://www.digitaltrends.com/gaming/why-do-we-love-horror-games-analysis/#:~:text=There%20are%20several%20studies%20on,much%20anticipation%20built%20into%20them.

Epps, D. (2023c, November 1). Why do we love horror games? Psychologists explain our morbid curiosity. *Digital Trends*. https://www.digitaltrends.com/gaming/why-do-we-love-horror-games-analysis/#:~:text=There%20are%20several%20studies%20on,much%20anticipation%20built%20into%20them.

Houghton, D. (2016, October 31). *P.T. is still the purest horror game around, and one of the smartest on PS4*. Gamesradar. https://www.gamesradar.com/why-pt-first-real-horror-game-years-and-best-game-ps4/

https://brightlightsfilm.com/why-do-we-crave-horror-evolutionary-psychology-and-viewer-response-to-horror-films/

https://blackshellmedia.com/2016/06/13/redefining-genre-secret-five-nights-freddys-success/

Indie games come close to AA/AAA games in revenue on Steam for first time ever, grossing $4 billion in 2024 so far. (2024a, October 16). WN Hub. https://wnhub.io/news/stores-and-publishing/item-45826

Indie games come close to AA/AAA games in revenue on Steam for first time ever, grossing $4 billion in 2024 so far. (2024b, October 16). WN Hub. https://wnhub.io/news/stores-and-publishing/item-45826

Jones, J. (n.d.). *Green Man Gaming Blog*. https://www.greenmangaming.com/blog/why-until-dawn-deserves-its-remake-treatment/#:~:text=Another%20reason%20for%20Until%20Dawn's,their%20choices%20in%20key%20situations.

Little Nightmares (Video Game 2017) - User reviews - IMDb. (n.d.). IMDb. https://www.imdb.com/title/tt6795176/reviews/

Liu, Y. (2023, December 9). *Fear and love: dig deeper into the potential reasons why players like horror games*. https://www.gamedeveloper.com/design/fear-and-love-dig-deeper-into-the-potential-reasons-why-players-like-horror-games

Martin, G. N. (2019). (Why) Do You Like Scary Movies? A Review of the Empirical Research on Psychological Responses to Horror Films. *Frontiers in Psychology*, *10*. https://doi.org/10.3389/fpsyg.2019.02298

MORTIS 101: "FAITH's" Horror Design Toolkit. (n.d.). https://www.gdcvault.com/play/1027195/MORTIS-101-FAITH-s-Horror

Ningalei, N., & Wöhrman, S. (2018). *The Impact of Sound on Player Experience*. https://www.diva-portal.org/smash/get/diva2:1480010/FULLTEXT01.pdf

Npc, J. T. (2024, April 17). Gaming News: Why Are Gamers Losing Their Enthusiasm for AAA Releases? *The Portal*. https://www.zleague.gg/theportal/gaming-news-why-are-gamers-losing-their-enthusiasm-for-aaa-releases/#:~:text=With%20AAA%20titles%20becoming%20increasingly,back%20seat%20to%20corporate%20interests.

Outlast (Video Game 2013) - User reviews - IMDb. (n.d.). IMDb. https://www.imdb.com/title/tt2984660/reviews/

Purslow, M. (2024, October 7). *Silent Hill 2 Is Utterly Miserable (and That's Why It's Great) - IGN*. IGN. https://www.ign.com/articles/silent-hill-2-is-utterly-miserable-and-thats-why-its-great

Reed, A. (2014, October 14). *The Evil Within review*. Gamesradar. https://www.gamesradar.com/the-evil-within-review/

Reynolds, W. (2017, April 21). Little Nightmares review. *Polygon*. https://www.polygon.com/2017/4/21/15381324/little-nightmares-review

Rose, V. (2017, October 22). Doki Doki Literature Club is an uncontrollably horrific visual novel. *Polygon*. https://www.polygon.com/2017/10/22/16512204/doki-doki-literature-club-pc-explained

Sabbagh, M. (2023, December 8). *The key to designing an effectively tense survival horror experience*. https://www.gamedeveloper.com/design/the-key-to-designing-an-effectively-tense-survival-horror-experience

Silent Hill 2 Review - In My Restless Dreams, I see That Town. (2024, October 4). Wccftech. https://wccftech.com/review/silent-hill-2-restless-dreams-town/

Sliva, M. (2017, December 7). *Is Outlast the Scariest Game Ever? - IGN*. IGN. https://www.ign.com/articles/2013/03/26/is-outlast-the-scariest-game-ever

The Evil Within Review - IGN. (2014, October 12). IGN. https://www.ign.com/articles/2014/10/16/the-evil-within-review

Tomé, F., Pires, A., Jiskrová, A., Saial, A., & Campos, P. F. (2024). "I Found it Cathartic": Exploring Empathy and Mental Health Awareness in Psychological Horror Video Games. *Proceedings of the ACM on Human-Computer Interaction*, *8*(CHI PLAY), 1–21. https://doi.org/10.1145/3677083

Until Dawn (Video Game 2015) - User reviews - IMDb. (n.d.). IMDb. https://www.imdb.com/title/tt2742544/reviews/

Vorobeva, E. (2024). *Building atmosphere in a horror game through environment design* (p. 43) [Bachelor's thesis]. https://www.theseus.fi/bitstream/handle/10024/862614/ekaterina_vorobeva.pdf?sequence=2&isAllowed=y

Zhang, Z. (2022). ANALYSIS OF THE DESIGN AESTHETICS AND PLAYER EMOTIONS OF HORROR GAMES. In *Master Degree Project in Informatics*. https://www.diva-portal.org/smash/get/diva2:1677001/FULLTEXT01.pdf