Contents

1.0. Introduction 2
1.1. Inspiration 2
1.2. Structure 2
1.3. Playability and style 2
1.4. Game Engine 2
1.5. External sources 2
1.6. Report Overview 2
2.0. Professional considerations 3
2.1. BCS professional considerations 3
2.1.1. Safety 3
2.1.2. Third-party rights 3
2.1.3. Equality 3
2.1.4. Ability level 3
2.2. Ethical review 3
2.2.1. User testing 3
2.2.2. Data protection 3
3.0. Background research 4
3.1. Existing titles in the genre 4
3.1.1. New Super Lucky's Tale (Playful, 2019) 4
3.1.2. Yooka-Laylee (Playtonic, 2017) 8
3.1.3. Existing titles conclusion 11
3.2. Game engine comparisons 12
3.3. Other software 13
3.3.1. 3D Modelling 13
3.3.2. Photo Editing 13
3.3.3. Animation 13
4.0. Requirements Specification 14
4.1. Storyline 14
4.2. Character controls and abilities 14
4.3. Gameplay flow and mechanics 14
4.4. Level design 15
4.5. Characters 15
4.6. Project plan 15
5.0. Game Design 16
5.1. Main Character Design 16
5.2. Level Design 16
5.2.1. Level Layouts 17
5.3. Enemy Design 20
5.3.1. Enemy Types 20
5.3.2. Level Enemy Variants 20
6.0. Game Development 26
6.1. Main Character development 26
6.1.1. Modelling 26
6.1.2. Rigging and animation 30
6.1.3. Player Health 33
6.1.4. Character movement 33
6.1.5. Player abilities 33
6.1.6. Shooting 34
6.2. Level Development 35
6.2.1. Base Level (Home) 35
6.2.2. Level One (Deserted Dunes) 36
6.2.3. Level Two (Robot Wars) 37
6.3. Level Items Development 38
6.3.1. Collectables 38
6.3.2. Portals 39

6.3.3. Explodable Rock 39
6.3.4. Signposts 40
6.3.5. Moving Platforms 41
6.3.6. Lasers 42
6.3.7. Puzzles 43
6.4. Enemy Development 44
6.4.1. Modelling 44
6.4.2. Base Enemy Implementation 47
6.4.3. Line Of Sight 47
6.4.4. Enemy script variations 48
6.4.5. Scorpion 48
6.4.6. Cactus 48
6.4.7. Level One Boss – (Doozy) (Adobe, 2021) 48
6.4.8. Robot 48
6.4.9. Turret 48
6.4.10. Large Turret 48
6.4.11. Level Two Boss (YBot) (Adobe, 2021) 48
6.5. Game Saving 49
6.6. UI 50
6.6.1. Font 50
6.6.2. Main menu 50
6.6.3. In-Game Display 52
6.7. Audio 54
6.7.1. Audio Mixer 54
6.7.2. Background Music 54
6.7.3. Sound Effects 54
6.7.4. Audio Used 54
7.0. Playtesting 56
7.1. Playtest one – 16th March 2022 56
7.1.1. Questionnaire responses 56
7.1.2. Problems noted / bugs 62
7.1.3. Changes implemented 62
7.2. Playtest two – 10th April 2022 63
7.2.1. Questionnaire responses 63
7.2.2. Problems noted / bugs 67
7.2.3. Changes implemented 67
8.0. Future work 68
9.0. References 69
10.0. Appendix 73
Appendix A User testing compliance form 73
Appendix B Initial project proposal 78
Appendix C Project Gantt Chart 81

1.0. Introduction

For this project, I aim to create a 3D platform game based around the characters Bow-Bow the dog and Bum-Bum the cat, inspired by my family pets.

1.1. Inspiration

The game is intended to be based around the common premise of the main characters visiting different levels/worlds and collecting items that have been stolen/gone missing. Whilst facing various challenges and puzzles along the way. Although there have been many games of this genre created in the past, such as Spyro the Dragon (Activision, 2020), Croc Legend of the Gobbo's (Argonaut, 2000) and Banjo-Kazooie (Rare, 1998), there are not so many being developed in recent years and as such, this will be a good genre to develop a game for as there is a need for this genre in the modern market.

More recent games developed in this genre include Yooka-Laylee (Playtonic, 2017) and New Super Lucky's Tale (Playful, 2019), which are intended to be used as background research for this project.

1.2. Structure

The main structure of the game will comprise the main hub level containing portals to various levels. Each of these levels will be based upon a different theme (i.e., sci-fi, prehistoric, wild west) and will each contain its mini storyline. Each level will contain the main story collectable, as well as various level collectables that will need to be collected to complete the level/game 100%.

I also plan to implement a skill tree or shop system to acquire different abilities, which can be used to access new areas in both the hub level and previously completed levels. This will mean that the player will need to return to previous levels to collect the remaining collectables.

1.3. Playability and style

The visual style of the game will be quite cartoon-like to appeal to a younger audience. As the project is targeting a younger audience, the final product will need to be easy enough that children can play without issue but still provide enough of a challenge and storyline to keep players of all ages engaged.

1.4. Game Engine

The intention is to use Unity Engine (Unity, 2021) to create this project since I am personally more proficient with creating games using this game engine than with Unreal Engine (Epic, 2021), which are the two main game engines in use today. This will aid in meeting the requirement for ability levels stated in chapter 2.1.5.

Additionally, the visual style of a game created using Unity Engine (Unity, 2021) is more closely matched to the style of the game that is intended to be produced.

1.5. External sources

- **Source code –** The source code for the project can be found here.
- **Playable Build –** The playable build of the project can be found here
- **Documentation –** Full documentation and Class diagrams for the project can be found here. Throughout the report and scripts referenced will be hyperlinked to the relevant page within the documentation for reference.

1.6. Report Overview

The remainder of this report is set out as follows. Chapter 2 outlines the relevant professional issues regarding the project, including safety issues and how it conforms to the BCS Code of Conduct. Chapter 3 covers background research for existing titles in the target genre, options for the game engine, as well as software options for various aspects of the development of this project. An overview of the initial requirements of the project, as well as a proposed plan for development, is contained in Chapter 4. Chapter 5 contains an overview of the design of the project. Chapter 6 contains detailed information about the implementation of the features within the project. Chapter 7 outlines the playtesting conducted, and Chapter 8 contains details of future work planned for the project. All references and appendices for the report are contained in chapters 9 and 10.

2.0. Professional considerations

2.1. BCS professional considerations

The project will conform to the code of conduct set out by the BCS (BCS, 2021).

2.1.1. Safety

This section covers any relevant safety concerns to be considered for the project.

2.1.1.1. Children

As the game will be aimed at a younger audience, it must not contain any profanities, nudity, graphic violence, or scary scenes. The game will aim to conform to a PEGI rating of 7 or below.

2.1.1.2. Flashing images

As well as avoiding using strobe effects, there will be a disclaimer when the game loads with an epilepsy warning to make players aware of any flashing images and include advice to stop playing if feeling unwell.

2.1.1.3. Screen time and breaks

As video games are quite an addictive activity, the project will also include a warning at the start advising the player to take regular breaks when playing.

2.1.2. Third-party rights

Any third-party assets, sounds and material used in the creation of the game will be referenced and credited as appropriate both within the final report and within the game's credits screen.

2.1.3. Equality

The project will not make any reference to sexual orientation, nationality, colour, race, ethnic origin, religion, or disability.

2.1.4. Ability level

All planned features will be within my ability level at the time of their creation.

2.2. Ethical review

The project will consider the following ethical issues during development.

2.2.1. User testing

Although the game will be aimed at a younger audience, all testing will be conducted with candidates of age 18 and over following the user testing compliance form (see Appendix A).

2.2.2. Data protection

All testing will be conducted anonymously, and no personal data will be collected or stored. Additionally, there will be no online leader boards, or any other online login required to play the game and as such, no personal data will be collected or stored to play the game.

3.0. Background research

As mentioned in *1.1. Inspiration*, the project intends to analyse two more recent games in the 3D platformer genre to look at the story elements as well as general game mechanics and level design.

3.1. Existing titles in the genre

The two titles that have been chosen are New Super Lucky's Tale (Playful, 2019) and Yooka-Laylee (Playtonic, 2017).

3.1.1. New Super Lucky's Tale (Playful, 2019)

3.1.1.1. Storyline

In this title's story, you play Lucky Swiftail, a fox character living in the fictional world of Foxington, who embarks on a journey through various worlds to escape the book of ages. When the main antagonist, Jinx attempts to steal the book of ages it pulls Lucky, Jinx, and the other enemies in the game into itself. The story then revolves around Lucky collecting the missing pages from the book to open portals, complete levels, defeat Jinx and his minions, and return home to Foxington.

3.1.1.2. Character controls and abilities

The character is controlled using a third-person view and has quite basic controls and abilities. Lucky can jump, double jump and has a simple spin attack. He can also jump on certain enemies to kill them. The standout ability in this title is Lucky's ability to dive into the ground in certain areas and move around under the ground. This is used to great effect in puzzles and to access certain areas within the levels. It can also be used as an additional attack to stun enemies.

3.1.1.3. Gameplay flow and mechanics

The game is divided into six main hub levels, each one with a selection of themed levels and one boss fight level. The boss fight can only be accessed once the player has collected enough pages within that world. Each level contains four pages to collect to complete it 100%.

These four pages are awarded for

- Completing the levels sub-story and reaching the end of the level.
- Collecting three hundred coins in the level.
- Collecting the L, U, C, K, Y letters hidden throughout each level
- Collecting a hidden page within the level, usually by completing a mini-game.

Once the player has collected enough pages in each hub world, then they can enter the final boss fight with Jinx to complete the game.

3.1.1.4. Level design

The six main hub levels in this title are as follows.

- **Sky Castle** – a generic cartoon world revolving around helping a group of Golems as shown in figure 3.1 below.

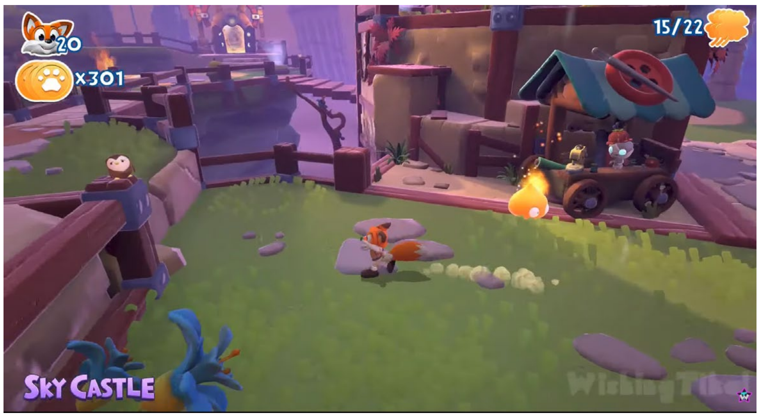


Figure 3.1 – Screenshot of the Sky Castle hub level

- **Veggie Village** – a country-themed world where you aim to help a group of worms as shown in figure 3.2 below.

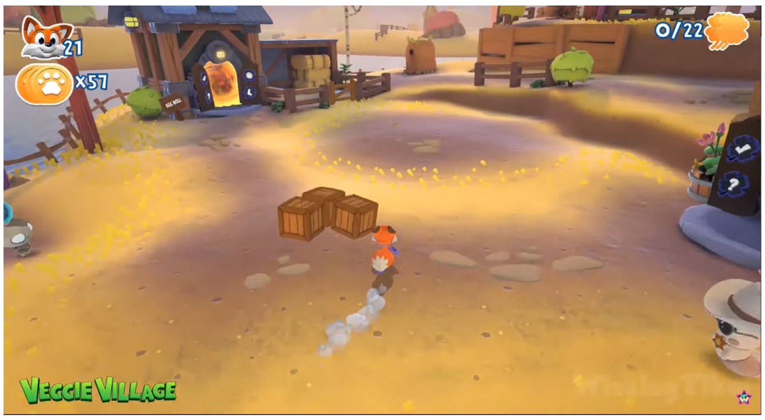


Figure 3.2 – Screenshot of the Veggie Village hub level

- **Wrestful Retreat** – a prehistoric themed world where you assist a group of wrestling Yetis as shown in figure 3.3 below.

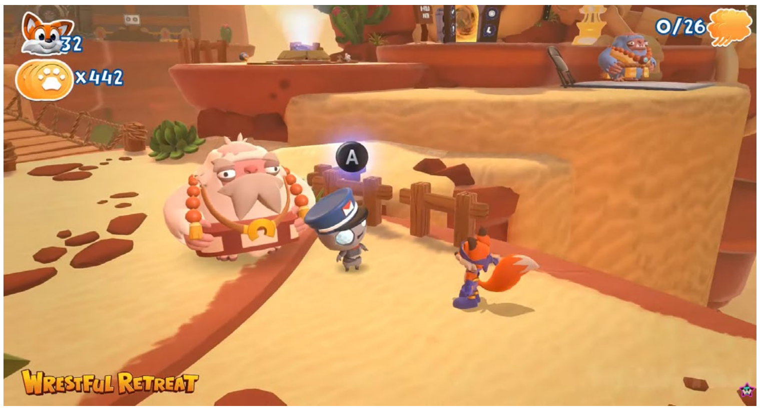


Figure 3.3 – Screenshot of the Wrestful Retreat hub level

- **Gilly Island** – A tropical island world inhabited by dancing fish as shown in figure 3.4 below.

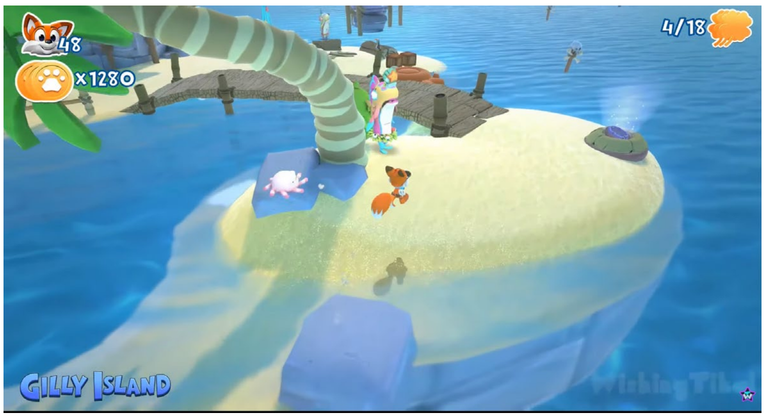


Figure 3.4 – Screenshot of the Gilly Island hub level

- **Hauntingham** – an eerie themed world where the friendly characters are ghosts as shown in figure 3.5 below.

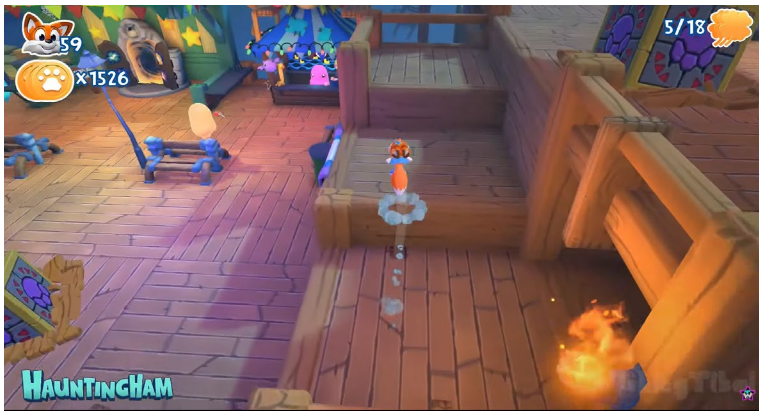


Figure 3.5 – Screenshot of the Hauntingham hub level

- **Foxington** – Lucky's homeworld is shown in figure 3.6 below.

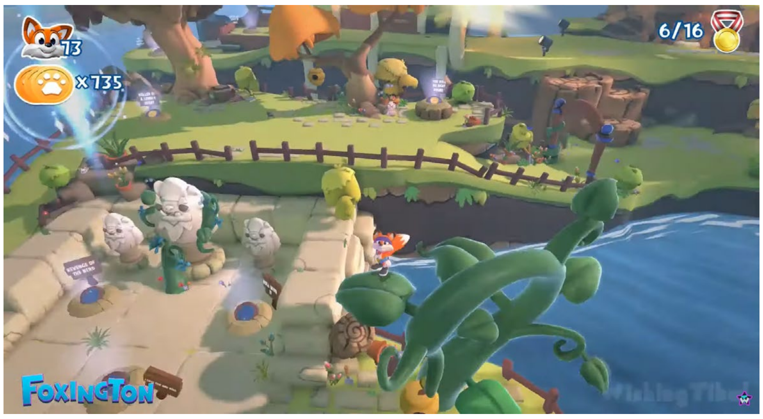


Figure 3.6 – Screenshot of the Foxington hub level

3.1.1.4.1. Level Varieties

Most of the levels are constructed in a 3D manner and can be explored freely. Certain levels are fixed to a 2D side-scrolling view and some of these have a constantly moving camera which adds to the challenge of collecting each of that level's pages.

3.1.1.4.2. Mechanics

The levels use many different mechanics to challenge the player including moving platforms, hidden areas, timed collectable runs to collect LUCKY letters or hidden pages, and secret tasks to collect LUCKY letters.

3.1.1.4.3. Enemies

Each level has its variety of enemies, each with its abilities and methods required to kill them. For example, certain enemies must be stunned using Lucky's burrowing ability before they can be killed.

3.1.1.5. Mini games

Each hub world also contains a selection of mini-games, which when completed award the player with an extra page.

These take two forms,

- **Sliding Puzzles** where you must move the statues in the correct order to place the Lucky statues on the highlighted spots as shown in figure 3.7 below.

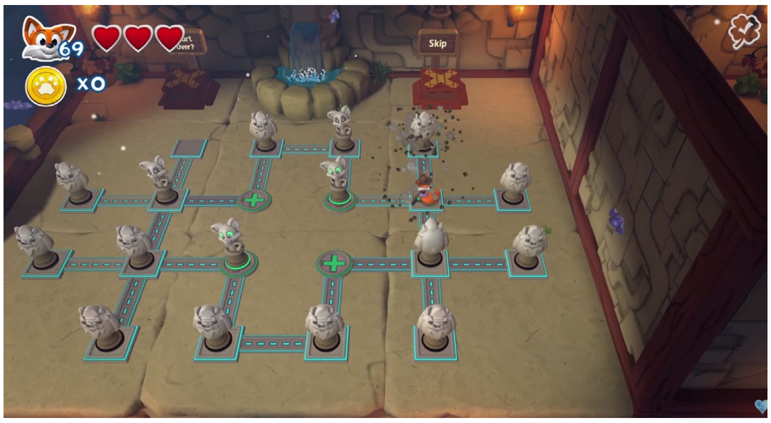


Figure 3.7 – Screenshot of the sliding statues mini-game

- **Rolling ball puzzles** where Lucky is placed in a ball, and you must tilt a maze to collect all the coins and reach the end platform as shown in figure 3.8 below.

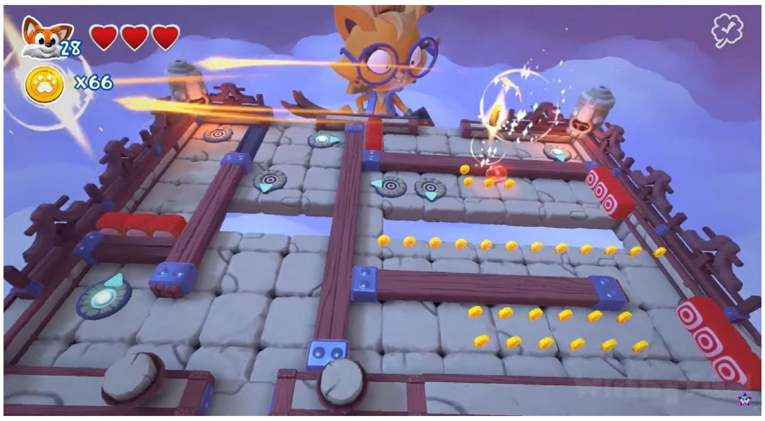


Figure 3.8 – Screenshot of the rolling ball maze mini-game

3.1.1.6. Overall Review

This title is a simple game that is easy to pick up and play. It is well suited to children with its cartoon style and simple controls.

The level layout with its hubs and portal system, and with the ability to replay levels to reach 100% completion, is good for replayability, as such this is something that will be implemented into this project. Additionally, the burrowing mechanism is a really good way to reach previously unexplorable areas, and so this will also be added to the planned abilities for this project.

3.1.2. Yooka-Laylee (Playtonic, 2017)

3.1.2.1. Storyline

The premise of this game is that Yooka the chameleon and Laylee the bat, have their “one book” stolen by the main antagonist Capital B of the Hivory Towers Corporation. Yooka and Laylee then proceed through the Hivory Towers building and its levels contained within to collect the missing “pagies” to re-assemble their “one book” and defeat Capital B.

3.1.2.2. Character controls and abilities

The player is controlled using a third-person control style. This game has a basic starting control which gets expanded upon using a shop system to buy new abilities using quills as you progress through the game. Many of these abilities use energy that needs to be recharged using pickups or waiting for it to recharge automatically over time.

You start with just being able to move and jump, and then you can purchase the following abilities from the in-game shopkeeper Trowzer as shown in figure 3.9 below.

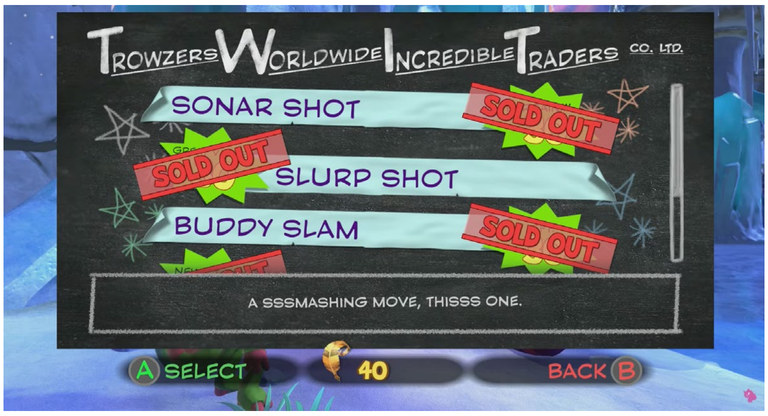

TROWZERS WORLDWIDE INCREDIBLE TRADERS CO. LTD.
SONAR SHOT
SOLD OUT
SOLD OUT
SLURP SHOT
BUDDY SLAM
SOLD OUT
A SSSMASHING MOVE, THISSS ONE.
A SELECT
40
BACK B

Figure 3.9 – Screenshot of Trowzer’s in-game shop.

1. **Tail Twirl** – A basic spin attack.
2. **Roll** – This allows the player to move around more quickly as well as the ability to roll up steeper slopes.
3. **Sonar Shot** – This allows the player to activate in-game items (switches) as well as to show invisible items.
4. **Slurp shot** – This allows the player to suck in elemental berries (fire/ice) and spit out attacks of that element.
5. **Buddy Slam** – This allows the player to slam down and destroy tougher enemies and certain objects and switches.
6. **Glide** – This allows the player to glide in mid-air and cross greater distance gaps to reach previously unreachable areas.
7. **Slurp State** – This allows the player to suck in certain game items and receive that item’s properties (Cannonballs make the player heavy and immune to wind, Honey makes the player sticky and able to walk up certain surfaces).
8. **Lizard Leap** – This allows the player to perform a super jump to reach higher areas.
9. **Buddy bubble** – This allows the player to walk underwater and requires surfacing for air much less often.
10. **Lizard lash** – This allows the player to use their tongue as a grappling hook to reach various platforms.
11. **Sonar ‘Splosion** – Ability to release a much larger Sonar Shot in all directions that also destroy nearby objects and stun enemies.
12. **Camo Cloak** – This allows the player to become invisible and sneak past enemies and checkpoint CCTV cameras.
13. **Reptile rush** – Allows charging up a spin dash attack to kill enemies and break cracked glass and ice.
14. **Flappy flight** – This allows the player to fly and reach all areas within the game.

15. **Sonar shield** – This allows the player to generate a shield whilst rolling.

3.1.2.3. Gameplay flow and mechanics

The game centres around the main hub level Hivory towers, which in turn contains five themed levels. These levels start as un-expanded versions which can later be expanded using quills to complete the level 100%, using abilities purchased later in the game.

3.1.2.4. Collectables

Each level contains the following collectable items which need to be collected to complete the level to 100%.

- A set number of pagies.
- A set number of quills.
- A Rextro's Arcade minigame to play. Beating the two high scores will award two pagies.
- A Molly cool, which allows the player to transform into a different shape and access certain areas / beat certain enemies.
- Five ghost-writers that must be collected in the following different ways.
    1. Yellow – Walk into it to collect it.
    2. Red – Need to dodge its attack, then attack it back to collect it.
    3. Green – Very fast so will need to be quick to touch it and collect it.
    4. Blue – This is invisible so will need to be located using the sonar shot.
    5. Pink – Needs to be fed using the slurp shot to catch it.

3.1.2.5. Level design

The six levels within the game are themed as follows.

- **Hivory Towers** – The main hub level and headquarters to the main antagonist Capital B as shown in figure 3.10 below.

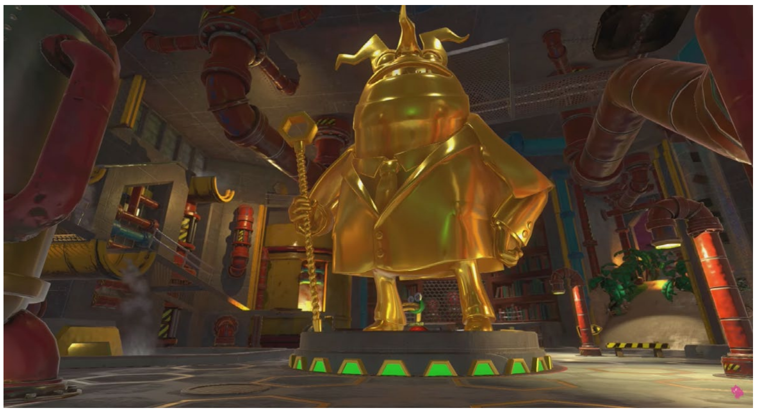


Figure 3.10 – Screenshot of the Hivory Towers hub level

- **Tribalstack Tropics –** A tropical Aztec themed world as shown in figure 3.11 below.

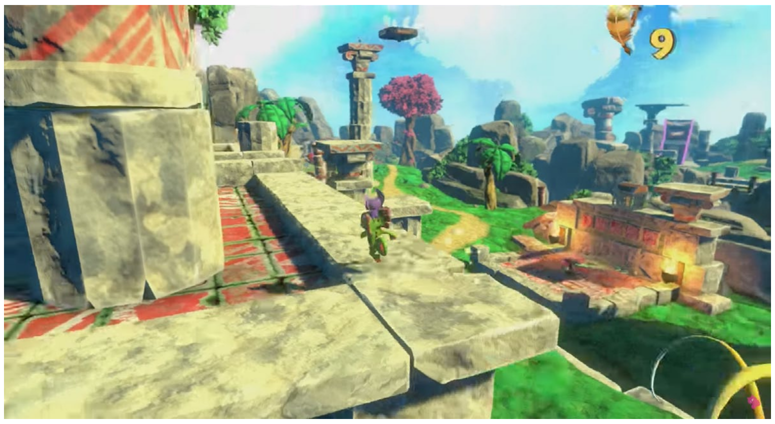


Figure 3.11 – Screenshot of the Tribalstack Tropics level

- **Glitterglaze Glacier –** An ice-themed glacier level as shown in figure 3.12 below.

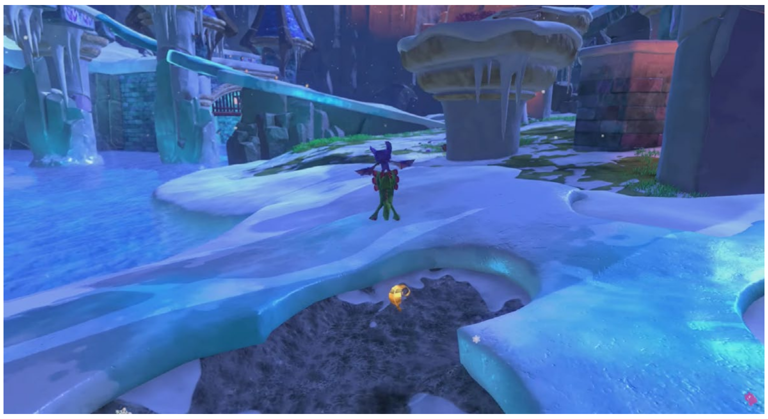


Figure 3.12 – Screenshot of the Glitterglaze Glacier level

- **Moodymaze Marsh –** A spooky swamp themed level as shown in figure 3.13 below.

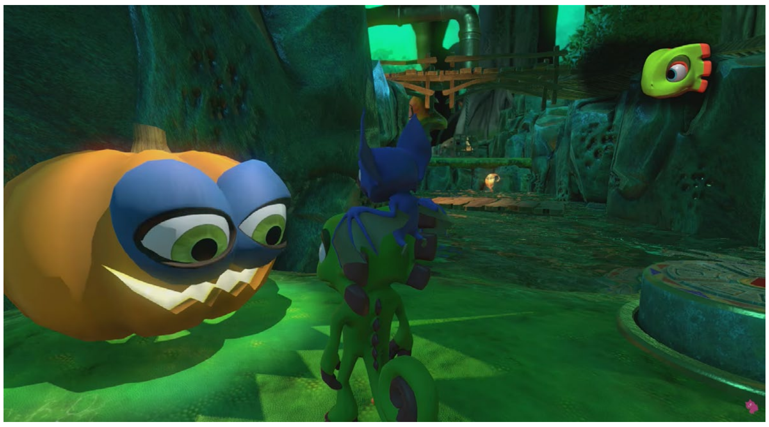


Figure 3.13 – Screenshot of the Moodymaze Marsh level

- **Capital Cashino –** A casino themed level as shown in figure 3.14 below.

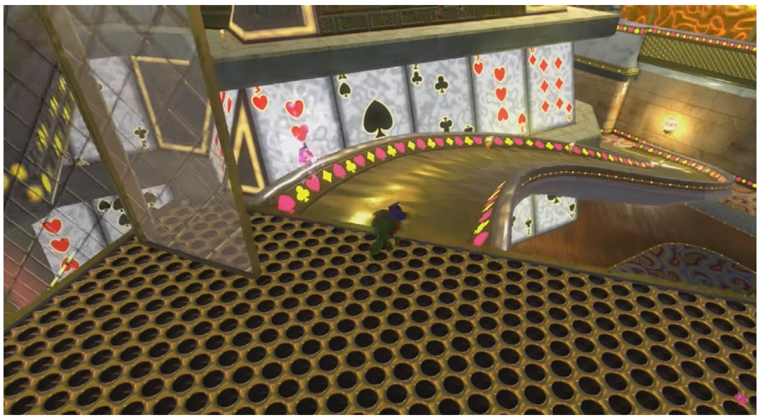


Figure 3.14 – Screenshot of the Capital Cashino level

- **Galleon Galaxy – A** space-themed level as shown in figure 3.15 below.

Figure 3.15 – Screenshot of the Galleon Galaxy level

3.1.2.5.4. Mechanics

This title has a vast number of different mechanics to challenge the player. These include races with AI-controlled characters, timed collectable runs, puzzles using switches and unlockable abilities and areas that are unreachable until you have progressed further in the game.

3.1.2.5.5. Enemies

The enemies in each level are themed and have different attacks and abilities of their own. Some enemies can only be defeated if you have purchased a certain ability and must be avoided otherwise.

3.1.2.6. Overall Review

This title is a good game with lots of excellent concepts, a good story and quite a retro feel.

The controls start simple and gradually build up with the addition of more abilities. This system along with the level expansion system allows for lots of replayability, with players required to return to previously completed levels once they have purchased new abilities to complete new tasks and access the remaining collectables.

This use of purchasing abilities to access previously inaccessible or new areas is a great concept to extend gameplay and will be seriously considered in the plan for this project.

3.1.3. Existing titles conclusion

In looking at the two titles detailed above, the main thing that both have in common is that they have a similar gameplay flow, where they contain the main hub level, which branches off into the various themed levels. This is something that will be one of the main requirements for this project.

Both titles also require the player to collect a required number of generic items to complete levels to 100%, this is something that will be added to the project as a requirement.

Super Lucky's Tale (Playful, 2019) also contained an interesting burrowing mechanic that will be intended to be implemented in this project.

Additionally, the gameplay feature of collecting letters from the protagonist's name, as part of the completion of each level is something that will be implemented in the project.

Yooka-Laylee (Playtonic, 2017) has a vast number of ability upgrades and there is the intention of implementing a variety of these in this project.

3.2. Game engine comparisons

The choice of the game engine for the project will come down to the two main engines used in modern games. These are the Unity Engine (Unity, 2021) and the Unreal Engine (Epic, 2021).

Unreal Engine (Epic, 2021) uses a mixture of C++ scripts and blueprints to implement the game logic, whereas Unity Engine (Unity, 2021) uses C# scripts to perform these actions.

In general Unreal Engine (Epic, 2021) developed games are more photorealistic in style, for example, Tony Hawks PRO Skater 1+2 (Activision, 2022) and Batman Arkham Knight (Rocksteady, 2015) as shown in figures 3.16 and 3.17 below.

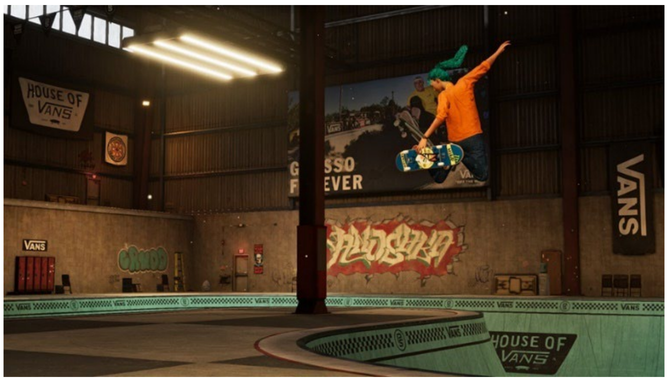


Figure 3.16 – Tony Hawks Pro Skater's (Activision, 2022) photorealistic graphical style

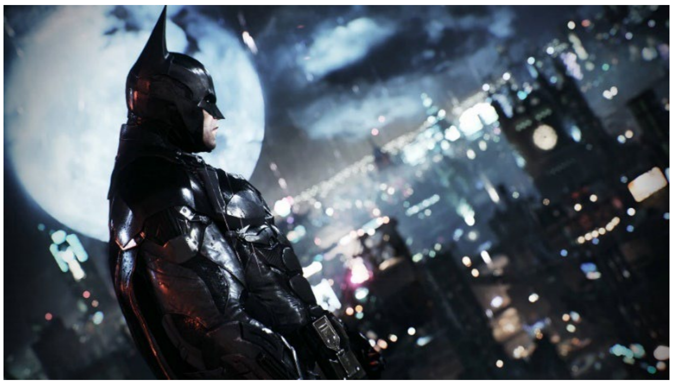


Figure 3.17 – Batman Arkham Knight's (Rocksteady, 2015) photorealistic graphical style

In comparison, Unity Engine (Unity, 2021) developed games, that are more cartoon styled such as Overcooked (Team17, 2022) and Two Point Hospital (Sega, 2021)as shown in Figures 3.18 and 3.19 below.

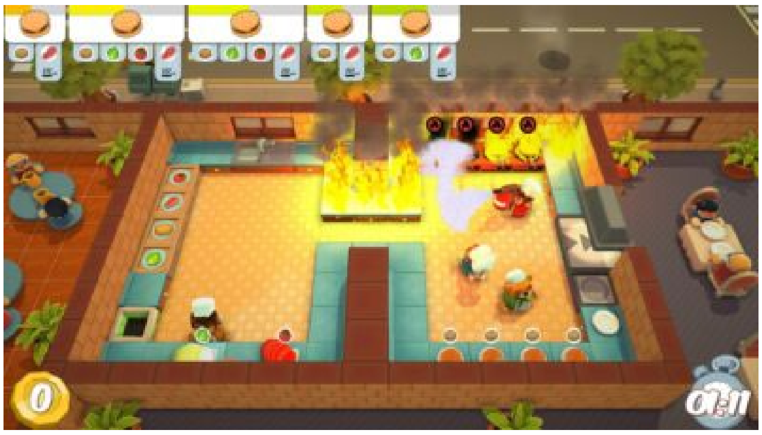


Figure 3.18 – Overcooked (Team17, 2022) cartoon graphical style

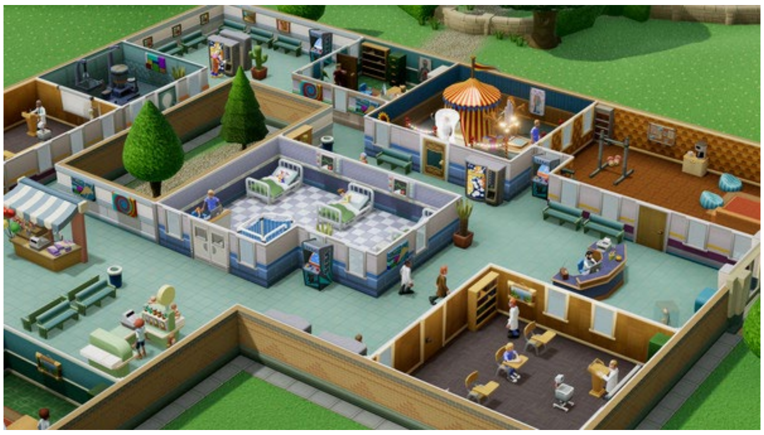


Figure 3.19 – Two Point Hospital's (Sega, 2021) cartoon graphical style

As such the intention is to use Unity Engine (Unity, 2021) as this is more suited to the desired style and feel of the project. Additionally, I am personally more proficient in the use of C# than in C++, and as such feel that the project will proceed more efficiently using the Unity Engine (Unity, 2021) and C#.

3.3. Other software

This section will consider various software choices for different areas in the development of the project.

3.3.1. 3D Modelling

For the 3D modelling required for the project, the software in use will be Blender (Blender, 2021) as this is free software that integrates seamlessly with the Unity (Unity, 2021) game engine that has been selected for the project. As I have a lot of experience in using Blender (Blender, 2021), using this software will aid in the productivity of the project.

3.3.2. Photo Editing

For photo editing, used in the creation of 3D model textures, logos and Ui elements, the intention is to use Adobe Photoshop (Adobe, 2021), as this is an extremely common photo editing software that is widely used and well suited to these tasks.

3.3.3. Animation

In creating the animations for the characters in the game, the intention is to use a mixture of custom made animations created in Blender (Blender, 2021) and Unity (Unity, 2021), alongside ready-made animations acquired from the Mixamo (Adobe, 2021) website. This approach will aid in increasing productivity as a lot of simple animations can be imported and edited a lot quicker than creating them from scratch using Blender (Blender, 2021) and Unity (Unity, 2021).

4.0. Requirements Specification

Following the background research conducted, the intended plan for this project will be as follows.

4.1. Storyline

The general story will revolve around the main antagonist Evil Lord Graham transporting Bow-Bow's prized collection of coloured bones to various themed worlds via portals he creates, to be protected by his minions because he is bored one Sunday morning. The player will then need to visit each world to collect the bones from Bow-Bow's collection.

4.2. Character controls and abilities

R1 For this project, the intention is to implement both the keyboard and mouse and gamepad controller input schemes.

R2 The controls will remain simple to enable children to be able to play quite easily, for example, the player will only need to press a maximum of two buttons simultaneously to perform any action in-game.

R3 There will be simple movement in the x and y directions.

R4 There will be a Basic jump.

R5 There will be a spin attack.

R6 The player will gradually unlock abilities either via a shop or by completing levels/beating bosses, this enables the learning curve to not be too steep for the player and gradually ease them into gameplay.

R7 Possible abilities will include.

- **Double jump** – The Ability to jump again at the top of the first jump to reach higher areas.
- **Throw fish bones** – Used to attack enemies and hit distant switches for various gameplay mechanics.
- **Gliding** – Holding a button whilst jumping will allow the player to glide for a limited time to cross larger gaps and reach new areas.
- **Stomp attack** – Pressing a button whilst jumping to stomp attack enemies from above or to access lower areas / uncover items and switches.
- **Dig up objects** – press a button over a highlighted area to find keys and items by digging in that area.
- **Burrowing** – Hold a button to move around under the ground in certain areas to access areas underneath low entrances and to stun enemies.
- **Walk up steep slopes** – Hold a button to allow the player to walk up steeper slopes for a limited time to access new areas.

R8 The camera will follow the player in the third person.

R9 The intended system for camera controls will be to use the mouse or right controller stick to rotate the camera around the player's character.

4.3. Gameplay flow and mechanics

The Game will start in the main hub level (R10) with just the first level portal unlocked. Then once each level has been completed by getting to the end and beating the boss (R11), it will unlock the next level. After each of the levels has been completed, a portal (R12) will appear to the final boss fight (R13) with the main antagonist.

Each level will contain the following collectables.

- A different coloured main story bone. (R14)
- A set number of regular bones. (R15)
- Collectable letters B, O, W (Brown/yellow). (R16)
- Collectable letters B, U, M (Black/white). (R17)

Collecting each of these will award a badge on the level portal, four of these badges (R18) will be required to complete the level 100%. The regular bones will possibly be used as currency for the abilities shop (R19), if not an ability will be awarded upon beating an end of level boss.

I plan to have three hearts as the health system (R20) which can be expanded by finding hidden heart containers/beating bosses. The hearts can be refilled by collecting steak health pickups (R21).

There will be fishbone pickups (R22) to be used as ammo for Bum-Bum to throw at enemies and switches (R23). Early levels will require abilities from later in the game to be completed 100%.

4.4. Level design

The initial ideas for the level themes are as follows

- Main hub level – Garden theme with a doghouse and unlockable areas for the portals. (R10)
- Sci-fi themed level. (R24)
- Prehistoric. (R25)
- Forest themed level. (R26)
- Medieval themed level. (R27)
- Tropical Island themed level. (R28)
- Industrial area themed level. (R29)

The levels and pickups will be made up of both custom models, as well as assets acquired from third parties to meet the time restrictions for the project.

4.5. Characters

The main player characters will be modelled in Blender, including texturing, rigging and animations to demonstrate my abilities in this area.

With the awareness that time is limited for the project, for most of the enemies, the models will be items available from the Unity asset store (Unity, 2021) and third-party asset websites such as Mixamo (Adobe, 2021).

4.6. Project plan

The initial project proposal is shown in ***Appendix B*** and the Gantt chart to indicate the proposed project plan for each game element is shown in ***Appendix C***

5.0. Game Design

5.1. Main Character Design

The main character was designed to have Bow Bow the dog carrying Bum Bum the cat in a backpack as shown in Figure 5.1 below. This setup was inspired by playing Banjo Kazooie (Rare, 1998) during the research stage of the project ***(see Appendix B)*** which has a similar character setup.

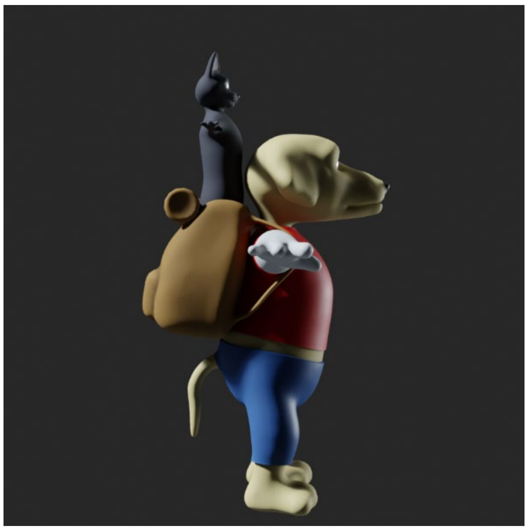


Figure 5.1 – Main character setup

The design of this character then means that the main movement, spin attack, jumping and drop attack are handled by the Bow Bow character and the look, aiming and glide abilities will be handled by the Bum Bum character. The character starts with three bars of maximum health which increases each time all letters in a level are collected.

5.2. Level Design

The main structure of the game levels is that there is a main base level that is split up into four sections, each new section requires an ability to be collected in the side levels to reach it and contains the next portal to another side level. Additionally, some levels require future abilities to be gained to access certain areas and collect all collectable items within the level to complete it 100%.

Each level contains the following collectables which must be collected to complete the level 100%

- One hundred bones
- Six letters (BOWBUM)
- One Golden Bone

Progress in each level is saved upon collecting any item or ability and each area has a checkpoint to respawn the player upon the player dying. The checkpoints for levels accessed from the hub level are reset when the player returns to the main hub level. Checkpoints within the hub level are persistent between game sessions.

5.2.1. Level Layouts

The level layout designs are shown below detailing the structure and progression through the game.

5.2.1.1. Base Level (Home)

The base level acts as a main hub for the game and contains four main areas. The layout is shown below in figure 5.2.

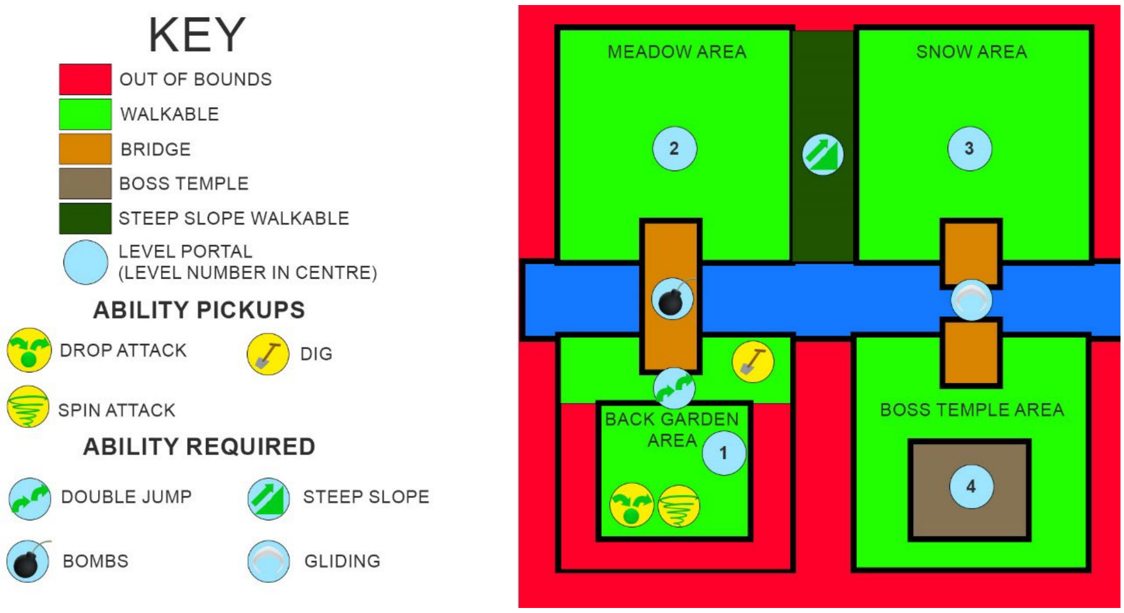


Figure 5.2 – Base level layout

- **Back garden** – This area is the starting area and gives the player the spin attack and drop attack abilities to start the game with. To progress past this area, the player must enter the portal to level one and gain the double jump ability to allow the player to jump over the back fence.
  This area contains chicken enemies (see 5.3.2.1 Base Level (Home) below).
- **Meadow Area** – This area can be accessed via a bridge that is blocked by a large rock and must be blown up using the bomb-throwing ability. This ability requires the shooting ability to be gained from level one before use. This area contains the portal to level two**.**
  This area contains chicken, cactus, and scorpion enemies (see 5.3.2.1 Base Level (Home) and 5.3.2.2 Level One (Deserted Dunes) below).
- **Snow Area** – This ice themed area can be accessed once the steep slope ability has been gained within level two allowing the player to walk over the steep divide from the meadow area. This area contains the portal to level three (not implemented yet).
  This area contains robot, and turret enemies (see 5.3.2.1 Base Level (Home) and 5.3.2.3 Level Two (Robot Wars) below).
- **Boss Temple area** – This contains a boss temple structure that can be traversed via a spiral ramp around the outside. This pyramid-shaped structure has the final boss level portal (not implemented yet) on top of it. This area is accessed via a broken bridge that requires the glide ability to reach by gliding across the broken bridge section.

5.2.1.2. Level One (Deserted Dunes)

The layout for level one is shown in figure 5.3 below.

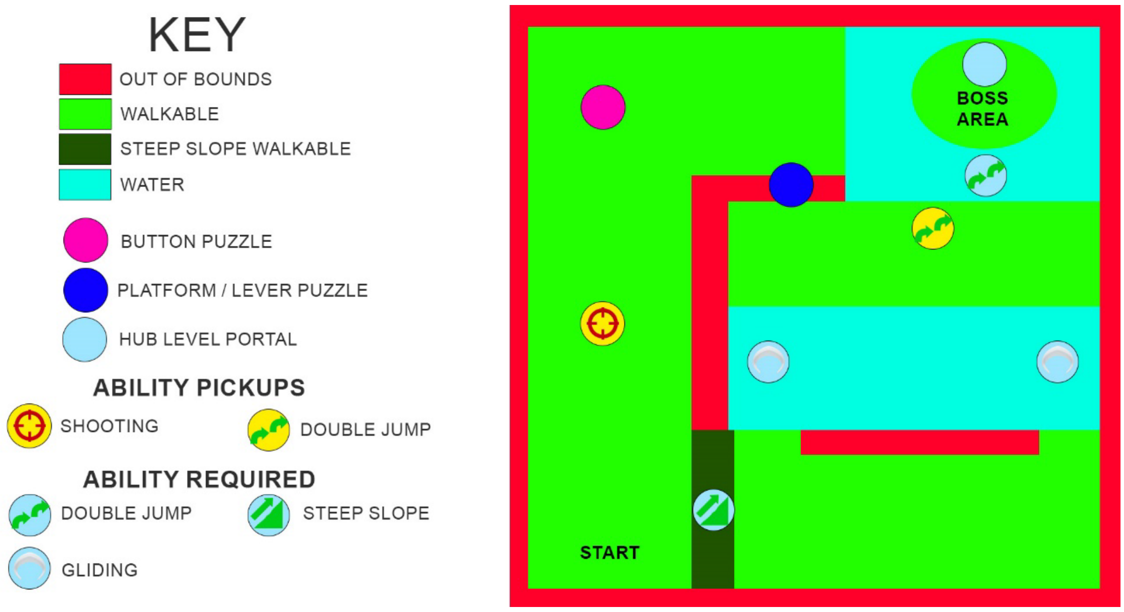


Figure 5.3 – Level one Layout

This level is fully themed around a desert scene. It offers the player the shooting ability at the start. In the next area, there is a button puzzle to complete to gain a letter collectable. Following on from this is a moving platform puzzle that requires the shoot ability to activate and progress over a high wall. Once over this wall, there is an area straight ahead that is not accessible until the player has acquired the glide ability further on in the game. Around to the left, the player can acquire the double jump ability which is needed to reach the level boss area across a wide gap. Once across this gap, the player must defeat the level boss (5.3.2.2 Level One (Deserted Dunes) below) to collect the golden bone for this level and unlock the portal back to the hub level.

This level contains static cactus and mobile scorpion enemies as well as the end boss (5.3.2.2 Level One (Deserted Dunes).

5.2.1.3. Level Two (Robot Wars)

The layout for level two is shown in figure 5.4 below.

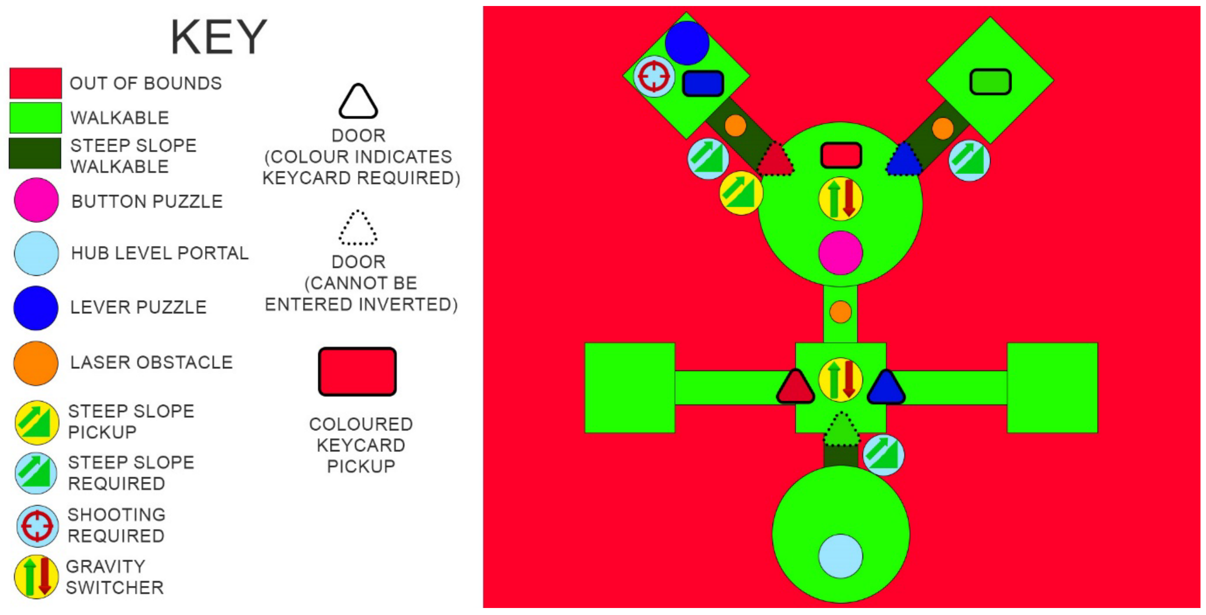


Figure 5.4 – Level two layout

Level two is themed around a Sci-fi space station theme. At the start of the level, there are four doors, three of which require coloured key cards to enter. Upon entering the unlocked door, the player is confronted with a laser obstacle that instantly kills them. To pass this obstacle the player must use the gravity switcher to invert gravity and enter this door inverted and walk along the ceiling above the lasers.

The next room contains two doors locked with red and blue key cards. There is a button puzzle in the centre of the room that has buttons on both the ceiling and the floor. Upon completing this puzzle, the player acquires the red key card to progress through the first door. Through this door, the user can gain the steep slope ability to walk up the steep corridor. This corridor contains another laser obstacle alternated which laser array is on, moving up the slope. The player must time their ascent to avoid the lasers and reach the top of the slope before the steep slope ability times out.

In the next room is a lever puzzle where the levers must be switched on and off in the correct sequence displayed on the wall by shooting them through a narrow slot in the wall. Upon completion of this puzzle, the user receives a blue key card. The player then needs to return to the previous room and enter the previously locked blue key card door.

Through this door, there is a corridor that is identical to the previous one except that the laser obstacle now alternates which array is switched off thereby making the traversal much harder.

In the room at the end of this corridor is a large Turret enemy that when defeated drops a green key card allowing access to the boss room from the start room.

Back in the starting room, the player can now access both side rooms unlocked with the red and blue key cards to collect all collectable items before entering the green key card door to enter the boss room.

In the final boss room, the player must defeat the level two boss to collect the golden bone for this level and unlock the portal back to the hub level. During this boss fight when the boss's health reaches 75% four sections of the floor open, followed by a further four when the boss's health reaches 50%. This makes the boss fight much harder to complete.

Upon entering this portal, a message is displayed indicating that the game is "to be continued…" and suggesting that the player returns to previous levels to complete them 100% by collecting all the collectable items.

5.3. Enemy Design

5.3.1. Enemy Types

There are three main types of enemies with different variants for each level. Each enemy can be set to inflict damage on contact with the player as well as if they are damageable using the drop attack or not.

These enemy types are,

- **Static Turret Enemy** – These enemies rotate in place either at a steady speed or randomly until the player is in range. When the player is within a certain range, the enemy checks if the player is within a set vision cone, and if they are the enemy rotates towards the player and fires projectiles toward them.
- **Mobile Enemy** – Mobile enemies randomly move around within a given radius from their starting position until they have a line of sight to the player. When the line of sight is achieved the enemies move toward the player's location until the line of sight is lost.
- **Boss Enemy** – The boss enemies are customised with different behaviours as detailed in the level enemy variants section 5.3.2 below.

5.3.2. Level Enemy Variants

5.3.2.1. Base Level (Home)

Chicken enemy

The chicken mobile enemy as shown in figure 5.5 below has two health points. It follows random waypoints within a set range, stopping at each and performing a random idle animation. If the enemy obtains a line of sight with the player or takes damage, it charges towards the player until the line of sight is lost, when it will return to its idle state. This enemy inflicts damage when it contacts the player but can be damaged from above using the drop attack. The Base level also contains variant enemies from all other levels in the area surrounding that level's portal.

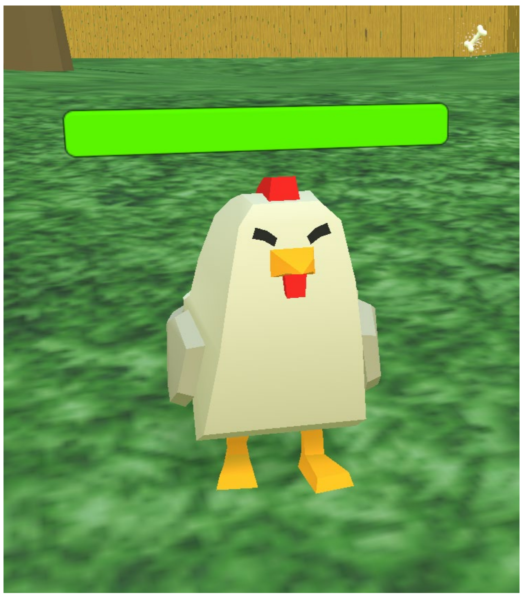


Figure 5.5 – The chicken enemy

5.3.2.2. Level One (Deserted Dunes)

Scorpion enemy
The scorpion mobile enemy as shown in figure 5.6 below has three health points. It follows random waypoints within a set range, stopping at each and performing a random idle animation. If the enemy obtains a line of sight with the player or takes damage, it runs towards the player until the line of sight is lost, when it will return to its idle state. If the enemy gets within attack range, it will attack the player using its tail sting. This enemy inflicts damage when it contacts the player ant with its sting. It cannot be damaged from above using the drop attack.

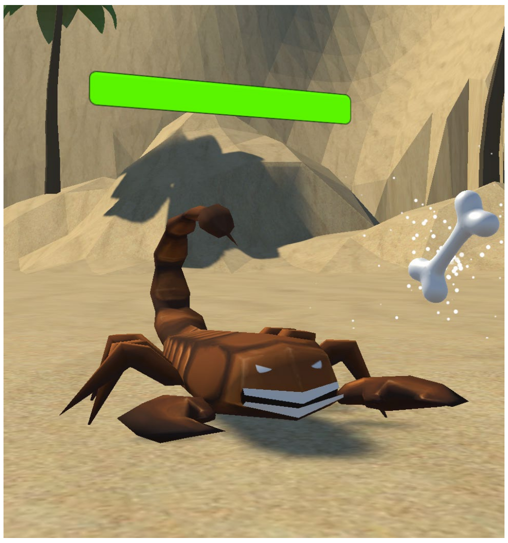


Figure 5.6 – The Scorpion enemy

Cactus enemy
The Cactus enemy as shown in figure 5.7 below has six health points. It rotates randomly in place until it gains line of sight to the player. The enemy then rotates around the Y-axis towards the player and when it is facing them shoots a cactus spine towards the player. If the player breaks line of sight the cactus returns to its rotating idle state. The cactus enemy inflicts damage on contact and cannot be damaged from above using the drop attack.

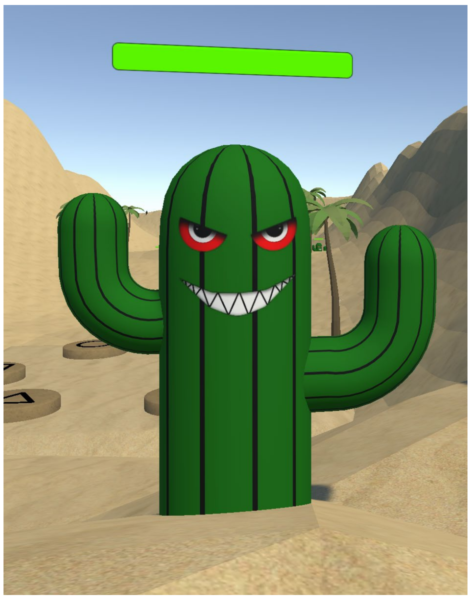


Figure 5.7 – The cactus enemy

Level One Boss – Doozy (Adobe, 2021)
The level one boss as shown in figure 5.8 below has twenty-five health points. It follows random waypoints within a set range, stopping at each and performing a random idle animation. If the enemy obtains a line of sight with the player or takes damage, it runs towards the player until the line of sight is lost, when it will return to its idle state. If the enemy gets within attack range, it will attack the player by swinging a large wooden club that it carries. This enemy inflicts damage on contact but can be damaged from above using the drop attack.

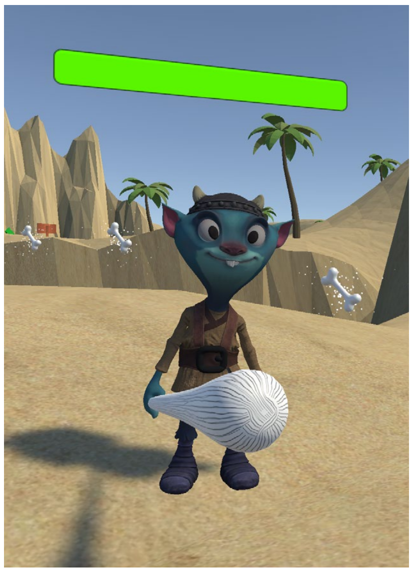


Figure 5.8 – The level one boss

5.3.2.3. Level Two (Robot Wars)

Robot Enemy
The Robot mobile enemy as shown in figure 5.9 below has five health points. It follows random waypoints within a set range, stopping at each and performing an idle animation. If the enemy obtains a line of sight with the player or takes damage, it charges towards the player until the line of sight is lost, when it will return to its idle state. This enemy inflicts damage when it contacts the player but can be damaged from above using the drop attack.

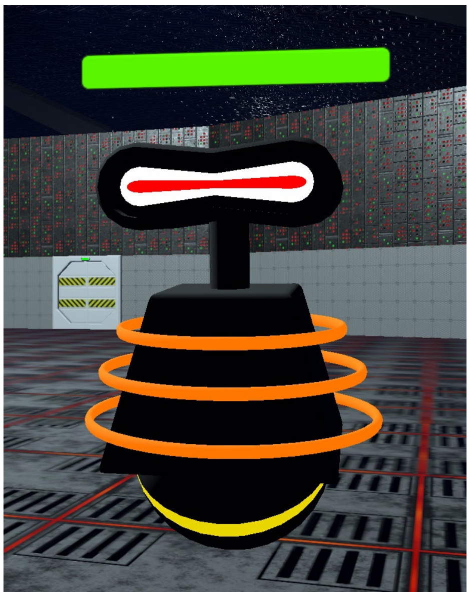


Figure 5.9 – The robot enemy

Turret enemy

The turret enemy as shown in figure 5.10 below has six health points. It rotates in place until it gains line of sight to the player. The enemy then rotates around the Y-Axis towards the player and when it is facing them shoots a laser burst towards the player. If the player breaks line of sight the turret returns to its rotating idle state. The turret enemy does not inflict damage on contact and can be damaged from above using the drop attack.

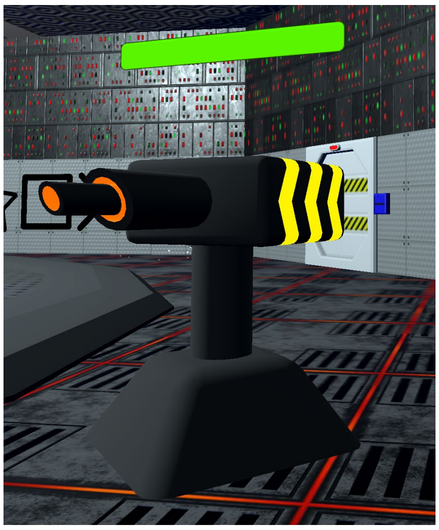


Figure 5.10 – The turret enemy

Large Turret Enemy

The large turret enemy as shown in figure 5.11 below has twenty-six health points. It rotates in place until it gains line of sight to the player. The enemy then rotates to look at the player and then shoots three laser bursts toward the player. If the player breaks line of sight the turret returns to its rotating idle state. The turret enemy inflicts damage on contact and cannot be damaged from above using the drop attack. This enemy drops the green key card needed to progress in the level and so is required to be beaten before accessing the final boss room.

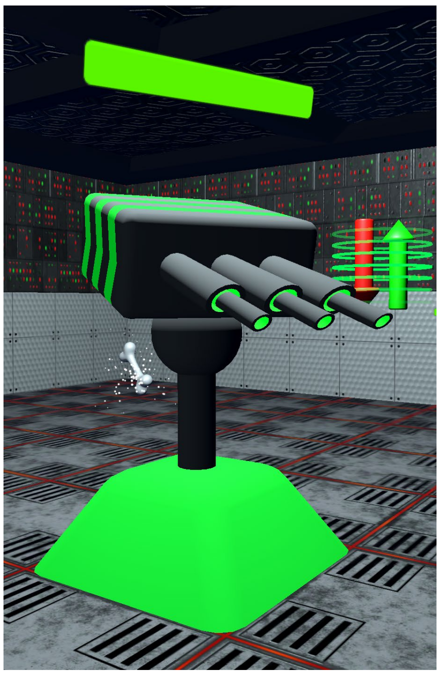


Figure 5.11 – The large turret enemy

Level Two Boss (YBot) (Adobe, 2021)
The level two boss as shown in figure 5.12 below has fifty health points. It follows random waypoints within a set range, stopping at each and performing a random idle animation. If the enemy obtains a line of sight with the player or takes damage, it runs towards the player until within shooting range and then fires laser blasts at the player. If the line of sight is lost, when it will return to its idle state. If the player gets within the enemies' attack range, it will spin and expel a forcefield that damages the player as shown in figure 5.13. When the enemies' health gets below 75%, 4 sections of the floor open as shown in figure 5.14. Then when the enemies' health drops below 50%, another four sections of the floor open as shown in figure 5.15. This makes this boss fight increase in difficulty as it proceeds.

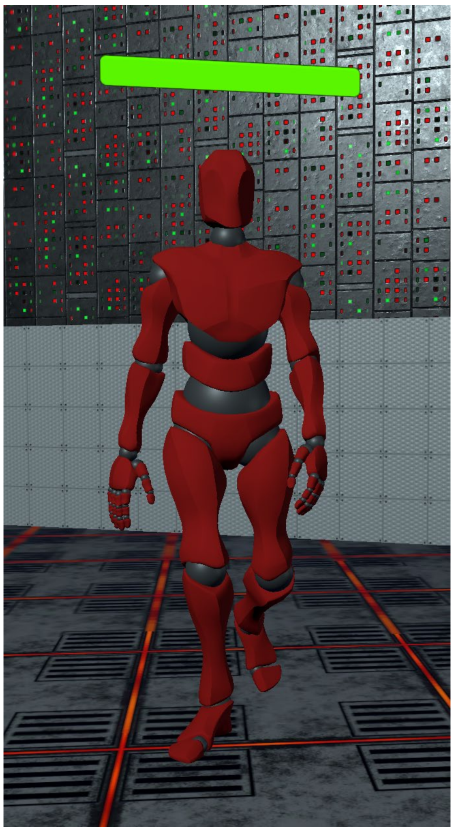


Figure 5.12 – The level two boss

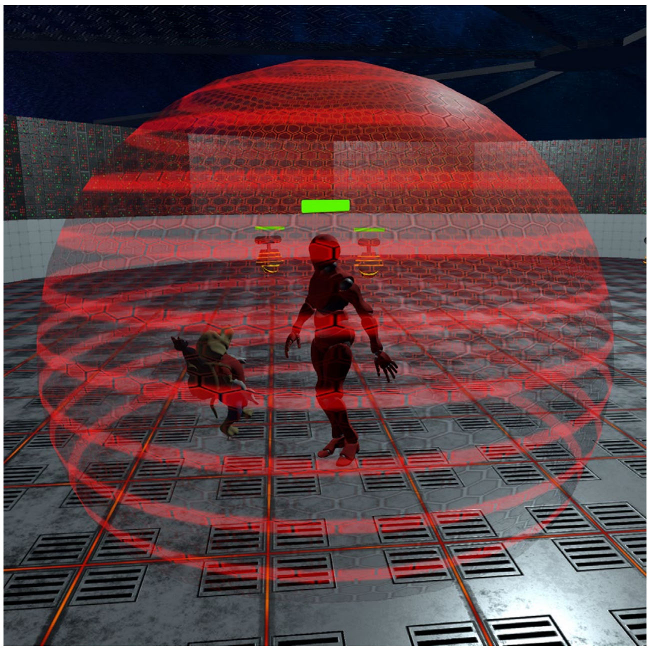


Figure 5.13 – The level two boss's spin forcefield attack

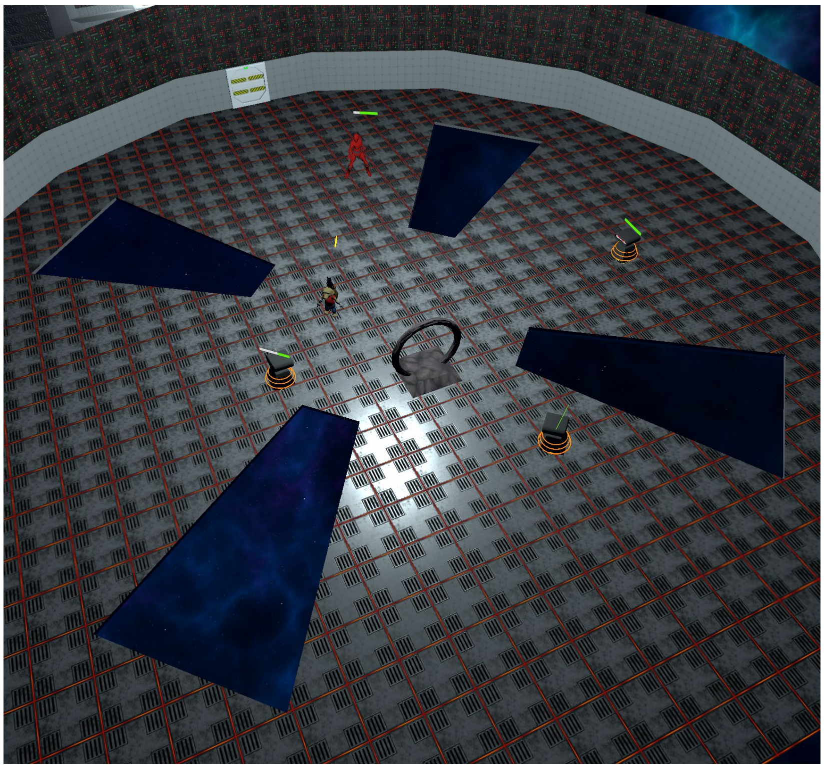


Figure 5.14 – Boss room floor stage one open at 75% boss health remaining

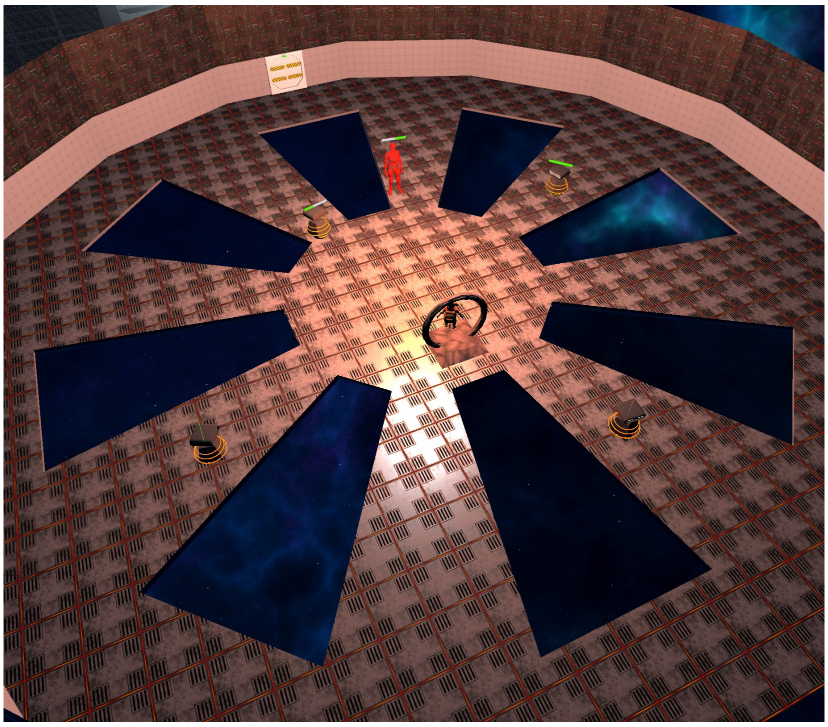


Figure 5.15 - Boss room floor stage open two at 50% boss health remaining

6.0. Game Development

6.1. Main Character development

6.1.1. Modelling

The main character models, first, were blocked out using basic cube shapes within Blender (Blender, 2021). To ensure symmetry, the Mirror modifier was applied to each object of the model as shown in Figure 6.1 below.

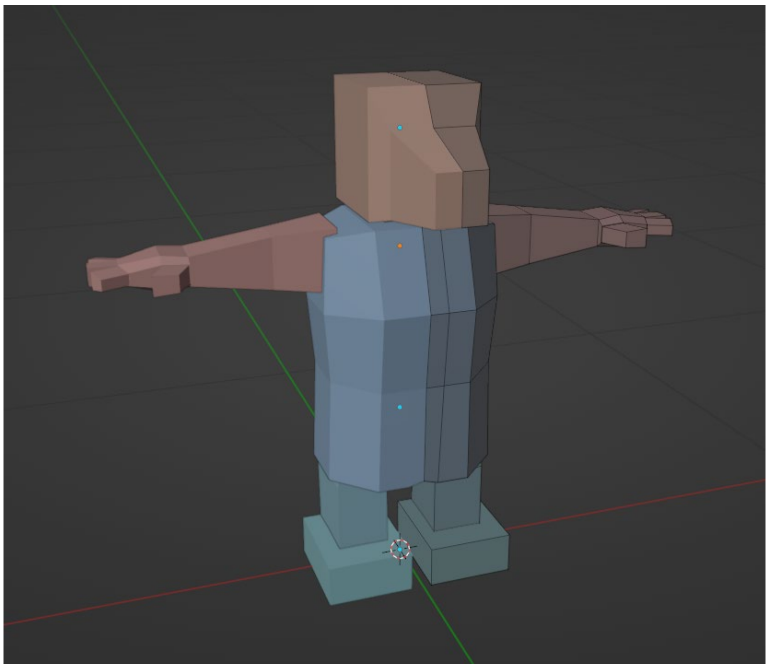


Figure 6.1 Basic block model of the main Bow Bow character.

These models were then sculpted using Blenders (Blender, 2021) sculpting feature and implementing the “Dyntopo” setting to add fine geometry to the model as shown in Figure 6.2 below.

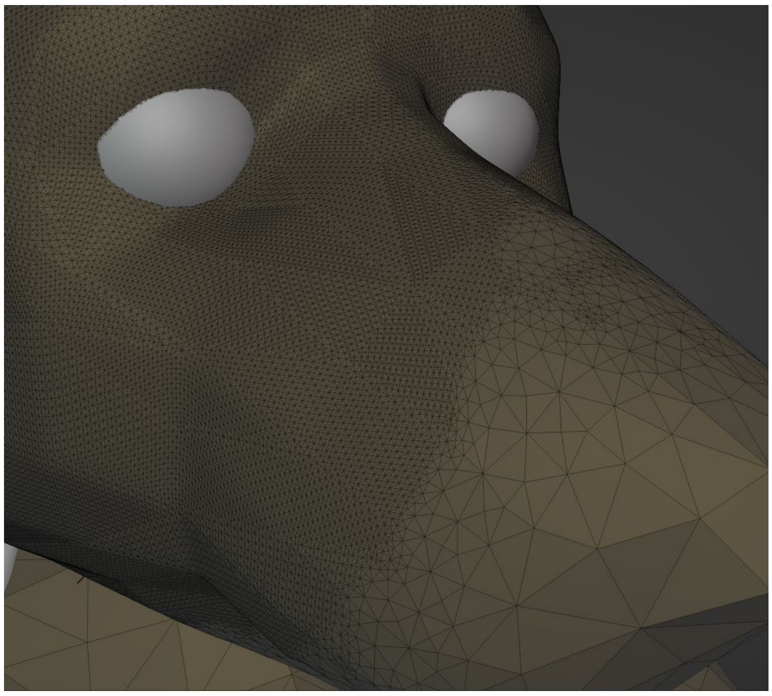


Figure 6.2 Sculpting character models using Blender “Dyntopo” feature.

Once the model had achieved the aesthetic desired for the project, they were re-meshed using a free trial of Quad Remesher (Exoside, 2020) with a setting of 250,000 faces to produce a final, high polygon count version of each model with even topology as shown in Figure 6.3 below.

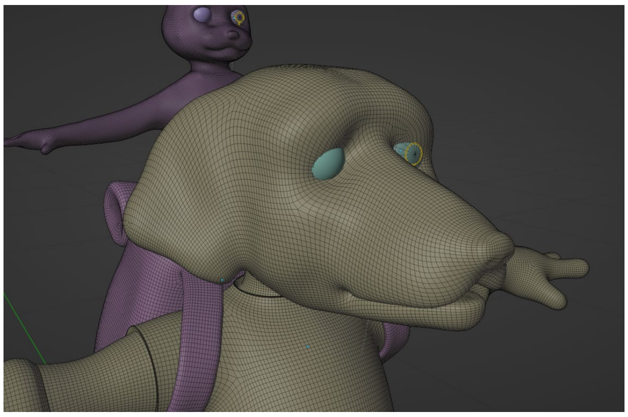


Figure 6.3 Character models re-meshed with a setting of 250,000 faces.

The high polygon Models were then re-meshed again with Quad Remesher (Exoside, 2020) using a setting of 15,000 faces to create a low polygon version. This was then marked with seams and UV unwrapped ready for adding textures as shown in figure 6.4 and figure 6.5 below.

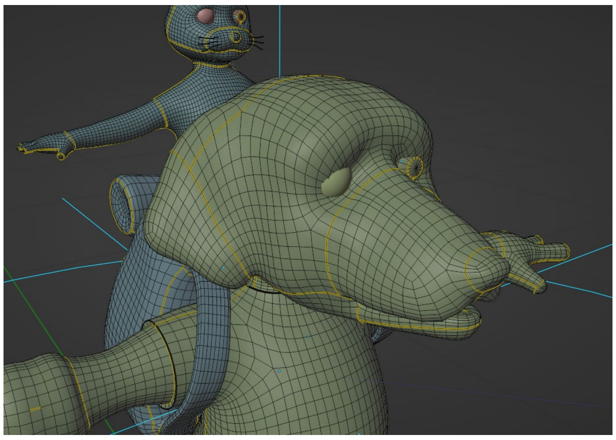


Figure 6.4 Character models re-meshed with a setting of 250,000 faces with UV seams marked

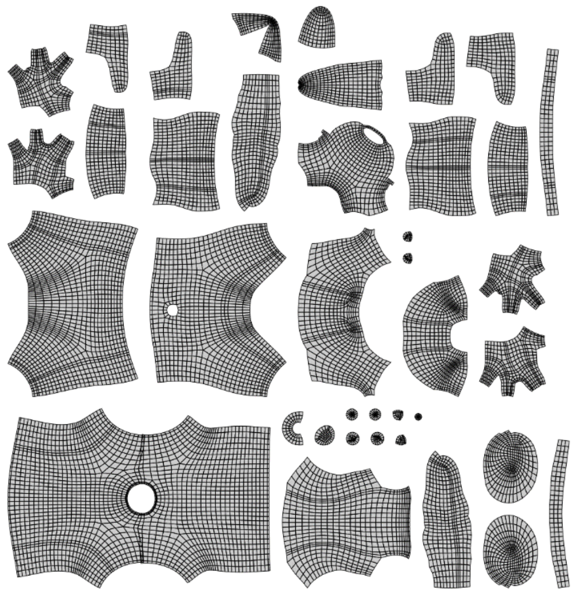


Figure 6.5 UV unwrap for Bow Bow character model

The differences were then baked into a normal map from the high polygon version to the low polygon version to retain the fine detail level, whilst using a far less detailed mesh as shown in Figure 6.6 below. This allows for a much lower polygon count model to vastly improve the performance one imported into Unity (Unity, 2021).

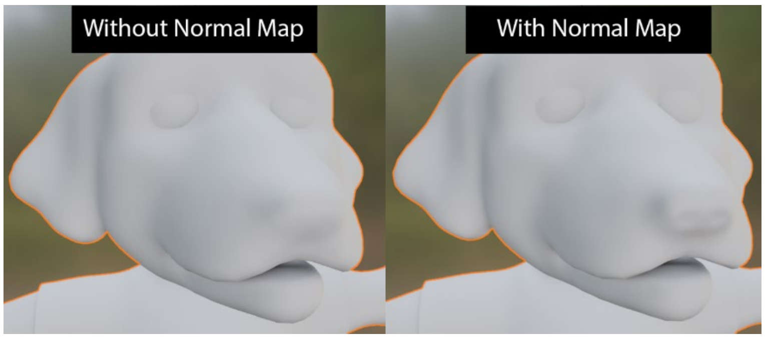


Figure 6.6 Low poly comparison showing the difference with Normal map applied.

The Low polygon model was then used as a base to paint all the basic colours onto a .png file as shown in Figure 6.7 below. The Low polygon model was also used to bake the ambient occlusion into a .png file to show the cavities of the model as shown in figure 6.8 below.

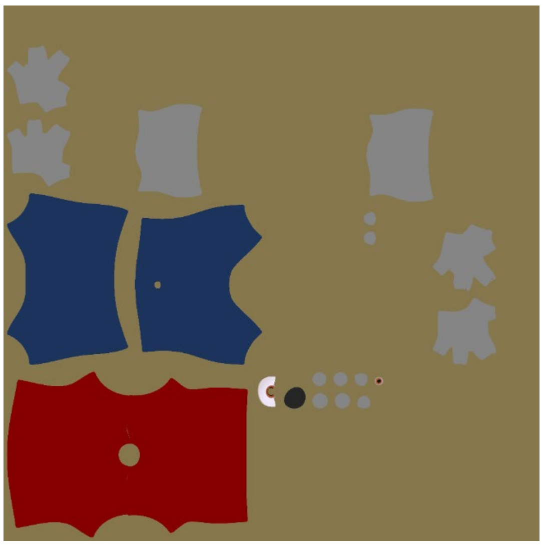


Figure 6.7 The painted colour map for the Bow Bow character model.

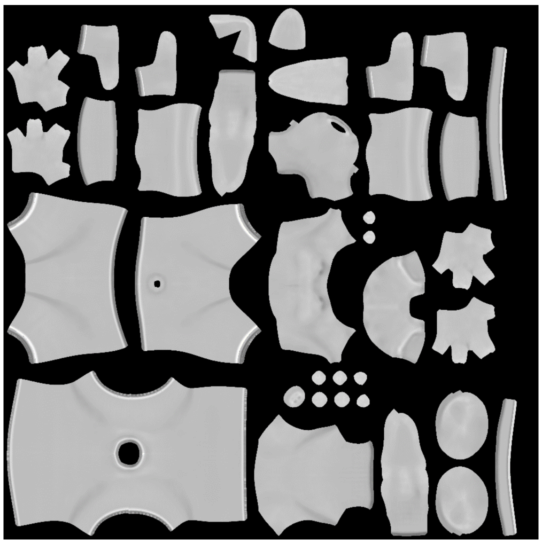


Figure 6.8 The baked Cavity map for the Bow Bow character model

These maps were then baked together using the overlay setting to create a finalised Albedo map for use within Unity (Unity, 2021). The difference between the colour map alone and with the cavity map overlaid can be seen below in Figures 6.9 and 6.10.

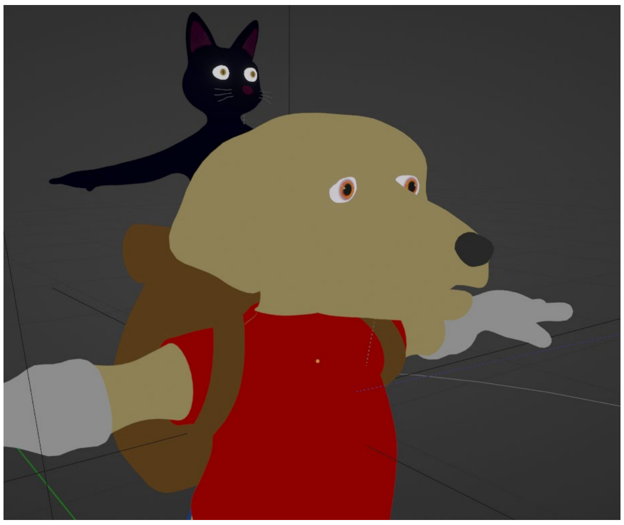


Figure 6.9 Bow Bow character model with just colour map applied.

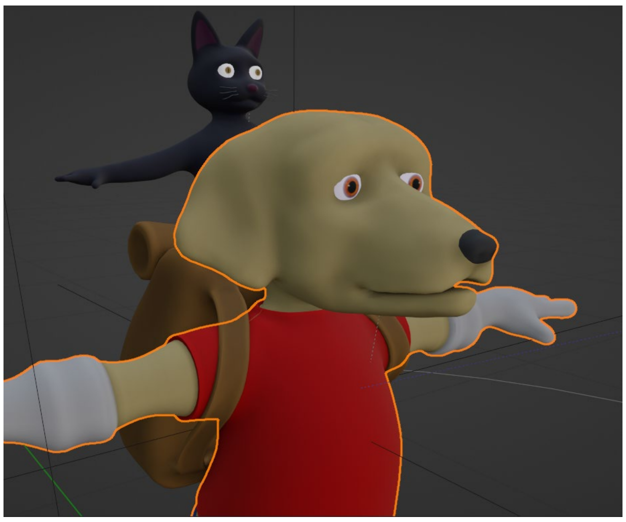


Figure 6.10 Bow Bow character model with colour map and cavity map overlaid.

6.1.2. Rigging and animation

6.1.2.1. Rigging

To animate the character models, they were first rigged using various methods. The main Bow Bow character was rigged using the Mixamo (Adobe, 2021) website as it is a basic humanoid shape and so could be uploaded and rigged automatically as shown in figures 6.11 and 6.12 below. This approach allowed the download of a variety of basic animations from the Mixamo (Adobe, 2021) website to use as a base for the in-game animations as shown in figure 6.13 below.

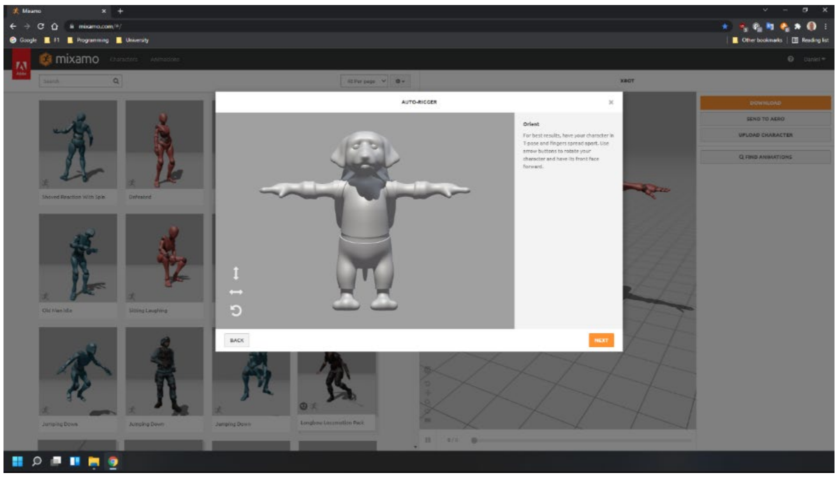


Figure 6.11 Uploading character model .fbx file to Mixamo (Adobe, 2021) website.

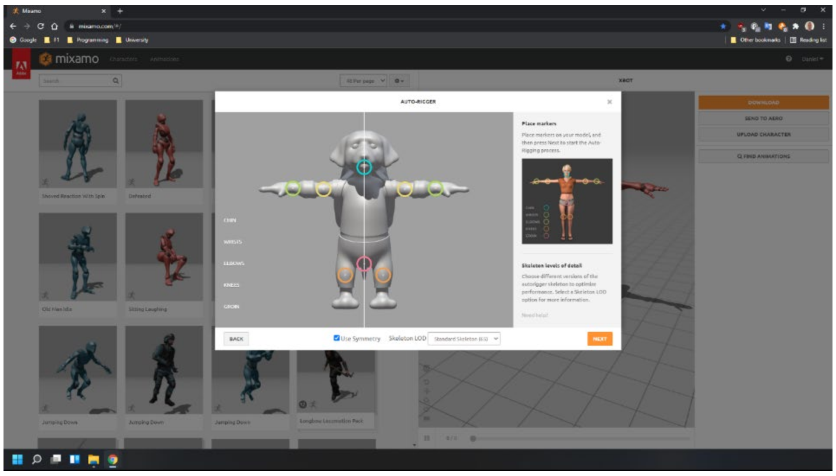


Figure 6.12 Using Auto-rigger tool on Mixamo (Adobe, 2021) website.

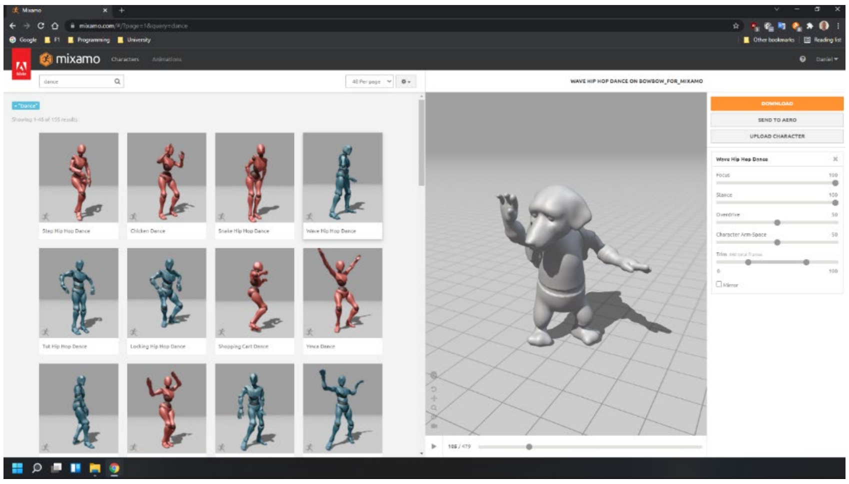


Figure 6.13 Selection of animations available on the Mixamo (Adobe, 2021) website.

The rig for this character was then downloaded as an FBX file and re-imported back into the Blender file ready for animation as shown in figure 6.14 below.

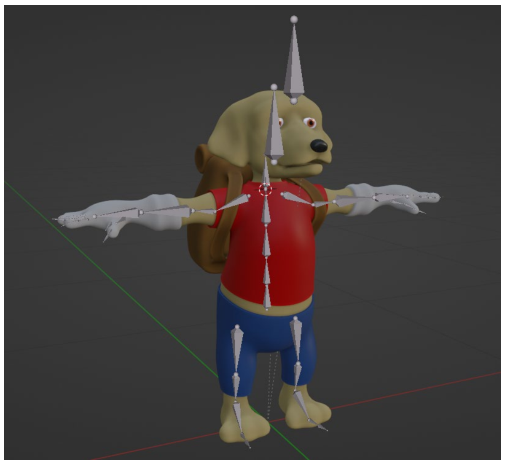


Figure 6.14 Bow Bow character model rig downloaded from the Mixamo (Adobe, 2021) website.

As the Bum Bum character model was not a basic humanoid shape, this model had to be rigged within Blender (Blender, 2021) as shown in figure 6.15 below.

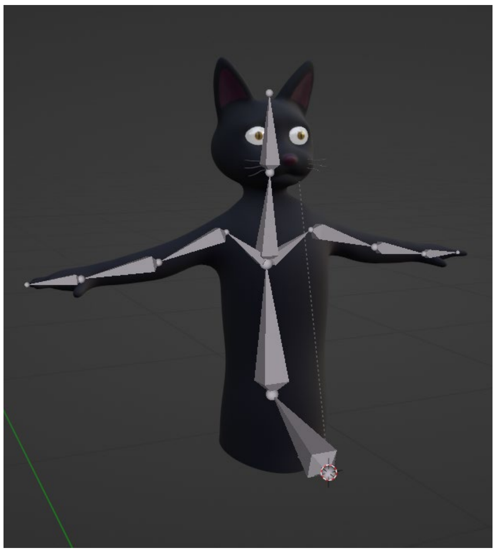


Figure 6.14 Bum Bum character model rig created within Blender (Blender, 2021).

6.1.2.2. Animations

The animations for Bow Bow were taken from the Mixamo (Adobe, 2021) website. Each animation was imported into Unity (Unity, 2021) and then added to an animator object as shown in figure 6.15 below, which resembles a finite state machine. This animator was configured with various parameters to set walking, idle, spin attack etc so that each one could be set from a script to blend from one state animation to another.

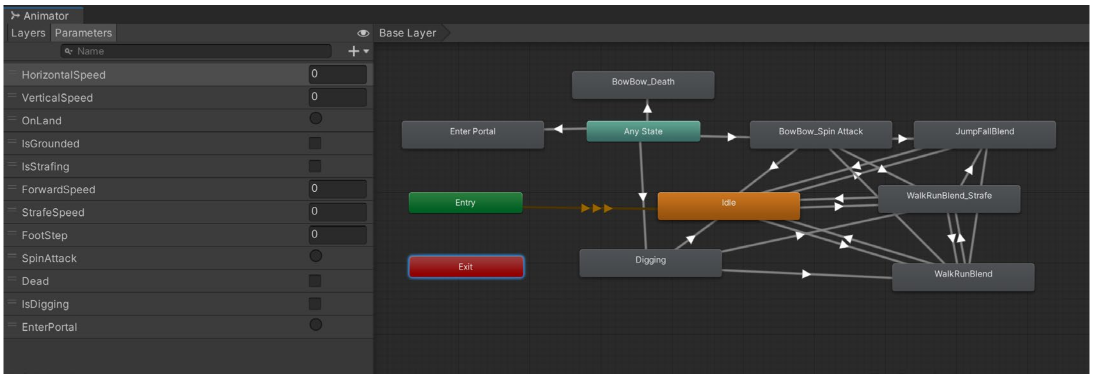


Figure 6.15 – The animator object for Bow Bow

As the Bum Bum character did not contain a regular humanoid rig, I had to create the animations for this character in Blender (Blender, 2021) as shown in figure 6.16, and then import it into Unity (Unity, 2021) and set up an animator object (figure 6.17) using the same procedure as above.

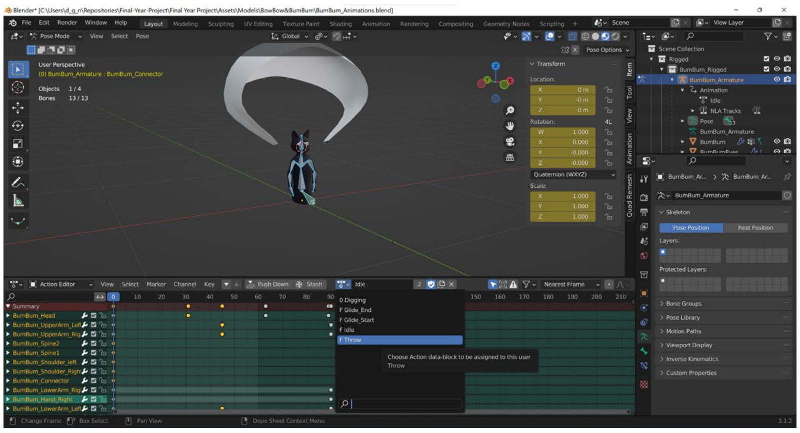


Figure 6.16 – Using Blender to create the Bum Bum character animations

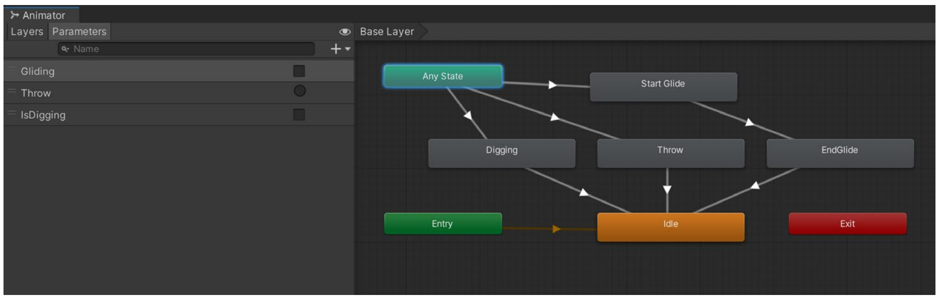


Figure 6.17 – The animator object for Bum Bum

6.1.3. Player Health

The player's health is handled by the PlayerHealth script. This inherits from the Health script that contains a base implementation for health in the game. This is then overridden to allow for the variations in behaviour between the player, enemies and interactable objects (e.g., Levers, Rocks).

The player has a maximum amount of health which can be increased by collecting all collectable letters in a level (see section 6.3.1.4. Collectable Letter). When the player takes damage, the game object's renderers flash for a small time and the player cannot take further damage until they stop flashing. If the player's health level drops below zero, then the death animation is played, controls disabled, and the current level is reloaded.

6.1.4. Character movement

The character movement and camera system were based upon the Character Movement Fundamentals (Ott, 2021) asset from the Unity asset store (Unity, 2022). This control system comprises three main elements.

- **ControlHandler** – This script listens for broadcast messages from the unity input system and then stores boolean variables indicating which keys and buttons have been pressed and released by the player. These values are then checked each frame by the PlayerController and AbilitiesController.
- **AnimationController** – This script listens for movement broadcast messages from the PlayerController and sets animation states as needed.
- **PlayerController** – This script is the main control script and constantly polls the ControlHandler to check if any buttons have been pressed or released in each frame. It then Broadcasts messages to related scripts to trigger player movement, camera movement, animations, and abilities.

6.1.5. Player abilities

The player abilities are handled in the same way as movement using the following scripts.

- AbilitiesController – This script keeps track of which abilities have been activated as well as checks the ControlHandler to see if any abilities have been triggered in each frame. If an ability has been triggered the script notifies the AbilitiesAnimationController.
- AbilitiesAnimationController – This script listens for messages broadcast by the Abilitiescontroller and then sets the animator objects properties as abilities are started and stopped.

The abilities implemented are.

- **Double Jump –** This ability removes the ground checking when jumping once so that the player can jump again when in the air to reach higher or further areas.
- **Spin Attack –** This ability triggers a spin animation which in turn triggers the SpinAttack script to sphere cast damage to nearby enemies.
- **Drop Attack –** This attack is handled by a collider attached to Bow Bows' feet. On collision, the enemy collided with is checked by the DropAttack script, and if it can be damaged it is. If the enemy cannot be damaged, then the player is damaged. The player is then propelled upwards gently to bounce from the enemy.
- **Glide attack –** The glide ability is implemented by animating Bum Bum to hold a parachute and then adjusting the gravity setting in the PlayerController.
- **Steep Slope –** When the steep slope is enabled, the slope limit of the PlayerController script is increased and then reset when the ability is deactivated.
- **Digging –** When digging is active, the relevant animation is played, and a sphere cast is performed using a layer mask to only affect nearby Diggable objects.
- **Shooting and Bombs –** The shooting and bombs implementation is described below in Section 6.1.5 Shooting.

6.1.6. Shooting

Player shooting is handled by the following scripts.

- **Aiming** – This script rotates Bum Bum in the same direction as the camera to aim at enemies and objects.
- **AimCamera** – This script switches between the main game camera and an over the shoulder aim camera.
- **Shooter** – This script stores all data relating to the player's current guns and ammo levels. It also handles shooting the current gun as well as saving and loading ammo levels using the SaveLoad script.
- **PlayerGun** – Inherited from Gun, this script handles storing the gun's ammo level as well as instantiating the appropriate Projectile object when fired.

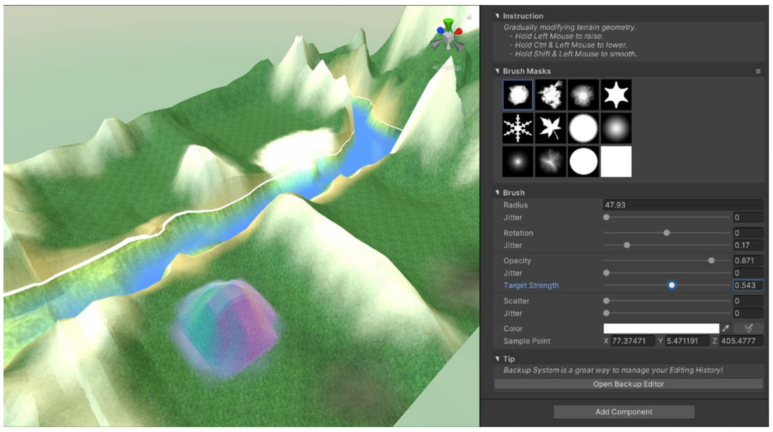
All ammo types inherit from the Projectile class and when instantiated are propelled forward with a given force. The two variants used by the player are,

- **FishProjectile –** The fish projectile is launched with a random rotational force applied and when it contacts an object, an effect and sound are played. If the hit object is an enemy, it is damaged.
- **BombProjectile** – The bomb projectile is launched with a random rotational force applied and when it contacts an object, an explosion effect and sound are played. Damage is then applied to any enemies within the blast radius based upon a customisable falloff curve.

6.2. Level Development

6.2.1. Base Level (Home)

This level's terrain was created using the Polaris Asset (Pinwheel, 2021) from the Unity Asset Store (Unity, 2022) and its associated tools to sculpt the terrain and paint textures upon the terrain as shown in figure 6.18 below.


Instruction
Gradually modifying terrain geometry.
- Hold Left Mouse to raise.
- Hold Ctrl & Left Mouse to lower.
- Hold Shift & Left Mouse to smooth.

Brush Masks

Brush
Radius 47.93
Jitter 0
Rotation 0
Jitter 0.17
Opacity 0.871
Jitter 0
Target Strength 0.543
Scatter 0
Jitter 0
Color
Sample Point X 77.37471 Y 5.471191 Z 405.4777

Tip
Backup System is a great way to manage your Editing History!

Figure 6.18 – Using Polaris to sculpt the level terrain

Once the terrain was sculpted, the bridges, houses, pyramid and fences were modelled and textured in Blender (Blender, 2021) (figure 6.19) before being imported into Unity (Unity, 2021) and placed around the scene.

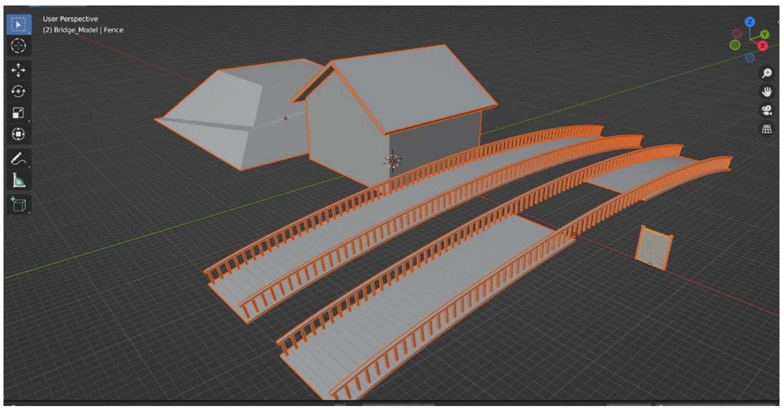
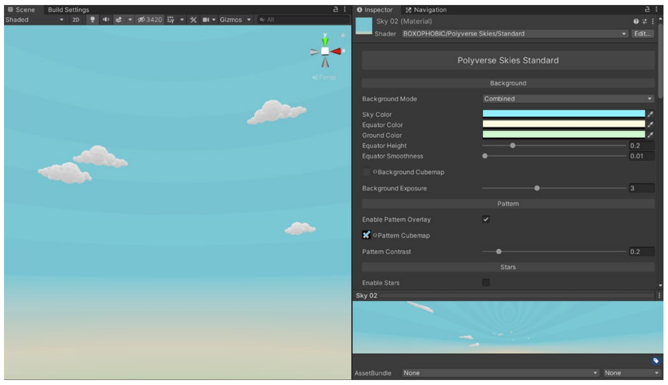


Figure 6.19 – Base Level items modelled in Blender (Blender, 2021)

Next, the skybox for the level was created and adjusted using Polyverse Skies (Boxophobic, 2018) from the Unity Asset Store (Unity, 2022) as shown in figure 6.20 below.



Sky 02 (Material)
Shader BOXOPHOBIC/Polyverse Skies/Standard

Polyverse Skies Standard

Background
Background Mode Combined
Sky Color
Equator Color
Ground Color
Equator Height 0.2
Equator Smoothness 0.01
Background Cubemap
Background Exposure 3

Pattern
Enable Pattern Overlay ✓
Pattern Cubemap
Pattern Contrast 0.2

Stars
Enable Stars

Sky 02

AssetBundle None None

Figure 6.20 – Creating the skybox using Polyverse skies (Boxophobic, 2018)

Next colliders were added to the level using standard box colliders to stop the player from going out of bounds, as well as adding the trees and required level items and enemies within the level using created prefabricated items as shown below in figure 6.21.

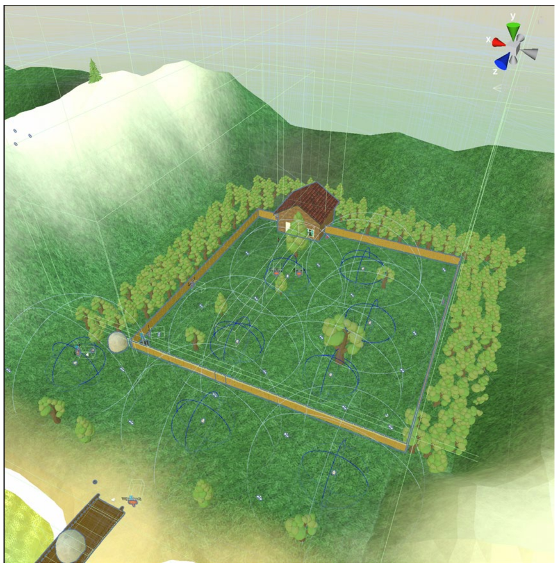


Figure 6.21 – Base level with added prefabricated items and colliders

One thing to note in this level is that I attempted to add dynamic grass using Polaris (Pinwheel, 2021) that was affected by wind and objects as shown in figure 6.22. However, this grass proved extremely resource-heavy and caused some issues with frame rates during gameplay. As a result of the effect on the gameplay experience, this grass was reverted to a basic grass texture.

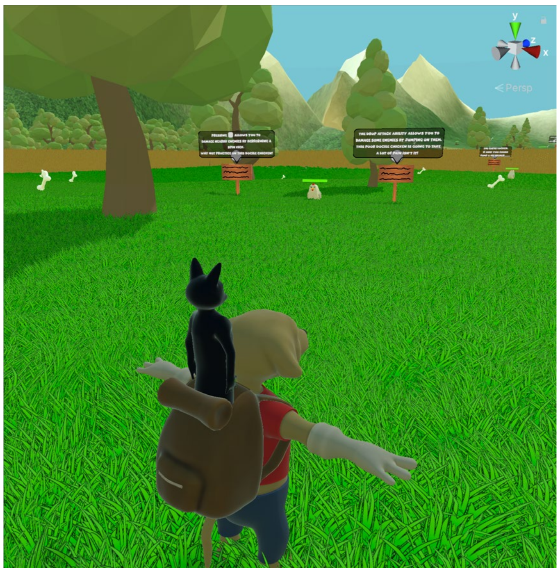


Figure 6.22 – Attempted grass using Polaris (Pinwheel, 2021)

6.2.2. Level One (Deserted Dunes)

This level was created in the same way as the base level but adjusted the layout, textures, and skybox to create a different looking environment.

6.2.3. Level Two (Robot Wars)

This level was intended to feel more enclosed and as such instead of using a terrain object, the level layout was modelled in Blender (Blender, 2021) and created a squarer looking level as shown in figure 6.23 below.

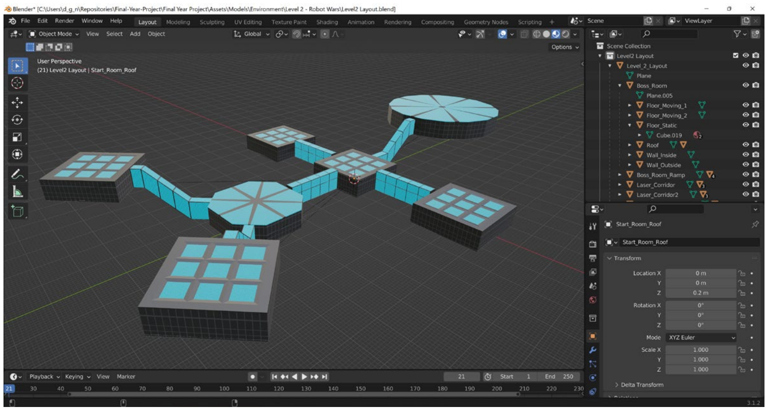


Figure 6.23 – Level two modelling in Blender (Blender, 2021)

Once the level model had been created and imported into Unity (Unity, 2021) the various items were placed around the level from prefab items as described in Section 5.2.1.3.

The gravity switcher in the level has a FlipGravity script. On collision with the player, this script toggles an inverted boolean flag and then notifies the PlayerController of the new inverted state via an Event. The script then reverses its up rotation and in FixedUpdate, lerps the gravity switcher model towards this new rotation until inverted.

When the PlayerController has its GravitySwitch method called, it performs the same action of inverting the up direction of the player game object and then linearly interpolating towards the new rotation as necessary in FixedUpdate. When the player is inverted, they are unable to shoot or glide as Bum Bum freezes and covers his eyes.

The Boss room floor is described in section 5.3.2.3. Level Two (Robot Wars) has been implemented as follows. The scene contains a game object that contains three different, prebaked nav-mesh surfaces. The initial one of these is enabled when the scene is loaded and the stage one opening and stage two nav-meshes are disabled as shown in figure 6.24 below.

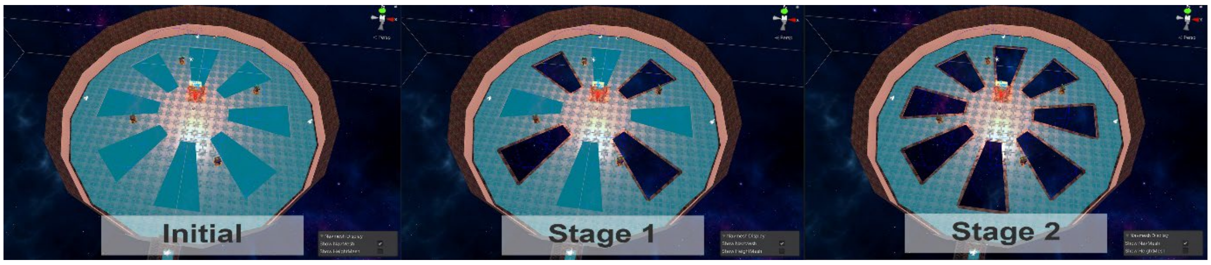


Figure 6.24 – The pre-baked nav-mesh surfaces for the boss room floor animation.

While the BossRoomFloor script is active, it checks the boss's health every second and if it drops below 75% then stage one is activated. Once this happens the nav-mesh is swapped to stop enemies floating across the gaps, then the floor animator is set to stage one, and the floor sound is played. The BossRoomFloor script then continues to check the boss's health again and if it drops below 50%, the nav-mesh is swapped once again, then the floor animator is set to stage two and the floor sound is played once more. The BossRoomFloor script is then disabled.

6.3. Level Items Development

This section describes the development of the various items used within the levels.

6.3.1. Collectables

All collectable items within the game inherit from a main Collectable script as shown in the class diagram in figure 6.25 below. This class handles a basic implementation for playing effects and sounds, disabling the Collectable and respawning it as necessary. This class can then be overridden to add extra collection actions for each collectable. After collecting any Collectable, the game is saved using the SaveLoad class.

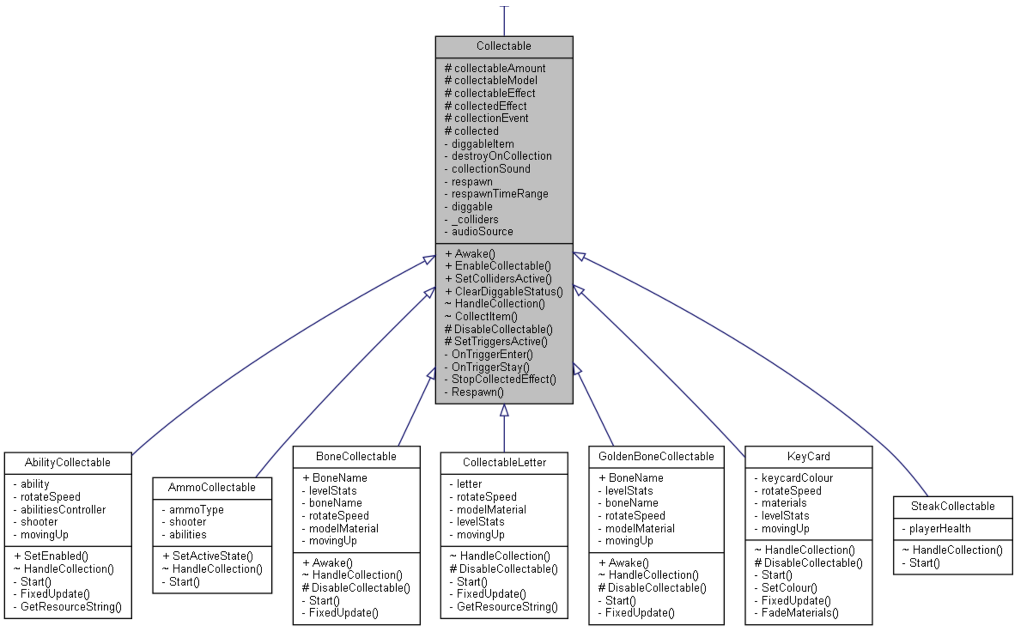


Figure 6.25 – The inheritance class diagram for the Collectable class

6.3.1.1. Ability Collectable

When the AbilityCollectable is instantiated, if the player already has the ability then the game object is destroyed. If not the correct model for the collectable is loaded from the resources folder using the ability enumeration. When the item is collected, the ability is added using the AbilitiesController class which in turn saves the progress of the game.

6.3.1.2. Ammo Collectable

When the AmmoCollectable has instantiated the correct model for the collectable is loaded from the resources folder using the ammo enumeration. When the item is collected, the Shooter class is checked and if the associated ammo type is not full then the ammo amount is added, the collectable is collected and game progress saved.

6.3.1.3. Bone Collectable

When the BoneCollectable is instantiated, it is first assigned a unique name using the current level and the bone's position. If the player has already collected the bone, then the game object is disabled. When the bone is collected the LevelStats class is notified and passed which bone has been collected. The LevelStats then saves the game progress and updates the UIController script.

6.3.1.4. Collectable Letter

When the CollectableLetter is instantiated, first the correct model is loaded from the resources folder using the letter enumeration, then if the letter has already been collected it is disabled. When the letter is collected the LevelStats class is notified and passed which letter was collected. The LevelStats then checks if all letters have been collected for that level and increase the player's maximum health if they have. It then saves the game progress and updates the UIController script.

6.3.1.5. Golden Bone Collectable

When the GoldenBoneCollectable is instantiated if the player has already collected the golden bone then the game object is disabled. When the golden bone is collected the LevelStats class is notified which in turn saves the game progress and updates the UIController script.

6.3.1.6. Key Card

When the KeyCard is instantiated its materials and effects are first set to the correct colour based upon the key card enumeration. Then if the player already has the key card's colour, the item is disabled. On collection, the LevelStats class is notified and passed which key card was collected. The then saves the game progress and updates the UIController script.

6.3.1.7. Steak Collectable

When the steak collectable is collected, the PlayerHealth script is checked and if the health is not full, the health amount is added and the item is collected.

6.3.2. Portals

The portal was modelled in Blender (Blender, 2021). When Instantiated, the Portal loads the stats for the level it leads to, then sets the sign above it to display the details from this loaded data as shown in figure 6.26 below. This includes the level name, number of bones collected, letters collected and golden bone collection status.

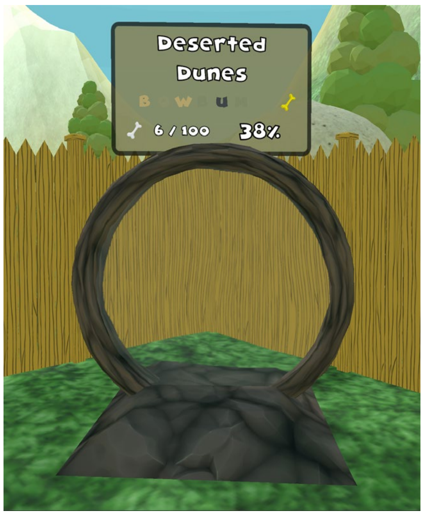


Figure 6.26 – The portal and level progress display

When the player enters the Portal its material parameters are adjusted to produce a ripple effect and a teleport animation is triggered on the PlayerController. The Portal then notifies the SceneLoader singleton script to trigger a fade-out screen and load the level stored in the portals level enumeration variable.

6.3.3. Explodable Rock

The Explodable rock object was modelled and textured in Blender (Blender, 2021), a tool called cell fracture to create a second collection of meshes to represent the rock after being split up into pieces as shown in figure 6.27.

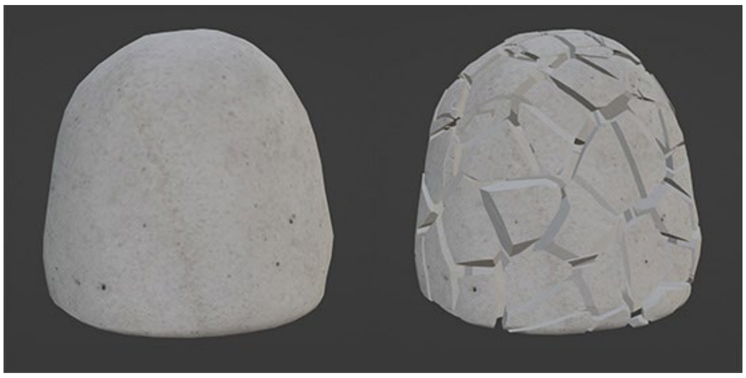


Figure 6.27 – The explodable rock before and after using cell fracture

Each of these smaller parts has a rigid body component added to them so that they react physically with each other and the environment. When the ExplodableRock is instantiated it first checks if the item has previously been destroyed using the LevelStats script and if so destroys itself. If the item has not been destroyed, then the whole rock model is left enabled, and the cell models are all disabled. If the object is activated by a sphere cast from the BombProjectile script, then the large rock is disabled, the small rocks are enabled and begin reacting to physics within the world as shown in figure 6.28 below. The ExplodableRock script then notifies the LevelStats to save the object being destroyed before flashing the renderers and destroying the game object after two seconds

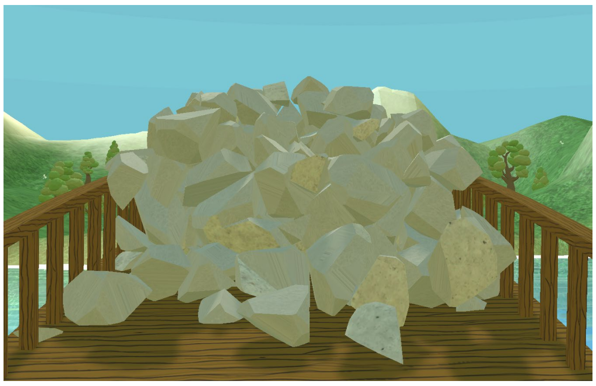
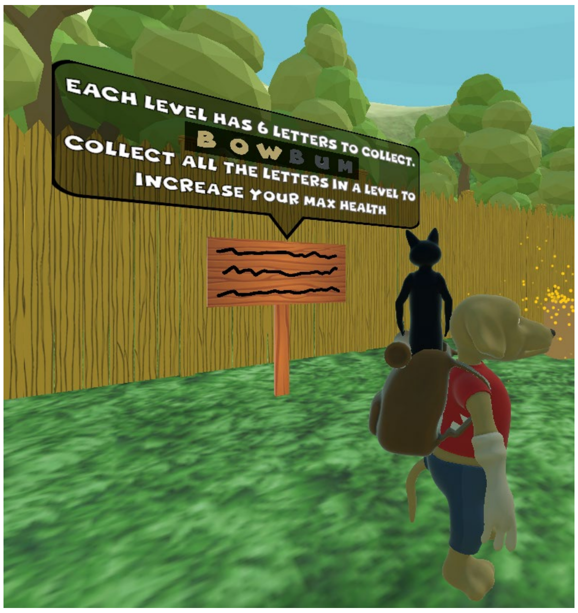


Figure 6.28 – After activation

6.3.4. Signposts

The Signposts used to instruct the player as shown in figure 6.29 below, were modelled and textured in Blender (Blender, 2021).


EACH LEVEL HAS 6 LETTERS TO COLLECT.
B O W B U M
COLLECT ALL THE LETTERS IN A LEVEL TO INCREASE YOUR MAX HEALTH
Figure 6.29 – The instructional signpost

When the player enters the Signpost trigger, a ControlIcon is displayed showing the control used to activate the signpost text. Upon pressing the interact button the signpost text is displayed. Upon the player leaving the trigger area the sign text is hidden again.

All ControlIcon objects in the scene are automatically updated to reflect the current control scheme by the ControlHandler script when it receives the OnControlsChanged message from the Unity (Unity, 2021) input system.

6.3.5. Moving Platforms

6.3.5.1. Platforms

The moving platforms and activation levers as shown in figure 6.30 below, were modelled in Blender (Blender, 2021).

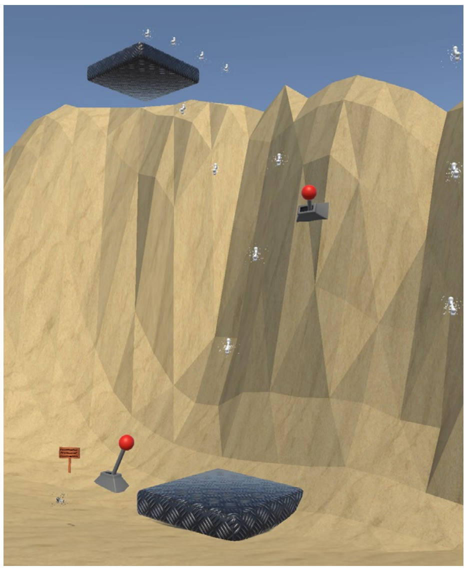
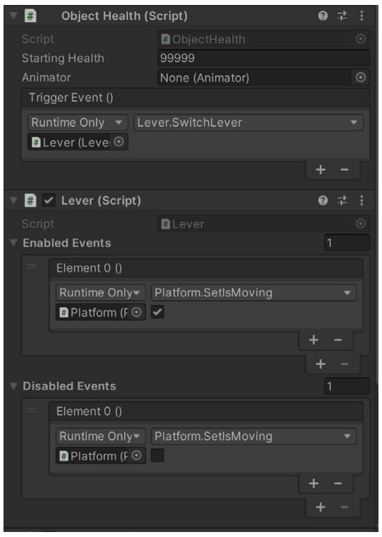


Figure 6.30 – Moving platforms and levers

Each Platform has a series of empty game objects placed in a path. When activated the platform moves toward the next point in the path. When the platform reaches the next path point, it pauses for the time set before updating the next point in the path and continuing. When the end of the path is reached, a boolean trigger is used to reverse the path and travel back towards the start point again. The platform also has a PlatformTrigger script that tracks the player entering and exiting the platform and updates the player's position by the correct amount each frame when on the trigger.

6.3.5.2. Levers

The Lever script has an Objecthealth script that detects damage from the player's projectiles and spin attack and switches the lever's animator state before invoking any Unity events set in the inspector for enabling and disabling the lever. Using this approach, it was easy to add a lever to a scene and simply drop the object that you wanted it to trigger within the Unity editor as shown in figure 6.31 below.



```
Object Health (Script)
Script             ObjectHealth
Starting Health    99999
Animator           None (Animator)
Trigger Event ()
  Runtime Only     Lever.SwitchLever
  Lever (Leve
Lever (Script)
Script             Lever
Enabled Events                      1
  Element 0 ()
    Runtime Only   Platform.SetIsMoving
    Platform (F    ✓
Disabled Events                     1
  Element 0 ()
    Runtime Only   Platform.SetIsMoving
    Platform (F
```

Figure 6.31 – The Object health and Lever elements in the inspector.

6.3.6. Lasers

6.3.6.1. Laser Setup

The Laser system is configured as follows, a LaserGroup contains several LaserArray objects. These in turn hold a series of Laser objects which have an animator for turning them on and off as well as a LaserDamage script to kill the player on contact as shown in figure 6.32 below.

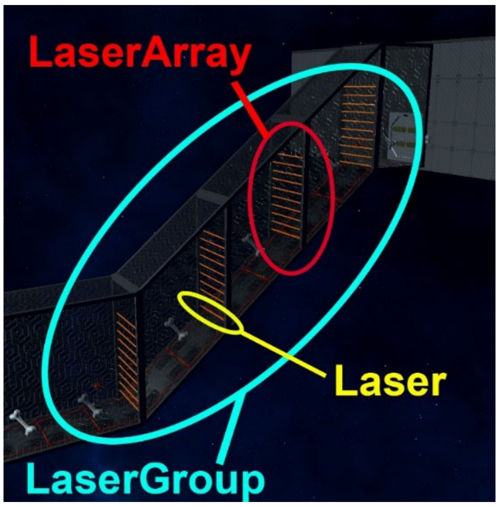


Figure 6.32 – The Laser setup hierarchy

The Laser script just contains methods for turning that individual laser on and off using the animator parameters. The LaserArray then has the same two methods to call the on and off methods for each Laser in the array simultaneously. The LaserGroup contains a pattern enumeration that is used to switch on and off each LaserArray in turn depending upon the pattern chosen using on and off times defined in the inspector. The patterns implemented are as follows,

- **All On -** All lasers are continuously enabled
- **All On Off –** All lasers are switched on and off together
- **One Off –** All arrays are switched on except for one that is cycled as shown in figure 6.33 below.
- **One On –** All arrays are switched off except for one that is cycled as shown in figure 6.34 below.

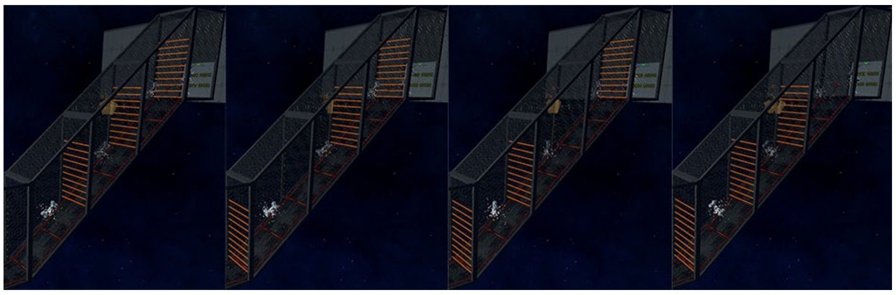


Figure 6.33 – The one off laser group pattern

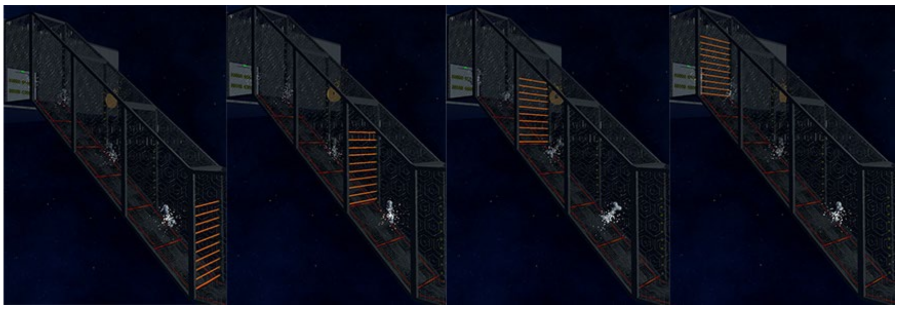


Figure 6.34 – The one on laser group pattern

6.3.7. Puzzles

6.3.7.1. Button Puzzle

The ButtonPuzzle as shown below in figure 6.35, was modelled in Blender (Blender, 2021).

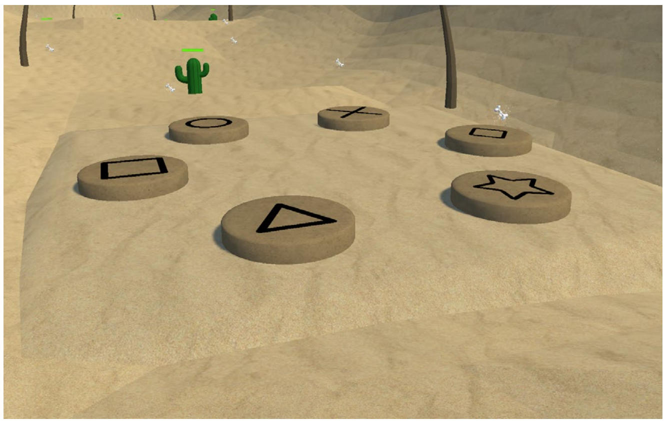


Figure 6.35 – The button puzzle

The ButtonPuzzle script contains two arrays of PuzzleButton objects, one stores a reference to the buttons and the other stores the correct sequence for the puzzle. The sequence can be either set in the editor, or a boolean flag can cause it to be randomised on instantiation. When a PuzzleButton is triggered by the player standing on it, it begins moving down. If it reaches a position threshold set relative to its starting position, it notifies the ButtonPuzzle that the button has been pressed. If the PuzzleButton is the next in the sequence, it is coloured green and remains pressed down. If it is not, all buttons are turned red and moved back to their starting positions. When a button is correctly pressed the ButtonPuzzle checks if all buttons have been pressed and if they have a completion Event is triggered that can be set in the inspector (e.g. to activate a Collectable letter).

6.3.7.2. Lever Puzzle

The LeverPuzzle as shown in figure 6.36 below, contains an array of eight Lever objects (see section 6.3.5.2. Levers above) that must be switched on and off to match a given pattern displayed on a sign.

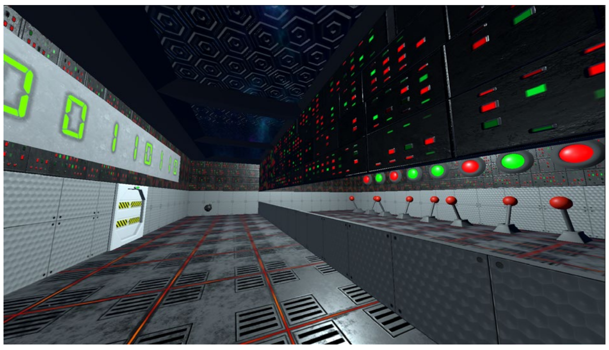


Figure 6.36 – The Lever puzzle and sign

On instantiation, the LeverPuzzle script randomises the sequence required and then prints it to the sign attached to the puzzle. The font used on the sign was Digital-7 (Style-7, 2008) from dafont.com (DaFont, 2022). When a Lever is switched, it notifies the LeverPuzzle script which then sets the RoundLight above the lever according to the Lever state. It then checks the array of levers against the boolean sequence array and if the lever array matches the sequence array a completion event is invoked (e.g., to enable a key card).

6.4. Enemy Development

6.4.1. Modelling

Below are the methods used to model and animate each enemy

6.4.1.1. Chicken

The chicken model and animations were from the Free Chicken Mega Toon Series (Meshint, 2019). An animator was then created as shown in figure 6.37 below.

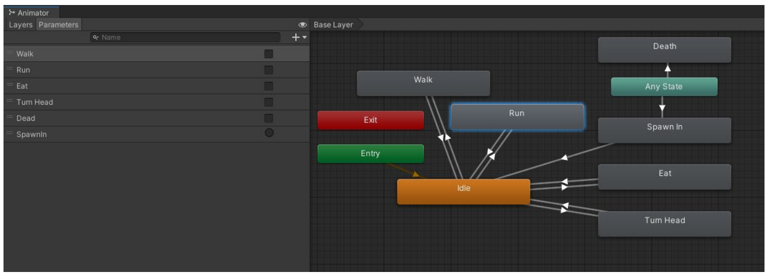


Figure 6.37 – The chicken enemy animator

6.4.1.2. Scorpion

The scorpion model and rig were downloaded from kindingerlek (kindingerlek, 2016) on Sketchfab (Sketchfab, 2022). The animations were not quite right for those required so the model was imported into Blender (Blender, 2021) and re-animated as shown in figure 6.38 below, before being imported into Unity (Unity, 2021). An animator was then created as shown in figure 6.39 below.

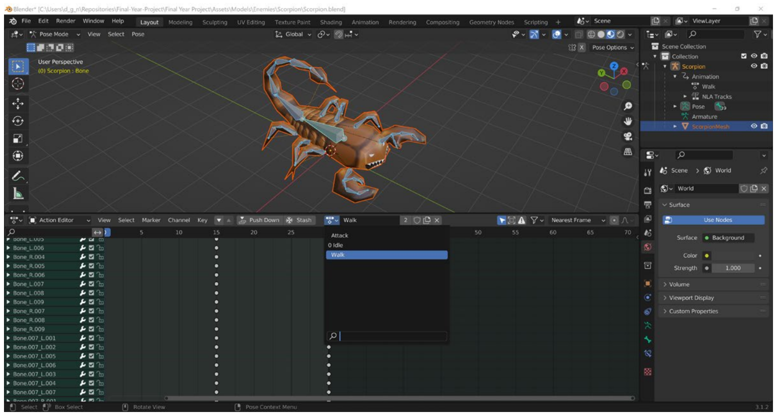


Figure 6.38 – Reanimating the scorpion in Blender (Blender, 2021)

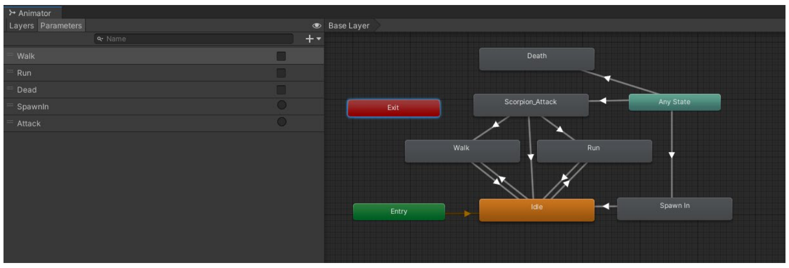


Figure 6.39 – The scorpion enemy animator

6.4.1.3. Cactus

The cactus enemy was modelled and textured in Blender (Blender, 2021) as shown in figure 6.40 below. As the animation for this enemy was extremely simple, and just involved rotating one element, I created this in Unity (Unity, 2021). An animator was then created as shown in figure 6.41.

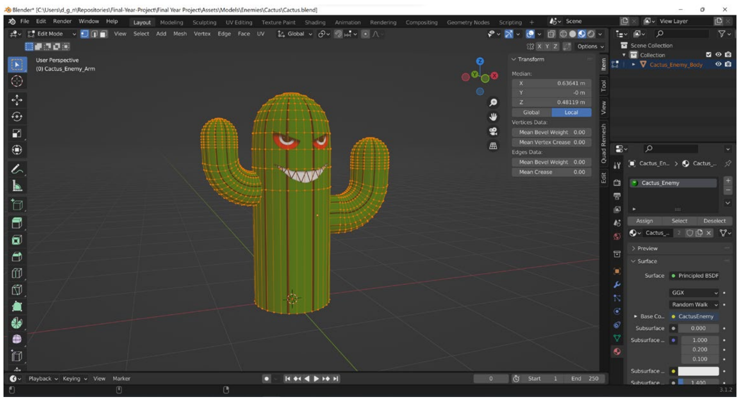


Figure 6.40 – Modelling the cactus enemy in Blender (Blender, 2021)

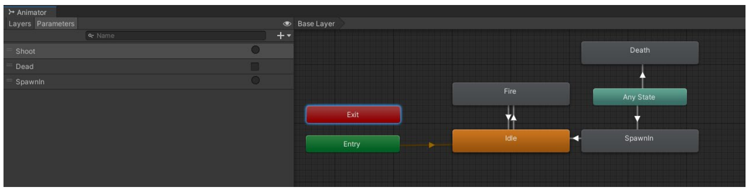


Figure 6.41 – The cactus enemy animator

6.4.1.4. Level One Boss – Doozy (Adobe, 2021)

The level one boss (Doozy) enemy and animations were downloaded from the Mixamo (Adobe, 2021) website. A wooden club was also modelled in Blender (Blender, 2021) for the enemy to use as a weapon. An animator was then created as shown in figure 6.42.

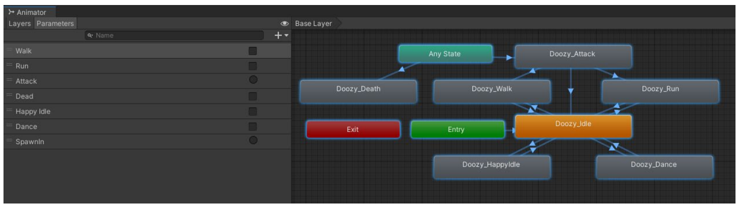


Figure 6.42 - The level one Boss enemy (Doozy) animator

6.4.1.5. Robot

The robot enemy was modelled and textured in Blender (Blender, 2021). The animations were then created within Unity (Unity, 2021). Finally, an animator was then created as shown in figure 6.43.

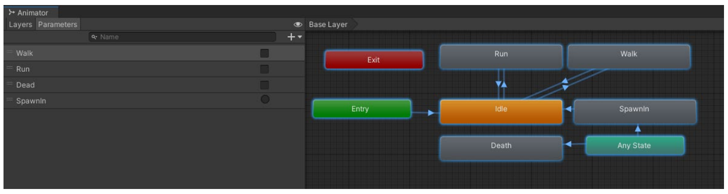


Figure 6.43 – The robot enemy animator

6.4.1.6. Turret

The turret enemy was modelled and textured in Blender (Blender, 2021). A simple shoot animation and animator were then created as shown in figure 6.44 below. The shoot animation also used an animation event to trigger shooting a projectile.

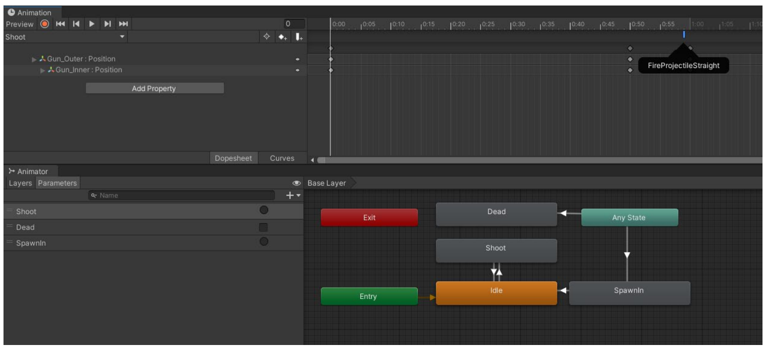


Figure 6.44 – The turret enemy animation and animator

6.4.1.7. Large Turret

The large turret enemy was modelled and textured in Blender (Blender, 2021). A simple shoot animation and animator were then created as shown in figure 6.45 below. The shoot animation also used an animation event to trigger shooting a projectile.

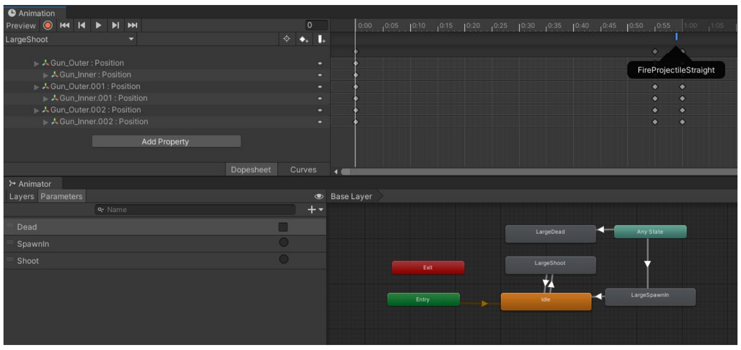


Figure 6.45 – The large turret enemy animation and animator

6.4.1.8. Level Two Boss (YBot) (Adobe, 2021)

The level two boss (YBot) enemy and animations were downloaded from the Mixamo (Adobe, 2021) website. An animator was then created as shown in figure 6.46.

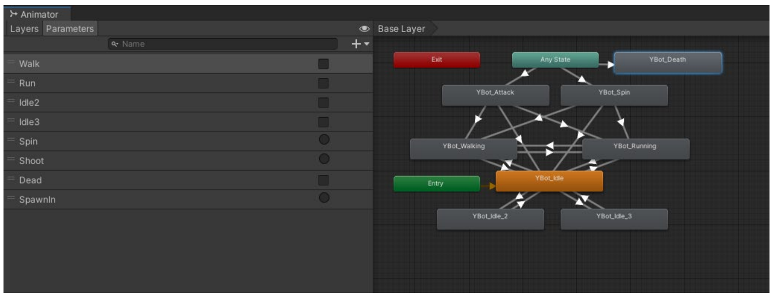


Figure 6.46 - The level two Boss enemy (YBot) animator

6.4.2. Base Enemy Implementation

An EnemyAI and EnemyController script contains a base implementation for all enemies. These scripts are then overridden by MobileEnemyAI / TurretEnemyAI and MobileEnemyController / TurretEnemyController scripts depending upon the type of enemy. Finally, these variant scripts are overridden to provide an implementation for each enemy.

The EnemyAI script implements a finite state machine which checks the current state at set intervals and then performs a think action for that state and changes the state if necessary. Finally, the behaviour of the current state is performed. In the base implementation, all Methods are empty to be overridden by the inheriting classes.

The EnemyController contains base implementation to spawn and respawn the enemy, damage the enemy on contact if enabled and take damage/death behaviour.

The MobileEnemyAI script configures the EnemyAI to find a valid waypoint using the Unity (Unity, 2021) nav-mesh system and walk towards it. When the enemy reaches the waypoint, it Idles briefly and then repeats this pattern. If at any point the enemy gains line of sight to the player (see section 6.4.2 Line Of sight below), then the enemy continuously runs toward the player until the line of sight is lost. If a mobile enemy takes damage it reacts by switching to the follow player state.

The MobileEnemyController script adds the ability to the EnemyController to set the walk and run animations, adjusts the agent speed and sets a waypoint position for the nav-mesh agent.

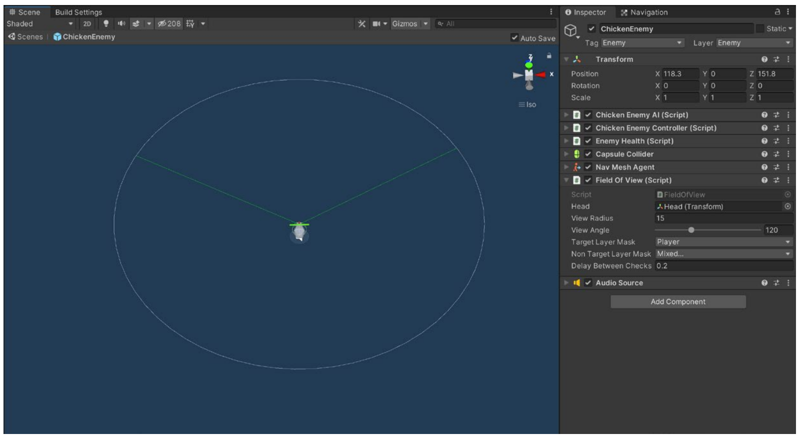
The TurretEnemyAI script configures the EnemyAI to either continually rotate or rotate to random rotations depending upon the setting made in the editor. If at any point the enemy gains line of sight to the player (see section 6.4.2 Line Of sight below), the enemy is continually rotated towards the player and when facing them shoot. If the line of sight is lost, then the enemy returns to idle.

The TurretEnemyController adds the ability to the EnemyController to shoot at the player.

6.4.3. Line Of Sight

The FieldOfView script was created with the help of a video tutorial by Comp-3 Interactive (Comp-3, 2021). The script has a reference to an empty head object on the enemy's head and periodically updates its playerVisible boolean variable by calculating the following. First, the script checks if the player is in range and if not sets playerVisible to false and returns. If the player is in range, then the script checks if the angle difference between the head objects forward vector and the player is within the set field of view angle. And set playerVisible to true if it is and false if not.

This object also has a Gizmo that is displayed in the editor to easily see an enemy's field of view setting as shown in figure 6.47 below.



ChickenEnemy Static
Tag Enemy Layer Enemy

Transform
Position X 118.3 Y 0 Z 151.8
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

✓ Chicken Enemy AI (Script)
✓ Chicken Enemy Controller (Script)
✓ Enemy Health (Script)
✓ Capsule Collider
✓ Nav Mesh Agent
✓ Field Of View (Script)

Script	FieldOfView
Head	Head (Transform)
View Radius	15
View Angle	120
Target Layer Mask	Player
Non Target Layer Mask	Mixed...
Delay Between Checks	0.2

✓ Audio Source

Figure 6.47 – The Field of view script visualisation and settings

6.4.4. Enemy script variations

Described below is how each enemy's AI and Controller scripts differ from their overridden classes.

6.4.4.1. Chicken

The Chicken enemy as described in 5.3.2.1. Base Level (Home) above, has a ChickenEnemyAI script that utilises the MobileEnemyAI behaviour as it is. The enemy also has a ChickenEnemyController script that extends the MobileEnemyController script to randomly choose between three idle animations (eating, looking around and standing still).

6.4.5. Scorpion

The scorpion enemy as described in 5.3.2.2. Level one (Deserted Dunes) above, has a ScorpionEnemyAI that extends the MobileEnemyAI to add an attack range that triggers the enemy to attack with its tail when entered. The ScorpionEnemyController utilises the MobileEnemyController as it is.

6.4.6. Cactus

The cactus enemy as described in 5.3.2.2. Level one (Deserted Dunes) above, uses the TurretEnemyAI and TurretEnemyController as the implementation requires no changes.

6.4.7. Level One Boss – (Doozy) (Adobe, 2021)

The level one boss (Doozy) has a DoozyEnemyAI script that extends the MobileEnemyAI that adds an attack range to the enemy. If the player enters this range then the boss swings a large wooden club that has a MeleeDamage script attached to damage the player on contact. The enemy also has a DoozyEnemyController script that extends the MobileEnemyController script to randomly choose between three idle animations (Happy idle, dancing and standing still).

6.4.8. Robot

The robot enemy as described in 5.3.2.3. Level Two (Robot Wars) above, has a RobotEnemyAI script that utilises the MobileEnemyAI behaviour as it is. The RobotEnemyController extends the MobileEnemyController to set a single idle animation when required.

6.4.9. Turret

The turret enemy as described in 5.3.2.3. Level Two (Robot Wars) above, uses the TurretEnemyAI and TurretEnemyController as the implementation requires no changes.

6.4.10. Large Turret

The large turret enemy as described in 5.3.2.3. Level two (Robot Wars) above, has a LargeTurretAI script that extends the TurretEnemyAI to rotate the enemy's guns to face the player on three axes rather than just rotating around the Y-Axis before shooting. The LargeTurretController utilises the TurretEnemyController script as it is.

6.4.11. Level Two Boss (YBot) (Adobe, 2021)

The level two boss (YBot) as described in 5.3.2.3. Level Two (Robot Wars) above, has a YBotEnemyAI script that extends the MobileEnemyAI to add the following functionality. When the player is within a set shoot range, the enemy will face the player and shoot a laser bullet at them. Additionally, if the player enters the smaller attack range, then the enemy will perform a spin attack and release a forcefield of energy to damage the player. The YBotEnemyController extends the MobileEnemyController to add the ability to trigger the spin animation as well randomly choose between three idle animations when necessary (Stretching, standing on guard and standing still).

6.5. Game Saving

The game is saved each time a significant item is collected, or a level is exited. This is performed by a main static SaveLoad script.

The game data is stored in a serializable GameSaveSlot class to enable it to be serialized to a file in the user's AppData folder on the player's system. The GameSaveSlot is built up from a PlayerData class, a list of LevelData classes and a TotalGameStats class. When a GameSaveSlot is created all three of these items are initialised to default values. When the game is loaded, if the save files are not present on the users' system, then one is created for each of the six save slots.

When the SaveLoad script is called to save a PlayerData or LevelData item it first loads the current GameSaveSlot from a file by first getting the current save slot index from the SaveSlotManager, before editing the relevant component which automatically recalculates the TotalGameStats item. The file is then resaved to the disk.

When each level is loaded there is a LevelStats object used to track level progress that calls the SaveLoad script to load a LevelData item containing the current levels data. The PlayerController also calls the SaveLoad script to load a PlayerData item containing the player's maximum health, ammo levels, abilities enabled and level checkpoints.

The SaveLoad class also contains methods to reset save slots as well as resetting a level's checkpoints or key cards collected as required.

6.6. UI

The UI in the project contains three main areas, the main menu, pause menu and in-game display.

6.6.1. Font

The font used throughout all UI elements in the project is Cartoon Fun (Flood, 2020) from dafont.com (DaFont, 2022)

6.6.2. Main menu

The main menu as shown in figure 6.48 below, is controlled by the MainMenu script. There is also a persistent display showing the currently selected save slot.

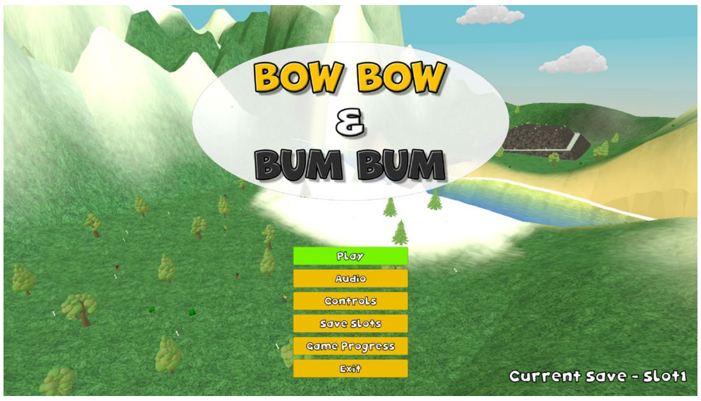


Figure 6.48 – The main menu screen

When the MainMenu is loaded all sub-screens are hidden, the play button is highlighted and the background image is set. A timer is used so that at a fixed interval the background image is switched randomly. The MainMenu also keeps track of which background images have been seen using a list that is cleared once all ten have been viewed. This helps to ensure that images are displayed evenly.

When the play button is selected the SceneLoader class is called to transition to the base level.

When the audio, controls, save slots or progress buttons are selected, the MainMenu script saves which button was highlighted before hiding the main menu screen and displaying the relevant screen. Each of these screens then has its own script to implement its own behaviour as described below. One common element of all these screens is that on exiting they hide themselves and re-enable the main menu screen. This process then prompts the main menu to reselect the saved button to aid in the user experience.

When the exit button is selected the MainMenu script closes the application.

6.6.2.1. Audio screen

The audio settings screen as shown in figure 6.49 below, contains three slider elements for controlling the main volume levels as well as the background music (BGM) and sound effects (SFX) levels.

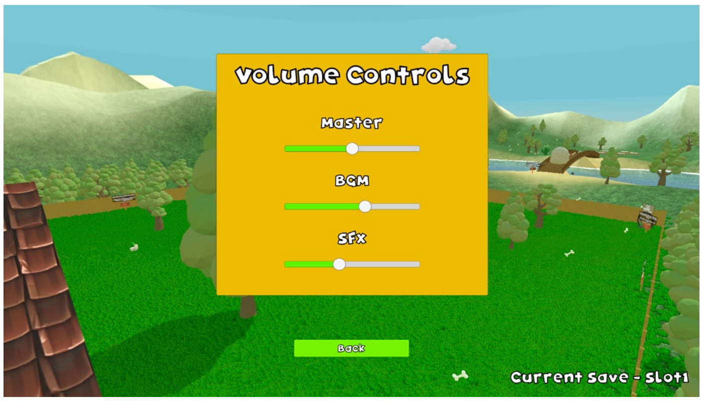


Figure 6.49 – The audio settings menu screen

When the SoundsScreen is loaded, the three sliders are set using values retrieved from the PlayerPrefs system in Unity (Unity, 2021). The relevant audio level is then set by setting an exposed parameter in the project's volume mixer (see section 6.7 Audio below). Before setting to level to the audio mixer the slider value is converted to a logarithmic scale to match the mixer's scale.

When the value of the slider is changed it calls a method to update the relevant mixer's level and save the new value into PlayerPrefs.

6.6.2.2. Controls Screen

The controls screen displays details of the current control layout and changes depending upon the current control scheme that is in use. The UIControlIcon items are updated by the ControlsScreen script when it receives the OnControlsChanges message from the Unity (Unity, 2021) input system as shown in figure 6.50 below.

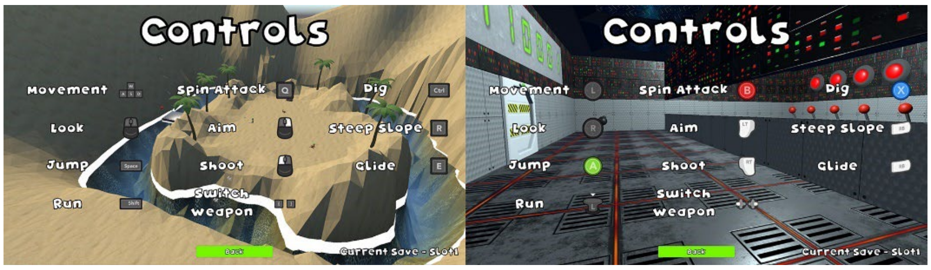


Figure 6.50 – The controls screen shows keyboard and mouse and controller layouts

6.6.2.3. Progress Screen

The game progress screen displays overall game progress for the current save slot as shown below in figure 6.51.

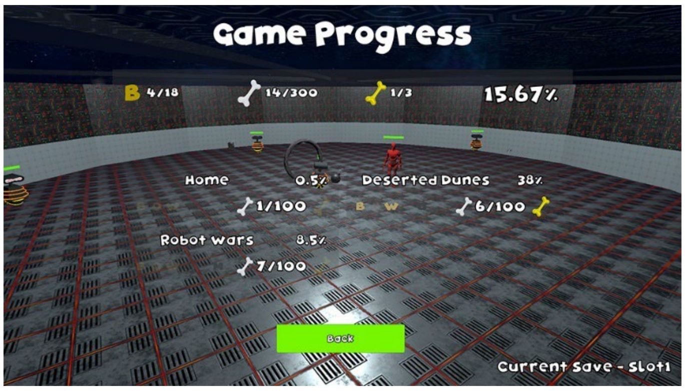


Figure 6.51 – The game progress screen

When the ProgressScreen script loads, it first loads the current save slots data (see section 6.5. Game Saving). Next, the script instantiates prefabricated panels to display the overall game progress and each levels progress and populates the text displays using the loaded data.

6.6.2.4. Save Slot Management

The Save slot management screen as shown in figure 6.52 below, displays details for each of the six save slots as well as allowing resetting and switching of save slots.

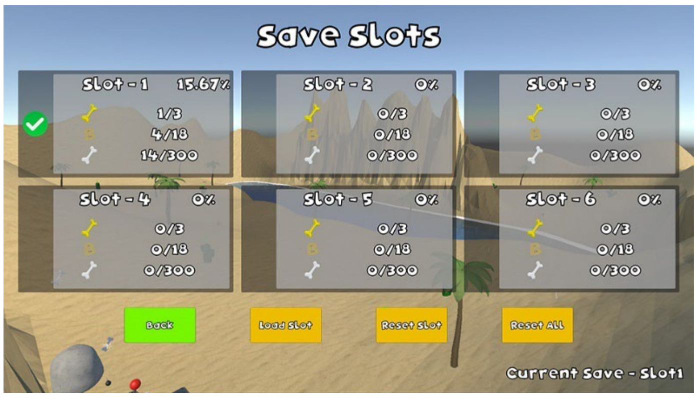


Figure 6.52 – The save slots management screen

When the SaveSlotsScreen script is loaded, it creates a prefabricated toggle panel for each of the six save slots and populates the text elements by loading that slots data (see section 6.5. Game Saving) and the current active save slots toggle is activated. The player can then select a different save slot to change the toggle to that slot before using the load and reset slot buttons to either switch to that save slot or reset it. The reset all button can be used to reinitialise all six save slots.

6.6.3. In-Game Display

The in-game display consists of an overlaid display as shown in figure 6.53 below. This display hides itself after a short period of inactivity and is reshown when any element is updated.

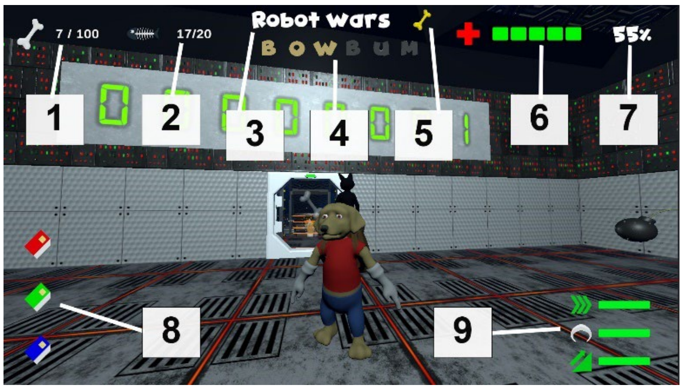


Figure 6.53 – The main game UI

The elements highlighted in figure 6.53 are described below.

1. **Bone display –** The number of bones collected and the maximum number of bones in the level.
2. **Ammo display –** The ammo type currently in use and the current and maximum ammo level.
3. **Level name display –** The name of the current level.
4. **Letter collectable display –** Which letters have been found in the current level.
5. **Golden bone display –** Whether or not the golden bone has been found in the current level.
6. **Health display –** The current and maximum health level.
7. **Level progress display –** The current level's completion percentage.
8. **Key card display –** Which key cards have been collected in the level.
9. **Abilities display –** Appear when an ability is in use or charging to show the current charge and cooldown time for the abilities.

Updating all UI displays as well as displaying the pause menu when the pause key is pressed are handled using the UIController class.

6.6.3.1. Pause menu

The pause menu as shown below in figure 6.54 is activated by the UIController class which pauses the game before displaying the pause menu. The PauseMenu script then controls navigation through the menu screens.

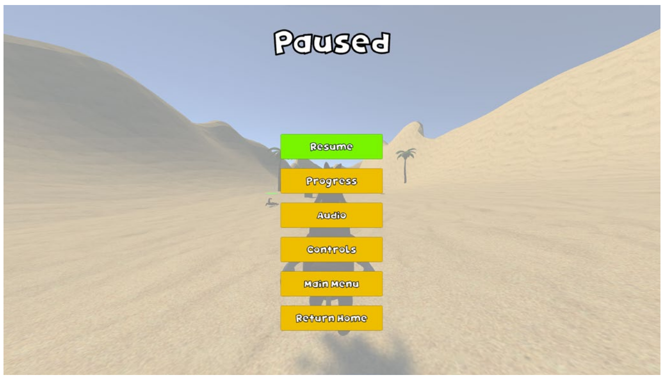


Figure 6.55 – The pause menu

The PauseMenu script controls the screens in an identical manner to the main menu (see section 6.6.2. Main Menu). The pause buttons perform the following actions.

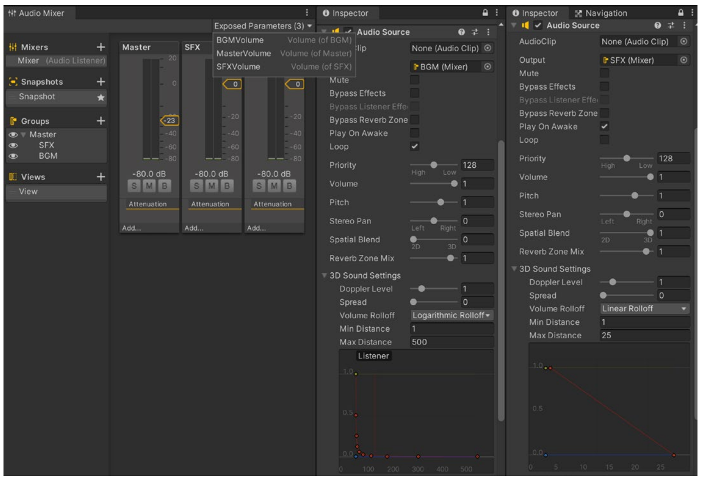
- **Resume –** Returns to the current level being played.
- **Progress, Audio and Controls –** perform the same behaviour as their counterparts in the main menu (see section 6.6.2. Main Menu).
- **Main menu –** Transitions to the main menu.
- **Return home -** Transitions to the base level (home).

6.7. Audio

Audio in the game is divided into two main categories, background music and sound effects

6.7.1. Audio Mixer

The audio volume in the project is controlled by a created AudioMixer Object with exposed channels for master volume, background music and sound effects. Each AudioSource then has its output set to the relevant channel so that its volume can be controlled. This setup is shown in Figure 6.56 below.



```
Audio Mixer
Exposed Parameters (3)
BGMVolume        Volume (of BGM)
MasterVolume     Volume (of Master)
SFXVolume        Volume (of SFX)

Mixers
  Mixer (Audio Listener)
Snapshots
  Snapshot
Groups
  Master
    SFX
    BGM
Views
  View

Master   -80.0 dB   Attenuation   -23
SFX      -80.0 dB   Attenuation   0
BGM      -80.0 dB   Attenuation   0
```

```
Inspector - Audio Source
AudioClip           None (Audio Clip)
Output              BGM (Mixer)
Mute                ☐
Bypass Effects      ☐
Bypass Listener Effects ☐
Bypass Reverb Zone  ☐
Play On Awake       ☐
Loop                ✓
Priority            128
Volume              1
Pitch               1
Stereo Pan          0
Spatial Blend       0
Reverb Zone Mix     1
3D Sound Settings
  Doppler Level     1
  Spread            0
  Volume Rolloff    Logarithmic Rolloff
  Min Distance      1
  Max Distance      500
```

```
Inspector - Audio Source
AudioClip           None (Audio Clip)
Output              SFX (Mixer)
Mute                ☐
Bypass Effects      ☐
Bypass Listener Effects ☐
Bypass Reverb Zone  ☐
Play On Awake       ✓
Loop                ☐
Priority            128
Volume              1
Pitch               1
Stereo Pan          0
Spatial Blend       1
Reverb Zone Mix     1
3D Sound Settings
  Doppler Level     1
  Spread            0
  Volume Rolloff    Linear Rolloff
  Min Distance      1
  Max Distance      25
```

Figure 6.56 – The audio mixer object with background/sound effect AudioSource examples

6.7.2. Background Music

Background music is handled in-game by a singleton BGMManager class. This class persists between scene changes and uses the SceneChanged message broadcast by Unity's (Unity, 2021) SceneManager class to detect scene changes.

If the new scene loaded is different to the one transitioned from, then the level has been changed and the background music is set using an AudioClip from an array that matches the scenes build index. If the scene transitioned to and from are the same, then the player has died and respawned, so the music is not restarted.

6.7.3. Sound Effects

All sound effects in the game are attached to the individual game objects and then triggered by various scripts on each object. The object's AudioSource has its spatial setting to 3D and the range adjusted. This creates a 3D stereo effect within the game by automatically adjusting the left and right channel volumes to create an illusion of 3D sound.

6.7.4. Audio Used

Below is a list of all sounds used in the game and their related references. All sounds used were downloaded from Freesound.org (Freesound, 2022) under creative commons licenses.

- **Menu BGM** - mario's way.mp3 (xsgianni, 2017)
- **Home BGM -** Bluesy Hip Hop Loop (DDmyzik, 2021)
- **Desert BGM -** Desert background music (Sunsai, 2018)
- **Robot Wars BGM -** Atmospheric Ambiance Loop #2 (Sirkoto51, 2016)
- **Take Damage -** Chihuahua Puppy Whine (AustinXYZ, 2016)
- **Throw -** throw.wav (marchon11, 2019)
- **Drop Attack -** Boing.wav (juskiddink, 2012)
- **Laser Shoot -** Laser05.wav (jeremysykes, 2016)
- **Boss Spin Attack -** wide forward woosh1.wav (newagesoup, 2017)
- **Spin Attack -** Whoosh (qubodup, 2008)
- **Glide -** wind.loop.ogg (xUMR, 2019)

- **Digging -** sfx-leaves-1.wav (davilca, 2012)
- **Collect Item -** OKAY! (Scrampunk, 2016)
- **Collect Bone -** bone shell (blukotek, 2014)
- **Ammo Pickup -** Ammo pickup #2 (zivs, 2018)
- **Health -** Chewing, Carrot, A.wav (InspectorJ, 2017)
- **Portal -** transport 2.wav (tim.kahn, 2011)
- **Fish Impact -** Wet Splat (JustInvoke, 2018)
- **Bomb Impact -** Nổ 4 (SieuAmThanh, 2022)
- **Cactus Spine Impact -** Dartboard 5.aif (tonnonic, 2016)
- **Laser Hit -** Impact_007a.wav (AlienXXX, 2013)
- **Chicken -** Chicken Single Alarm Call (Rudmer_Rotteveel, 2015)
- **Robot -** Retro video game sfx - R2D2 (OwlStorm, 2017)
- **Laser -** Laser (jmayoff, 2014)
- **Sliding Door -** SCI_FI_DOOR.wav (alexo400, 2020)
- **Siren -** Sci-Fi Alarm (Daleonfire, 2021)
- **BossRoomFloor -** airlock_extended.wav (primeval_polypod, 2012)
- **Steep Slope -** Wind Chimes Loop (Seidhepriest, 2012)

7.0. Playtesting

Playtesting was conducted in two stages, each time a group were given the current build of the project to play with very little instruction at the start just to give the context of the story. I then left the individual to play the game whilst noting any issues that arose.

Each member was then asked to anonymously complete an online questionnaire. Below is a summary of the results from these questionnaires as well as descriptions of resulting changes from the playtest sessions.

7.1. Playtest one – 16th March 2022

This playtest was completed before the third level was implemented and used to gauge all aspects of gameplay as well as help decide on what theme to use for the third level.

7.1.1. Questionnaire responses

For playtest one I received nine responses to the questionnaire.

7.1.1.1. Question 1 – Main Menu

"How did you find the Main Menu Layout?" (0 = Poor – 10 = Great)

The chart shown in figure 6.57 below shows that most individuals found the main menu to be satisfactory.

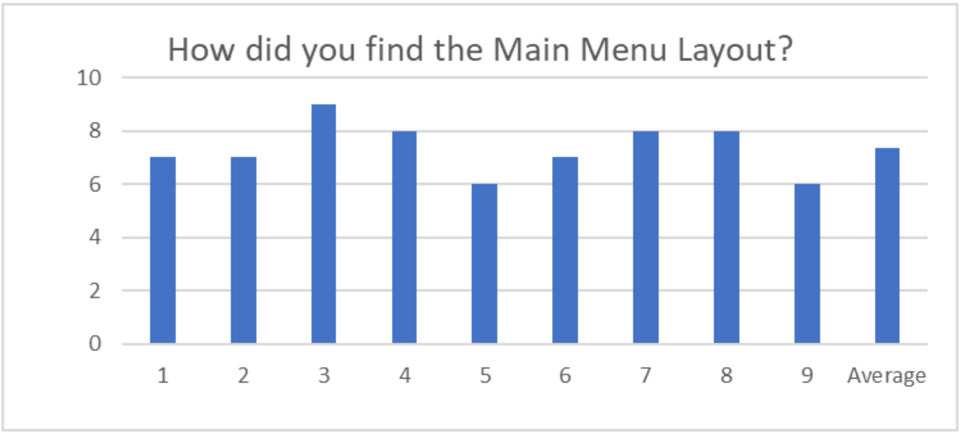


Figure 6.57 - How did you find the Main Menu Layout – Bar chart

7.1.1.2. Question 2 & 3 - Controls

"How did you find the control layout for each control scheme" (0 = Poor – 10 = Great)

The chart shown in figure 6.58 shows that a lot of players were not entirely happy with the control schemes, especially the keyboard and mouse controls.

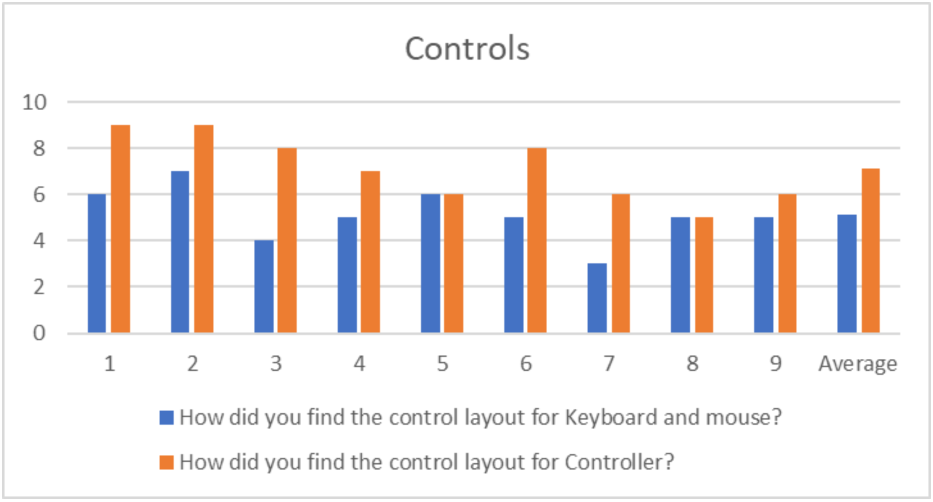


Figure 6.58 – The control scheme questions – Bar chart

The comments gathered alongside this question are listed below.

- sensitivity for keyboard movement is quite high, camera movement with analogue stick could have an increased sense
- sensetivity needs adjusting for aim and camera
- Layout is fine but camera to sensitive for mouse
- I like the controller layout but camera movement is to slow
- Look needs sensitivity adjustment

7.1.1.3. Questions 4 & 5 – Main Characters
"How did you like the aesthetics/animation of the main character" (0 = Poor – 10 = Great)

The chart shown in figure 6.59 below shows that the testers were happy with the appearance of the main characters.

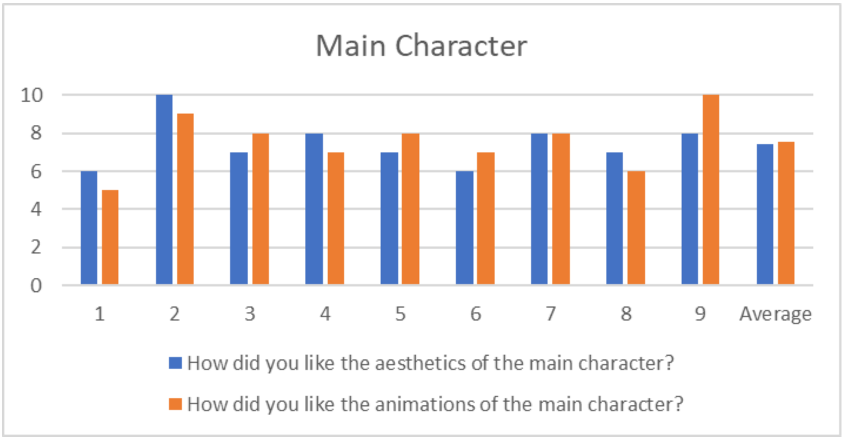


Figure 6.59 – The main character questions – Bar chart

The comments gathered alongside this question are listed below

- The cat is a cool feature - how it looks at the camera angle. It looks a little thin but the animations for it are good. I can tell the model is all good, I think the materials could be a little less shiny like for the shirt -> could make a custom shader for the clothing but that would a lot of effort for that.
- I liked the aesthetics of the main characters, super cool!
- Characters could have more facial expressions, thay have quite blank expressions

7.1.1.4. Question 6
"How did you like the aesthetics of the levels/enemies?" (0 = Poor – 10 = Great)

The chart in figure 6.60 below shows that there were a lot of negative thoughts about the level/enemy aesthetics.

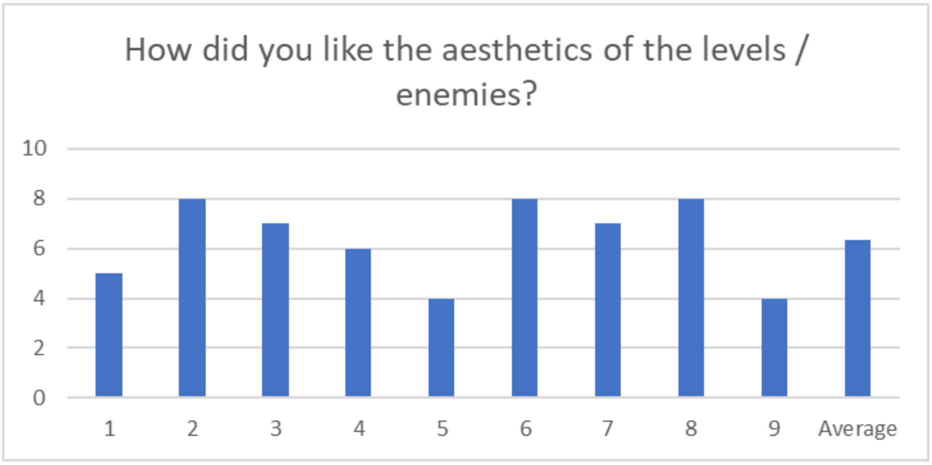


Figure 6.60 – The level/enemy aesthetics question – Bar chart

The comments gathered alongside this are listed below

- The levels feel a little too open, perhaps you could fill the environment with props like boxes, barrels, flora etc. You could also try closing off some areas into rooms. The terrain feels quite late '90s -> low fidelity, low res textures. For example you could shade the landscape smooth and create a better material with a higher res ground texture.
- levels does feel a bit empty
- Could do with more scenery to fill the levels out a bit.
- Loved the water effects.

7.1.1.5. Question 7

"How did you find the proposed story explained at the start of the demo?" (0 = Poor – 10 = Great)

The chart in figure 6.61 below shows that most individuals were happy with the story proposition for the project.

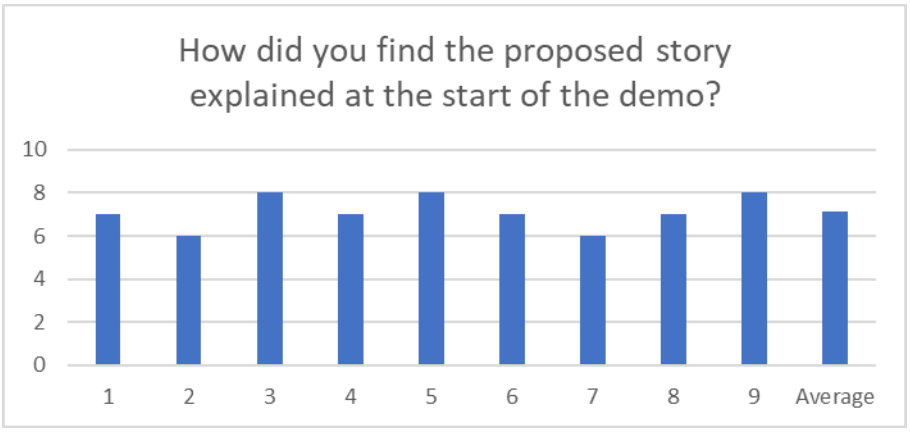


Figure 6.61 – The story question – Bar chart

7.1.1.6. Question 8

"How effective did you find the signpost method of explaining core mechanics?" (0 = Not effective – 10 = Very Effective)

The chart in figure 6.62 below shows that most individuals were happy with the signpost tutorial methods implemented to explain the game's features.

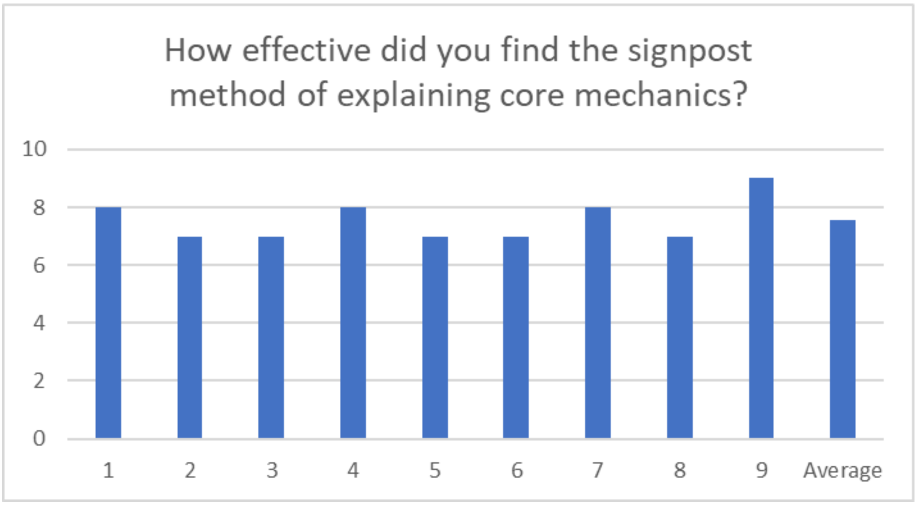


Figure 6.62 – The signpost/tutorial question – Bar chart

7.1.1.7. Question 9
“How did you find the enemy difficulty in the game?” (0 = Too Easy – 10 = Too Hard)

As shown in figure 6.63 below, this question prompted a wide range of responses.

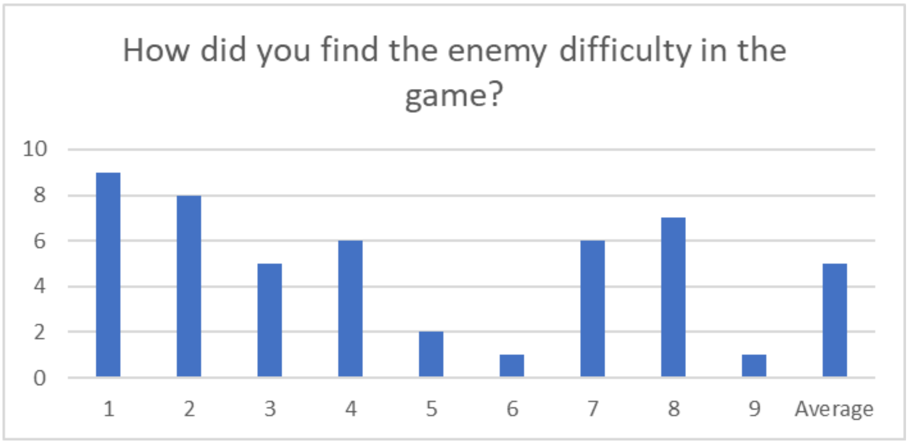


Figure 6.63 – The enemy difficulty question – Bar chart

The comments gathered alongside this question are listed below.

- Boss enemy is too hard
- some enemies are too easy
- Boss could be easier / quicker to defeat
- Enemies can just be avoided!

7.1.1.8. Question 10
“How did you find the puzzle elements in the game” (0 = Poor – 10 = Great)

As shown in figure 6.64 below, all individuals were happy with the puzzle elements implemented so far.

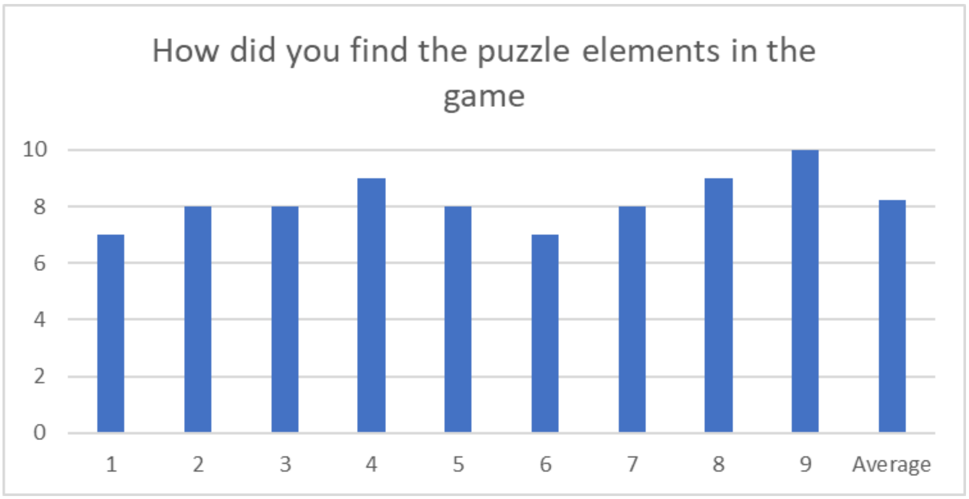


Figure 6.64 – The puzzle elements question – Bar Chart

7.1.1.9. Question 11

"How did you find the Gameplay flow with the need to collect abilities and return to previous levels to get 100% completion?" (0 = Poor – 10 = Great)

As shown in figure 6.65 below, most testers seemed to like the gameplay flow laid out in the project.

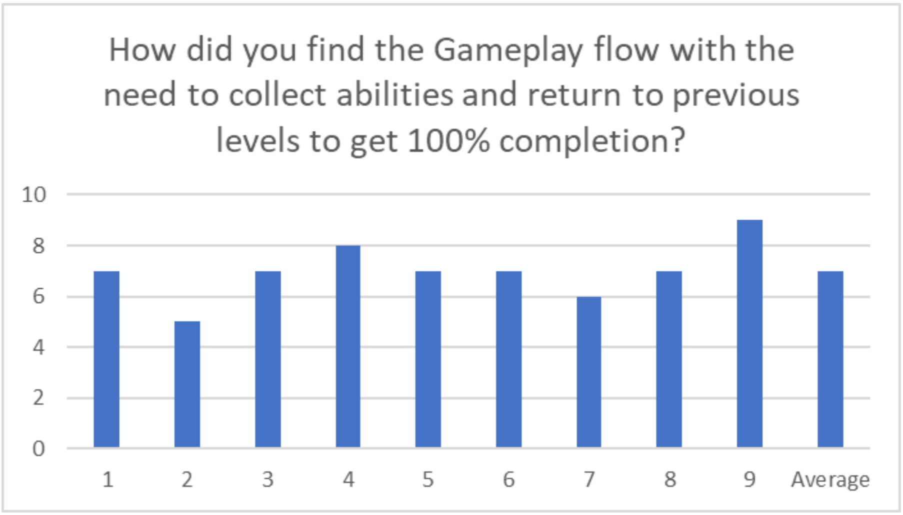


Figure 6.65 – Gameplay flow question – Bar Chart

7.1.1.10. Question 12

"How much did you like the ______ ability?" (0 = Poor – 10 = Great)

The chart in figure 6.66 below shows that the testers liked all abilities equally except the drop attack which was not found by some users.

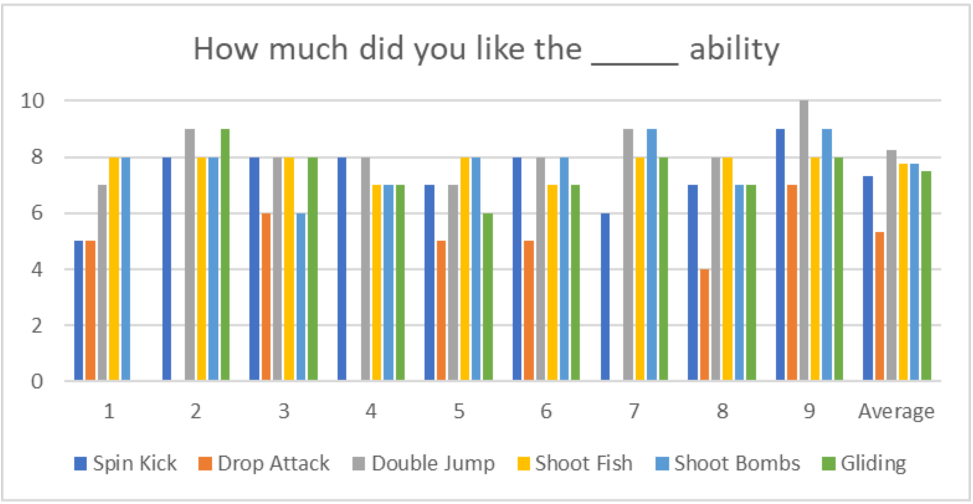


Figure 6.66 – The abilities question – Bar chart

7.1.1.11. Question 13

"Please choose which of the following ideas you would prefer to play in the game.

I had originally planned to add a further two levels and the boss's lair level. However, given time constraints I am now looking to implement only one of the following and the boss level.

A sci-fi themed level with robot enemies, laser traps, and anti-gravity sections. This would be less enclosed with a flat platform style that the player could fall off and need to be respawned.

--------------Or---------------

An ice-themed level with penguin enemies, falling icicle traps. This would be more like the Desert level and be enclosed within a terrain?"

The chart in figure 6.67 below shows that almost three-quarters of the users that answered this question voted for the sci-fi level proposal.

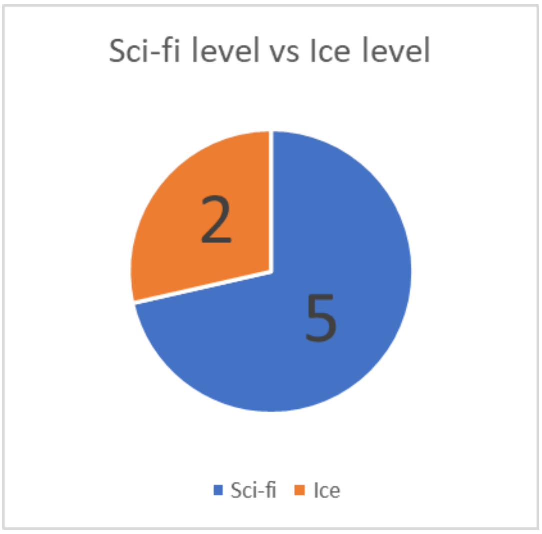


Figure 6.67 – The new level choice question – Pie chart

7.1.1.12. Question 14

"Which part of your selected option would you most like to see?"

The comments left for this question are listed below.

- Anti gravity would make an excellent idea, you could change the gravity to get to certain places when you need it or something.
- The ice level would fit the aesthetic of the game more, with it being more on the nature side like the other levels. I would like to see icicle traps or moving objects over ice to solve puzzles.
- Gravity switchers
- gravity changers
- slide mechanics
- laser traps
- robots and anti-gravity would be excellent!

7.1.1.13. Question 15

"Did you have any other feedback or suggestions for this title??"

The comments left for this question are listed below.

- If you were looking for an extra level of polish I think a custom/animated skybox would do the trick!
- enemy difficulty needs balancing
- add more scenery to enclose the levels a bit more.

7.1.2. Problems noted / bugs

The main points that came up during this playtest were,

- The camera controls need to be adjusted for sensitivity. (The keyboard and mouse is to fast and the controller is to slow). Could also add sensitivity setting in the menu.
- The levels need more scenery as they feel quite empty
- The boss is too hard to beat and other enemies are too easy
- Some people missed the drop attack completely so needs to be in a more prominent position
- The third level should be a Sci-fi level with gravity switchers and lasers.
- One tester used the signpost hinting about getting over the fence, to get over the fence. Once over the fence, they were stuck as they did not have the double jump to get back, or the shooting to move forward. This signpost needs adjusting and the collider removing.
- One user also noted that the default skybox was still being used.

7.1.3. Changes implemented

Following this playtest, the following actions were taken

- Upon testing it was found that the camera controls in the build version acted completely differently from being run in the Unity (Unity, 2021) editor. A multiplier was added to the camera controls that gets reset whenever the controls are switched. This was balanced to the build version and so is slow when run in the editor but okay in the build version.
- A lot of trees and rocks were added to both levels to make them feel less empty.
- The level one boss's health was reduced by half and its colliders increased in size to make it easier to shoot.
- Mobile enemy's speed was increased to make them slightly harder to evade.
- The drop attack ability and tutorial sign was moved next to the spin attack ability and sign. They now share the same practice chicken enemy.
- The third level was created around a spaceship theme and included requested gravity switchers and laser traps.
- The collider was removed from the signpost in the base level that was used to jump over the fence.
- The skybox was implemented as originally planned.

7.2. Playtest two – 10th April 2022

This playtest was completed after the third level was implemented and used to gauge all aspects of gameplay and to look for any final bugs. I used the same questions as before but removed the question about implementing the next level as this had now been done.

7.2.1. Questionnaire responses

For playtest two I received five responses to the questionnaire.

7.2.1.1. Question 1 – Main Menu

"How did you find the Main Menu Layout?" (0 = Poor – 10 = Great)

The chart shown in figure 6.68 below shows that most individuals again found the main menu to be satisfactory.

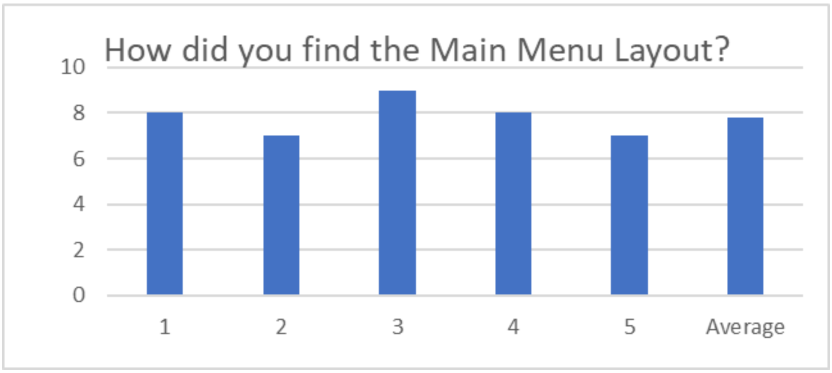


Figure 6.68 - How did you find the Main Menu Layout – Bar chart

7.2.1.2. Question 2 & 3 - Controls

"How did you find the control layout for each control scheme" (0 = Poor – 10 = Great)

The chart shown in figure 6.69 shows that all testers liked the controls better during this playtest compared to the last one.

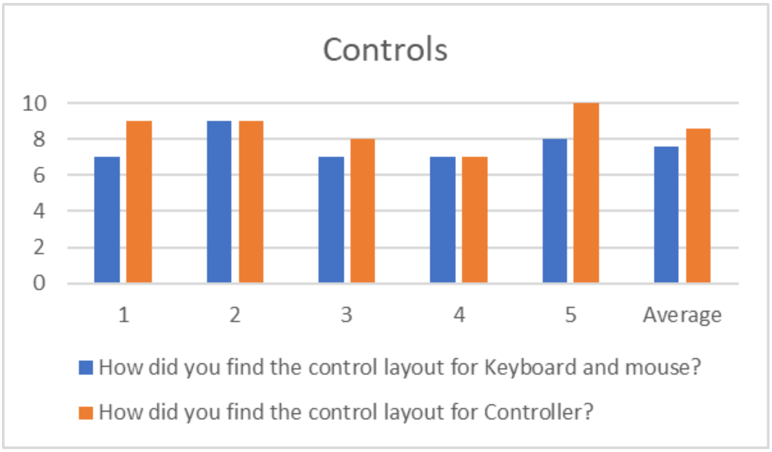


Figure 6.69 – The control scheme questions – Bar chart

The comments gathered alongside this question are listed below.

- the controller layout makes sense, would be nice to adjust controls for keyboard and mouse though as the layout was quite strange.
- controls fine
- only used keyboard and mouse, seemed ok.

7.2.1.3. Questions 4 & 5 – Main Characters
“How did you like the aesthetics/animation of the main character” (0 = Poor – 10 = Great)

The chart shown in figure 6.70 below shows that the testers were again happy with the appearance of the main characters.

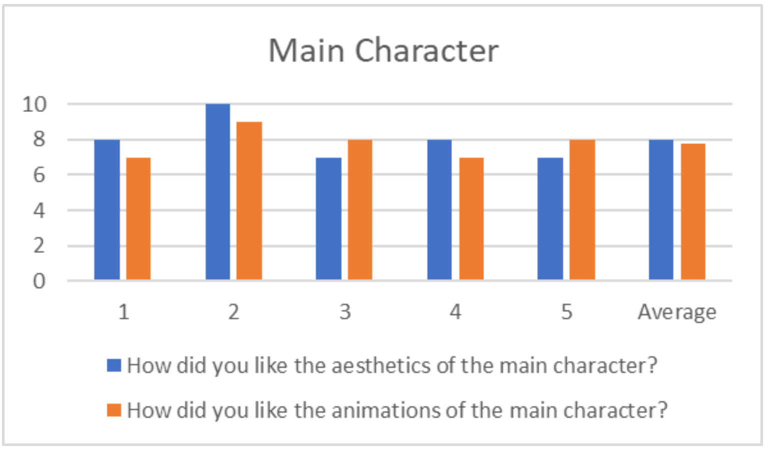


Figure 6.70 – The main character questions – Bar chart

The comments gathered alongside this question are listed below

- The dog is super cute!
- I loved the characters, reminded me of banjo kazooie but with different animals.

7.2.1.4. Question 6
“How did you like the aesthetics of the levels/enemies?” (0 = Poor – 10 = Great)

The chart in figure 6.71 below shows that feedback regarding the level and enemy aesthetics was a lot more positive this time around.

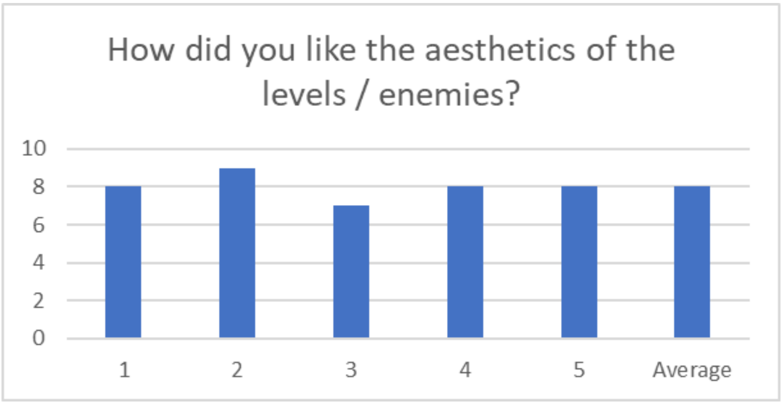


Figure 6.71 – The level/enemy aesthetics question – Bar chart

The comments gathered alongside this are listed below

- The chickens are great.
- the space level was my favourite.

7.2.1.5. Question 7

"How did you find the proposed story explained at the start of the demo?" (0 = Poor – 10 = Great)

The chart in figure 6.72 below shows that most individuals were again happy with the story proposition for the project.

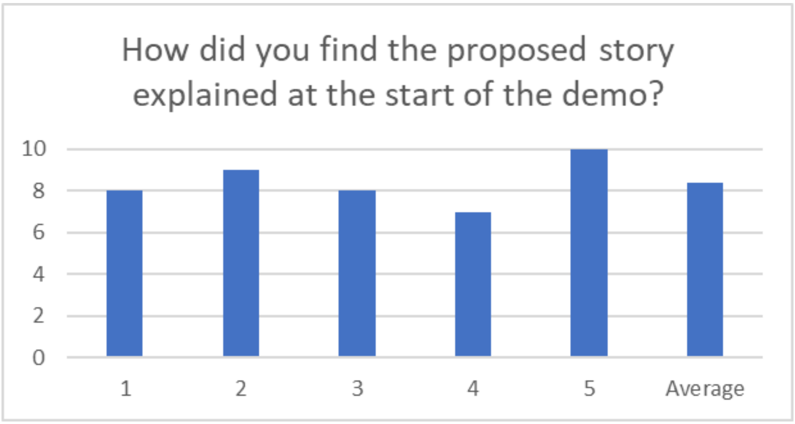


Figure 6.72 – The story question – Bar chart

7.2.1.6. Question 8

"How effective did you find the signpost method of explaining core mechanics?"
(0 = Not effective – 10 = Very Effective)

The chart in figure 6.73 below shows that most individuals found the signposts to work well to explain the game.

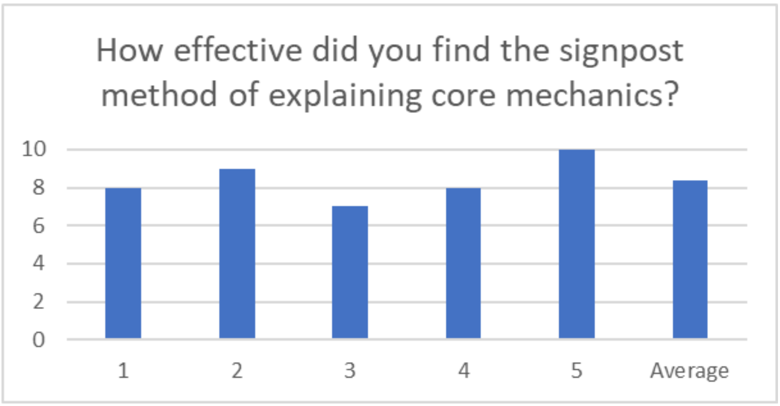


Figure 6.73 – The signpost/tutorial question – Bar chart

7.2.1.7. Question 9

"How did you find the enemy difficulty in the game?" (0 = Too Easy – 10 = Too Hard)

As shown in figure 6.74 below, this question received a more balanced response, with the average value showing that the enemy difficulty was balanced more evenly this time.

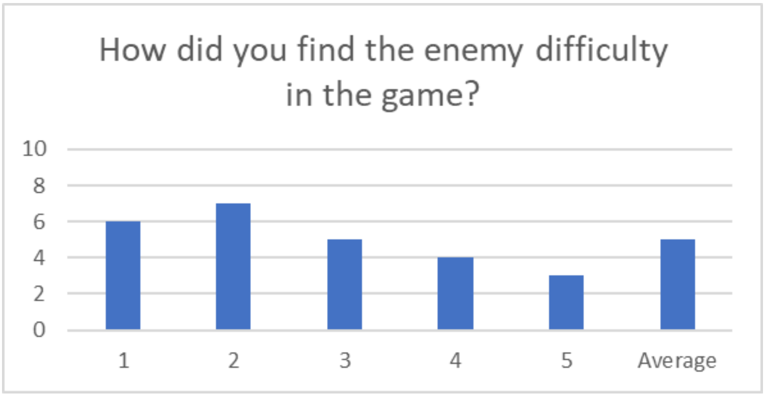


Figure 6.74 – The enemy difficulty question – Bar chart

The comments gathered alongside this question are listed below.

- Final boss is quite hard with the floor opening, took ages to die!
- Most enemies are quite easy to avoid.

7.2.1.8. Question 10
"How did you find the puzzle elements in the game" (0 = Poor – 10 = Great)

As shown in figure 6.75 below, all individuals were happy again with the puzzle elements implemented so far.

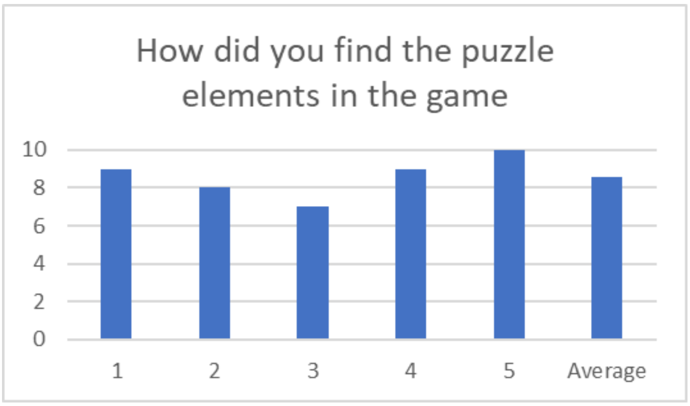


Figure 6.75 – The puzzle elements question – Bar Chart

7.2.1.9. Question 11
"How did you find the Gameplay flow with the need to collect abilities and return to previous levels to get 100% completion?" (0 = Poor – 10 = Great)

As shown in figure 6.76 below, most testers seemed to like the gameplay flow laid out in the project.

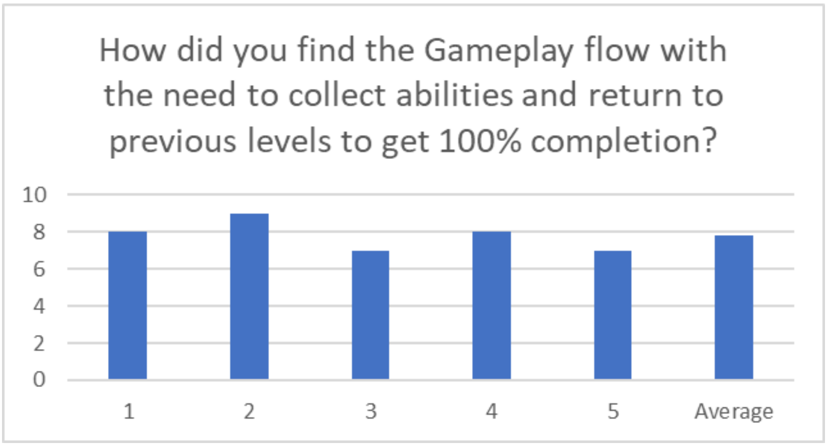


Figure 6.76 – Gameplay flow question – Bar Chart

7.2.1.10. Question 12

"How much did you like the ______ ability?" (0 = Poor – 10 = Great)

The chart in figure 6.77 below shows that the testers liked all abilities with all users finding all abilities this time around.

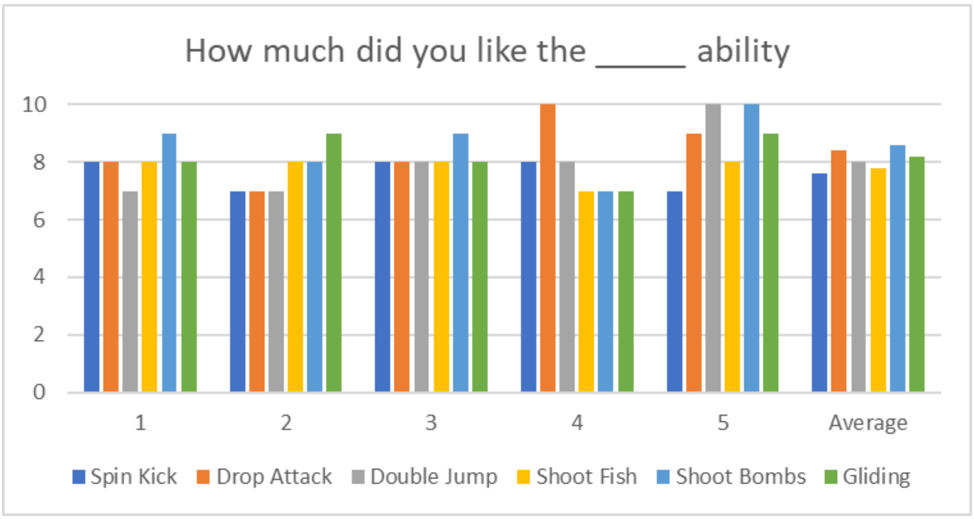


Figure 6.77 – The abilities question – Bar chart

7.2.1.11. Question 13

"Did you have any other feedback or suggestions for this title??"

The comments left for this question are listed below.

- Really enjoyed this game, well done!
- Great fun, would like to continue if you carry on with it.

7.2.2. Problems noted / bugs

Although this playtest received more positive responses, the main points that were highlighted are listed below.

- One user noted that it would be good to be able to re-define the controls in-game.
- The final boss was mentioned as being too difficult.
- In level one some missing colliders were allowing the player to get out of bounds once they had the steep slope ability
- There were a couple of glitches with the gravity switchers that teleported the player outside of the level.

7.2.3. Changes implemented

Following this playtest, the following actions were taken

- Redefining controls was investigated, but it will be quite a lengthy process and so due to time constraints, work on the project will focus on bug fixing and implementing the sound for the remainder of development time.
- The final boss's health was reduced by half to make it easier and quicker to defeat.
- Colliders were added/adjusted in level one.
- The gravity switchers were adjusted by adding a lerp function when rotating the player and adjusting their colliders to try and help with the bug found. As a fail-safe kill floors were added surrounding the level so if the player is teleported outside of the level, they can simply jump off to respawn.

8.0. Future work

Moving forward with the project, here is a list of features that are planned to be implemented.

- Add cut scenes to get the main story across in a clearer way.
- Create an ice/glacier level with icicle traps, and a sliding mechanism to make the controls harder. This would have a large penguin type boss.
- Create the final boss level that would be a large castle level with the final boss being an evil wizard type of character.
- Add the sensitivity setting to the menu system.
- Add the ability to reassign the controls in-game.
- Add a screen resolution to the settings menu.

9.0. References

Activision, 2020. *Spyro the Dragon.* [Online]
Available at: https://www.spyrothedragon.com/
[Accessed 15 October 2021].

Activision, 2022. *Tony Hawks Pro Skater 1 + 2.* [Online]
Available at: https://www.tonyhawkthegame.com/content/atvi/tony-hawk/alcatraz/web/en/home.html
[Accessed 18 April 2022].

Adobe, 2021. *Mixamo.* [Online]
Available at: https://www.mixamo.com/
[Accessed 25 October 2021].

Adobe, 2021. *Photoshop.* [Online]
Available at: https://www.adobe.com/products/photoshop.html
[Accessed 25 October 2021].

alexo400, 2020. *Freesound - SCI_FI_DOOR.wav.* [Online]
Available at: https://freesound.org/people/alexo400/sounds/543404/
[Accessed 23 April 2022].

AlienXXX, 2013. *Freesound - Impact_007a.wav.* [Online]
Available at: https://freesound.org/people/AlienXXX/sounds/196221/
[Accessed 23 April 2022].

Argonaut, 2000. *Croc: Legend of the Gobbos.* [Online]
Available at: https://en.wikipedia.org/wiki/Croc:_Legend_of_the_Gobbos
[Accessed 15 October 2021].

AustinXYZ, 2016. *Freesound - Chihuahua Puppy Whine.* [Online]
Available at: https://freesound.org/people/AustinXYZ/sounds/350593/
[Accessed 23 April 2022].

BCS, 2021. *BCS - Code of Conduct.* [Online]
Available at: https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf
[Accessed 25 October 2021].

Blender, 2021. *Blender.* [Online]
Available at: https://www.blender.org/
[Accessed 25 October 2021].

blukotek, 2014. *Freesound - bone shell.* [Online]
Available at: https://freesound.org/people/blukotek/sounds/249319/
[Accessed 23 April 2022].

Boxophobic, 2018. *Unity Asset Store - Polyverse Skies.* [Online]
Available at: https://assetstore.unity.com/packages/vfx/shaders/polyverse-skies-low-poly-skybox-shaders-104017
[Accessed 21 April 2022].

Comp-3, 2021. *YouTube - How to Add a Field of View for Your Enemies [Unity Tutorial].* [Online]
Available at: How to Add a Field of View for Your Enemies [Unity Tutorial]
[Accessed 22 April 2022].

DaFont, 2022. *DaFont.com.* [Online]
Available at: https://www.dafont.com
[Accessed 23 April 2022].

Daleonfire, 2021. *Freesound - Sci-Fi Alarm.* [Online]
Available at: https://freesound.org/people/Daleonfire/sounds/559476/
[Accessed 23 April 2022].

davilca, 2012. *Freesound - sfx-leaves-1.wav.* [Online]
Available at: https://freesound.org/people/davilca/sounds/159496/
[Accessed 23 April 2022].

DDmyzik, 2021. *Freesound - Bluesy Hip Hop Loop.* [Online]
Available at: https://freesound.org/people/DDmyzik/sounds/555801/#
[Accessed 23 April 2022].

Epic, 2021. *Unreal Engine.* [Online]
Available at: https://www.unrealengine.com/
[Accessed 25 October 2021].

Exoside, 2020. *Quad Remesher.* [Online]
Available at: https://exoside.com/quadremesher/
[Accessed 17 November 2021].

Flood, D., 2020. *dafont.com - Cartoon Fun.* [Online]
Available at: https://www.dafont.com/cartoon-fun.font
[Accessed 23 April 2022].

Freesound, 2022. *Freesound.org.* [Online]
Available at: https://freesound.org/
[Accessed 23 April 2022].

InspectorJ, 2017. *Freesound - Chewing, Carrot, A.wav.* [Online]
Available at: https://freesound.org/people/InspectorJ/sounds/412068/
[Accessed 23 April 2022].

jeremysykes, 2016. *Freesound - Laser05.wav.* [Online]
Available at: https://freesound.org/people/jeremysykes/sounds/344512/
[Accessed 23 April 2022].

jmayoff, 2014. *Freesound - Laser.* [Online]
Available at: https://freesound.org/people/jmayoff/sounds/253446/
[Accessed 23 April 2022].

juskiddink, 2012. *Freesound - Boing.wav.* [Online]
Available at: https://freesound.org/people/juskiddink/sounds/140867/
[Accessed 23 April 2022].

JustInvoke, 2018. *Freesound - Wet Splat.* [Online]
Available at: https://freesound.org/people/JustInvoke/sounds/446115/
[Accessed 23 April 2022].

kindingerlek, 2016. *Sketchfab - Low Poly Scorpion.* [Online]
Available at: https://sketchfab.com/3d-models/low-poly-scorpion-5058385abe644476bc964b5eb2dab0b2
[Accessed 22 April 2022].

marchon11, 2019. *Freesound - throw.wav.* [Online]
Available at: https://freesound.org/people/marchon11/sounds/493224/
[Accessed 23 April 2022].

Meshint, 2019. *Unity Asset Store - Meshtint Free Chicken Mega Toon Series.* [Online]
Available at: https://assetstore.unity.com/packages/3d/characters/animals/meshtint-free-chicken-

mega-toon-series-151842
[Accessed 22 April 2022].

newagesoup, 2017. *Freesound - wide forward woosh1.wav.* [Online]
Available at: https://freesound.org/people/newagesoup/sounds/377834/
[Accessed 23 April 2022].

Ott, J., 2021. *Unity Asset Store - Character Movement Fundamentals.* [Online]
Available at: https://assetstore.unity.com/packages/tools/physics/character-movement-fundamentals-144966
[Accessed 21 April 2022].

OwlStorm, 2017. *Freesound - Retro video game sfx - R2D2.* [Online]
Available at: https://freesound.org/people/OwlStorm/sounds/404759/
[Accessed 23 April 2022].

Pinwheel, 2021. *Unity Asset Store - Polaris 2021 - Low Poly & Mesh Terrain Editor.* [Online]
Available at: https://assetstore.unity.com/packages/tools/terrain/polaris-2021-low-poly-mesh-terrain-editor-196648
[Accessed 21 April 2022].

Playful, 2019. *New Super Lucky's Tale.* [Online]
Available at: https://playfulstudios.com/new-super-luckys-tale/
[Accessed 15 October 2021].

Playtonic, 2017. *Yooka-Laylee.* [Online]
Available at: https://www.playtonicgames.com/games/yooka-laylee/
[Accessed 15 October 2021].

primeval_polypod, 2012. *Freesound - airlock_extended.wav.* [Online]
Available at: https://freesound.org/people/primeval_polypod/sounds/156508/
[Accessed 23 April 2022].

qubodup, 2008. *Freesound - Whoosh.* [Online]
Available at: https://freesound.org/people/qubodup/sounds/60013/
[Accessed 23 April 2022].

Rare, 1998. *Banjo-Kazooie.* [Online]
Available at: https://en.wikipedia.org/wiki/Banjo-Kazooie
[Accessed 15 October 2021].

Rare, 2001. *Conker's Bad Fur Day.* [Online]
Available at: https://en.wikipedia.org/wiki/Conker%27s_Bad_Fur_Day
[Accessed 15 October 2021].

Rocksteady, 2015. *Rocksteady Games - Batman Arkham knight.* [Online]
Available at: https://rocksteadyltd.com/#arkham-knight
[Accessed 18 April 2022].

Rudmer_Rotteveel, 2015. *Freesound - Chicken Single Alarm Call.* [Online]
Available at: https://freesound.org/people/Rudmer_Rotteveel/sounds/316920/
[Accessed 23 April 2022].

Scrampunk, 2016. *Freesound - OKAY!.* [Online]
Available at: https://freesound.org/people/Scrampunk/sounds/345299/
[Accessed 23 April 2022].

Sega, 2021. *Two Point Hospital.* [Online]
Available at: https://www.twopointhospital.com/en
[Accessed 18 April 2022].

Seidhepriest, 2012. *Freesound - Wind Chimes Loop.* [Online]
Available at: https://freesound.org/people/Seidhepriest/sounds/171957/
[Accessed 23 April 2022].

SieuAmThanh, 2022. *Freesound - Nổ 4.* [Online]
Available at: https://freesound.org/people/SieuAmThanh/sounds/620229/
[Accessed 23 April 2022].

Sirkoto51, 2016. *Freesound - Atmospheric Ambiance Loop #2.* [Online]
Available at: https://freesound.org/people/Sirkoto51/sounds/370175/
[Accessed 23 April 2022].

Sketchfab, 2022. *Sketchfab.* [Online]
Available at: https://sketchfab.com/feed
[Accessed 22 April 2022].

Style-7, 2008. *DaFont.com - Digital-7.* [Online]
Available at: https://www.dafont.com/digital-7.font
[Accessed 23 April 2022].

Sunsai, 2018. *Freesound - Desert background music.* [Online]
Available at: https://freesound.org/people/Sunsai/sounds/415806/
[Accessed 23 April 2022].

Team17, 2022. *Overcooked.* [Online]
Available at: https://www.team17.com/games/overcooked/
[Accessed 18 April 2022].

tim.kahn, 2011. *Freesound - transport 2.wav.* [Online]
Available at: https://freesound.org/people/tim.kahn/sounds/128587/
[Accessed 23 April 2022].

tonnonic, 2016. *Freesound - Dartboard 5.aif.* [Online]
Available at: https://freesound.org/people/tonnonic/sounds/332201/
[Accessed 23 April 2022].

Unity, 2021. *Unity.* [Online]
Available at: https://unity.com/
[Accessed 25 October 2021].

Unity, 2022. *Unity Asset Store.* [Online]
Available at: https://assetstore.unity.com/
[Accessed 21 April 2022].

xsgianni, 2017. *Freesound - Marios's way.mp3.* [Online]
Available at: https://freesound.org/people/xsgianni/sounds/388079/
[Accessed 23 April 2022].

xUMR, 2019. *Freesound - wind.loop.ogg.* [Online]
Available at: https://freesound.org/people/xUMR/sounds/487433/
[Accessed 23 April 2022].

zivs, 2018. *Freesound - Ammo pickup #2.* [Online]
Available at: https://freesound.org/people/zivs/sounds/433771/
[Accessed 23 April 2022].

10.0.Appendix

Appendix A User testing compliance form

User Testing Compliance Form for UG and PGT Projects[*]

School of Engineering and Informatics

University of Sussex

This form should be used in conjunction with the document entitled "Research Ethics Guidance for UG and PGT Projects".

Prior to conducting your project, you and your supervisor will have discussed the ethical implications of your research. If it was determined that your proposed project would comply with **all** the points in this form, then both you and your supervisor should complete and sign the form on page 3 and submit the signed copy with your final project report/dissertation.

If this is not the case, you should refer to the "Research Ethics Guidance for UG and PGT Projects" document for further guidance.

---

1. Participants were not exposed to any risks greater than those encountered in their normal working life.

Investigators have a responsibility to protect participants from physical, mental, and emotional harm during the investigation. The risk of harm must be no greater than in ordinary life. Areas of potential risk that require ethical approval include, but are not limited to, investigations that require participant mobility (e.g. walking, running, use of public transport), unusual or repetitive activity or movement, physical hazards or discomfort, emotional distress, use of sensory deprivation (e.g. ear plugs or blindfolds), sensitive topics (e.g. sexual activity, drug use, political behaviour, ethnicity) or those which might induce discomfort, stress or anxiety (e.g. violent video games), bright or flashing lights, loud or disorienting noises, smell, taste, vibration, or force feedback.

2. The study materials were paper-based, or comprised software running on standard hardware.

Participants should not be exposed to any risks associated with the use of non-standard equipment: anything other than pen-and-paper, standard PCs, mobile phones, and tablet computers is considered non-standard.

3. All participants explicitly stated that they agreed to take part, and that their data could be used in the project.

[*]This checklist was originally developed by Professor Steven Brewster at the University of Glasgow and modified by Dr Judith Good for use at the University of Sussex with his permission.

Participants cannot take part in the study without their knowledge or consent (i.e., no covert observation). Covert observation, deception or withholding information are deemed to be high risk and require ethical approval through the relevant C-REC.

If the results of the evaluation are likely to be used beyond the term of the project (for example, the software is to be deployed, the data is to be published or there are future secondary uses of the data), then it will be necessary to obtain signed consent from each participant. Otherwise, verbal consent is sufficient, and should be explicitly requested in the introductory script (see Appendix 1).

4. No incentives were offered to the participants.

The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle. People volunteering to participate in research may be compensated financially e.g., for reasonable travel expenses. Payments made to individuals must not be so large as to induce individuals to risk harm beyond that which they would usually undertake.

5. No information about the evaluation or materials was intentionally withheld from the participants.

Withholding information from participants or misleading them is unacceptable without justifiable reasons for doing so. Any projects requiring deception (for example, only telling participants of the true purpose of the study afterwards so as not to influence their behaviour) are deemed high risk and require approval from the relevant C-REC.

6. No participant was under the age of eighteen.

Any studies involving children or young people are deemed to be high risk and require ethical approval through the relevant C-REC.

7. No participant had a disability or impairment that may have limited their understanding or communication or capacity to consent.

Projects involving participants with disabilities are deemed to be high risk and require ethical approval from the relevant C-REC.

8. Neither I nor my supervisor are in a position of authority or influence over any of the participants.

A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any study.

9. All participants were informed that they could withdraw at any time.

All participants have the right to withdraw at any time during the investigation. They should be told this in the introductory script (see Appendix 1).

10. All participants have been informed of my contact details, and the contact details of my supervisor.

All participants must be able to contact the investigator and/or the supervisor after the investigation. They should be given contact details for both student and supervisor as part of the debriefing.

11. The evaluation was described in detail with all the participants at the beginning of the session, and participants were fully debriefed at the end of the session. All participants were given the opportunity to ask questions at both the beginning and end of the session.

Participants must be provided with sufficient information prior to starting the session, and in the debriefing, to enable them to understand the nature of the investigation.

12. All the data collected from the participants is stored securely, and in an anonymous form.

All participant data (hard-copy and soft-copy) should be stored securely (i.e., locked filing cabinets for hard copy, password protected computer for electronic data), and in an anonymised form.

Project title:	Bow-Bow and Bum-Bum (Working Title)
Student's Name:	Daniel Newsom
Student's Registration Number:	21902155
Student's Signature:	[illegible]
Date:	08/10/2021
Supervisor's Name:	Paul Newbury
Supervisor's Signature:	Paul Newbury
Date	12.10.2021

User Testing Compliance Form for UG and PGT Projects*

Appendix 1: Introduction and Debriefing Scripts

If you intend to obtain verbal consent from your participants, you need to create an introduction script, to read out at the start of the study, and a debriefing script, to read out at the end of the study.

You should get your supervisor's approval for your introduction and debriefing scripts before commencing your study. **NB: When submitting the signed User Testing Compliance form, you do not need to include this Appendix.**

Introduction Script

The introduction script must:

- state the general aim of the study
- explain why you need the involvement of other people
- describe what will happen in the study
- describe what data will be collected
- reassure the participant that any data collected will be stored securely, and in an anonymous format
- explain what interaction the participant may have with you during the study
- reassure the participant that this is not a test of ability
- state that the participant may withdraw at any time
- seek explicit consent
- allow the participant to ask questions

An example introductory script (italics indicate required information, some of which will be specific to your project):

Web Interface Investigation

Final Year Project, 2020-21

Jane Student

The aim of this study is to investigate the suitability of a new website [*state the general aim of the experiment*]. We cannot tell how good this website is unless we ask those people who are likely to be using it, which is why we need to run studies like these [*explain why you need the involvement of other people*]. I will give you some time to browse the website, before asking you to answer some questions [*describe what will happen in the study*].

I will be observing you while you perform the tasks [*describe what data will be collected*]. If you have any questions, please ask me, and please let me know when you are finished [*explain what interaction the participant may have with you during the study*].

I will ask you some questions at the end of the study [*describe what will happen in the study*]. Please remember that it is the system, not you, that is being evaluated [*reassure the participant that this is not a test of ability*].

You are welcome to withdraw from the study at any time [*state that the participant may withdraw at any time*]. Please be assured that any data collected will be stored securely and in an anonymous form [*describe how the data will be anonymised and stored*].

Do you agree to take part in this evaluation? [*seek explicit consent*]. Do you have any questions before we start? [*allow the participant to ask questions*].

Debriefing Script

The debriefing script that is used at the end of the study must:

- restate the main aim of the experiment
- if applicable, explain any other, related aims of the experiment, and any data collected
- allow the participant to make comments or ask questions
- give contact details for the student and supervisor
- thank the participant

An example debriefing script (italics indicate required information, some of which will be specific to your project):

Web Interface Investigation

Final Year Project, 2020-21

Jane Student

The main aim of the experiment was to investigate the suitability of this website [*restate the main aim of the experiment*]. I was looking to see whether you made use of the site map, and whether any of the links on the website seemed to be confusing. I wanted to know how easy the pages were to navigate [*if applicable, explain any other, related aims of the experiment, and any data collected*].

Do you have any comments or questions about the experiment? [*allow the*

participant to make comments or ask questions]. Here are my contact details, and those of my supervisor, and please let us know if you have any further questions about this study [*give contact details of student and supervisor to participant*]. Thank you for your help [*thank the participant*].

Appendix B Initial project proposal

Candidate: Daniel Newsom – 215851

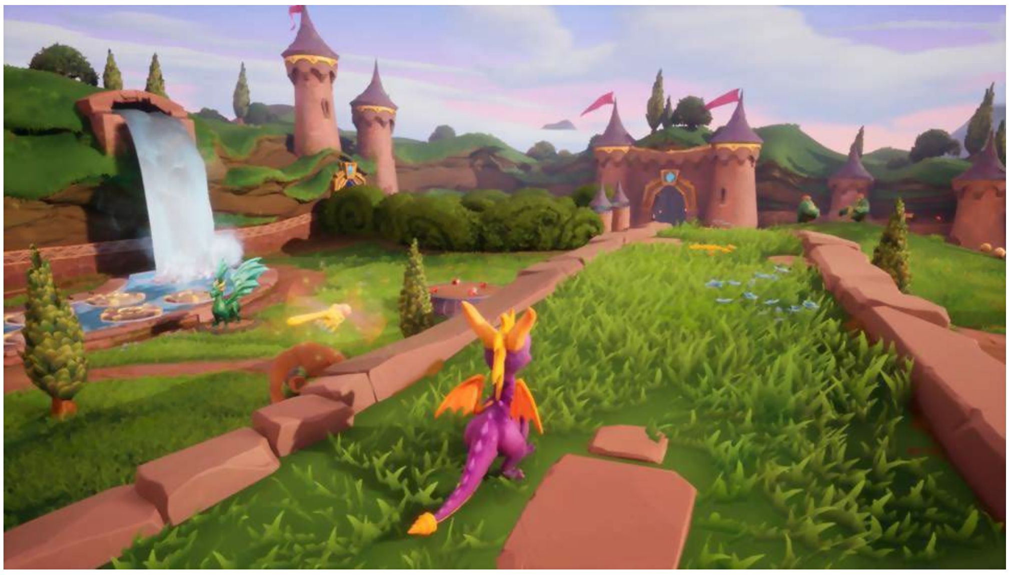
Supervisor: Paul Newbury

Working Title: Bow-Bow and Bum-Bum

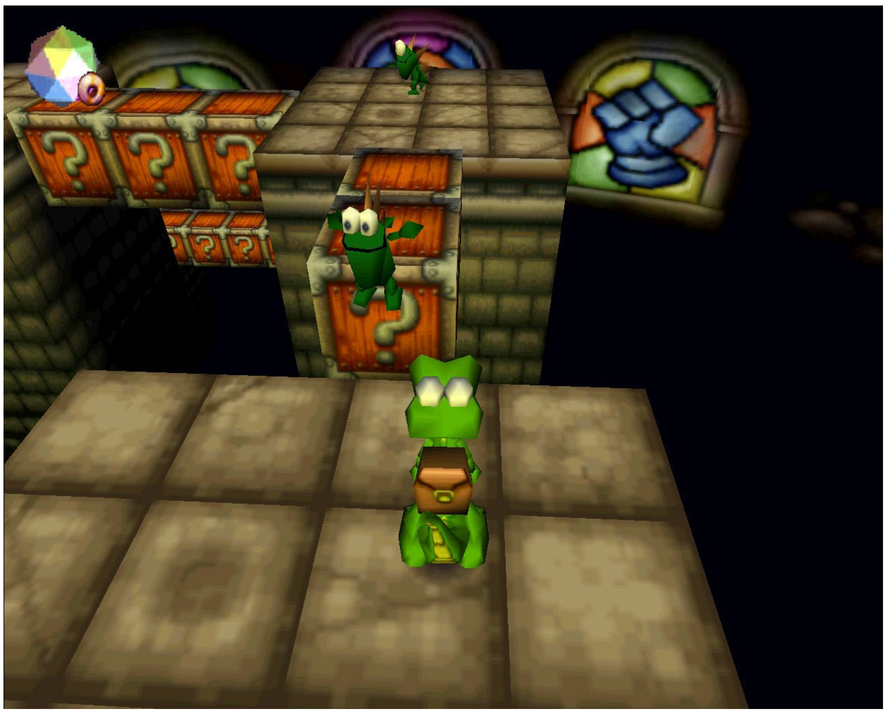
Aims and objectives

I am intending to create a 3D platformer game based on our dog Bow and our cat Bum-Bum (blame my children). I am planning to make a game in the same general style as the following examples that I enjoyed playing when I was younger.

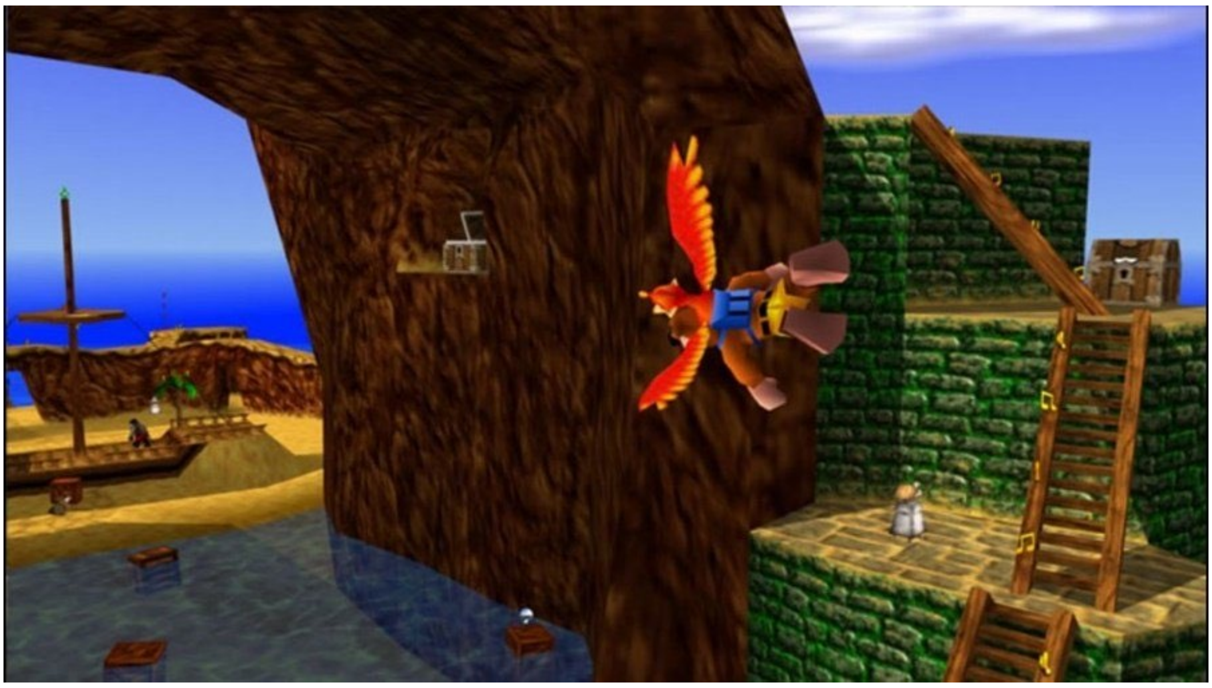
Spyro the Dragon (Activision, 2020)



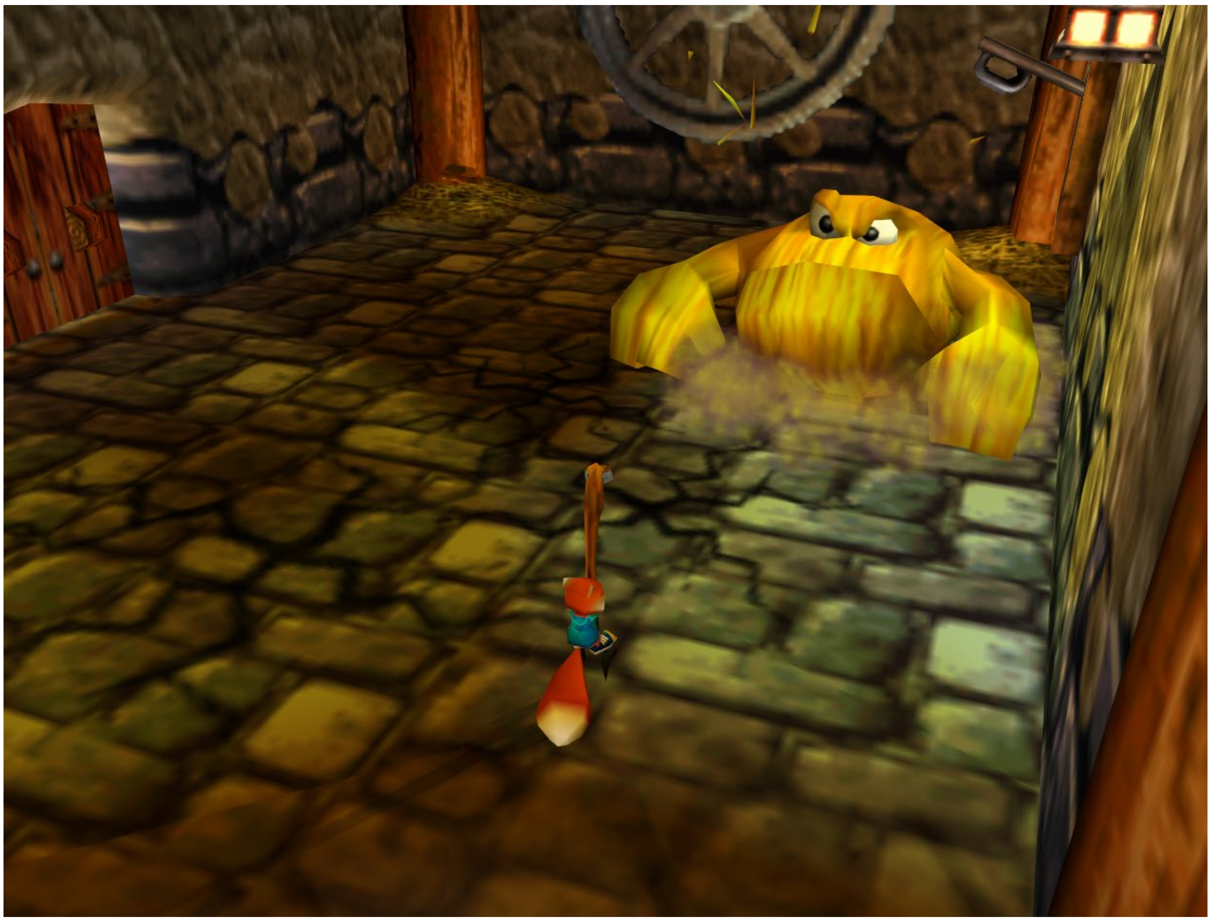
Croc legend of the Gobbo’s (Argonaut, 2000)



Banjo-Kazooie (Rare, 1998)



Conkers Bad Fur Day (Rare, 2001)



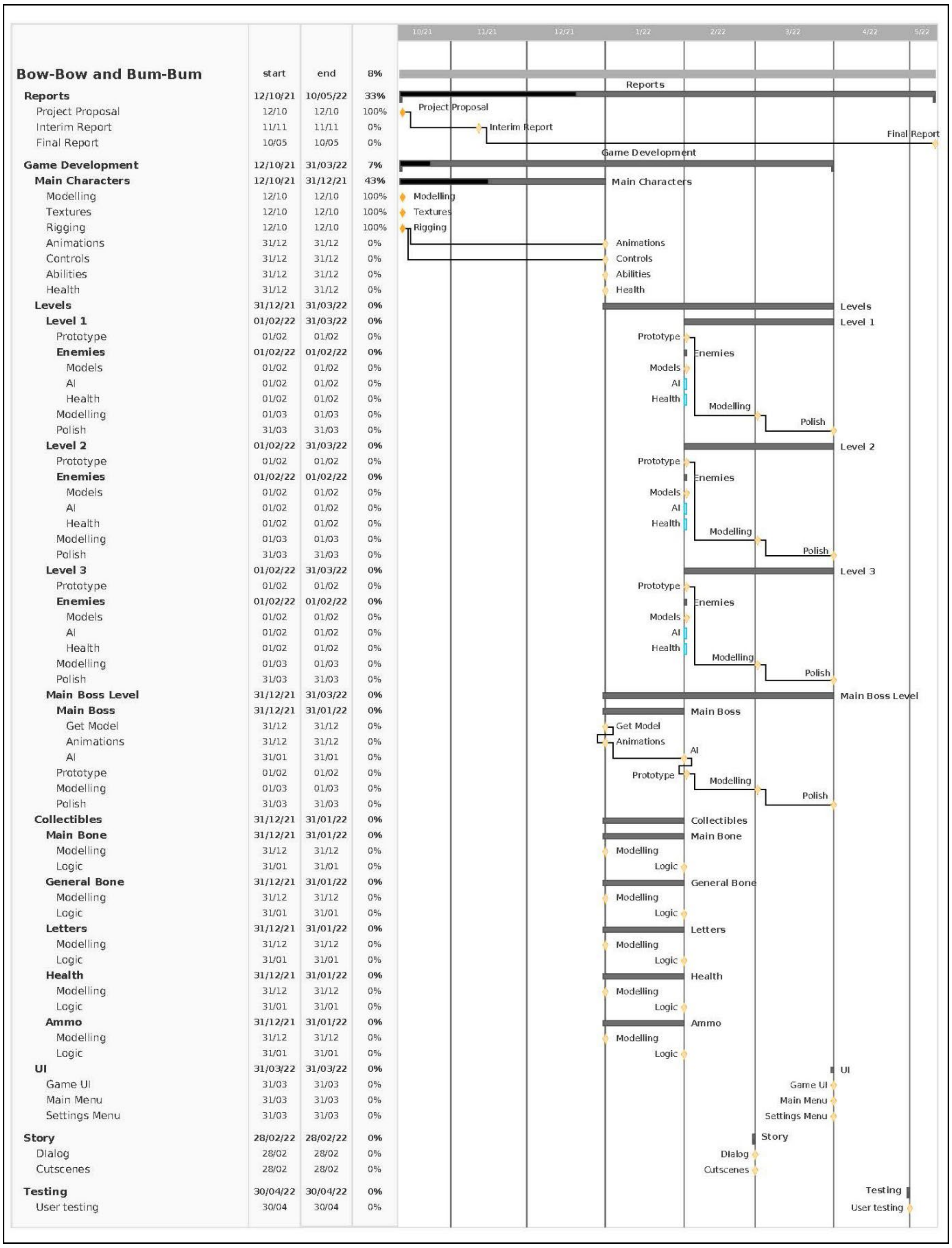
This means that the general feel will be quite a cartoon style game with a main 'Portal' level leading to various themed worlds. Each world will then contain various collectables that will be required to unlock further portals/worlds. The main premise will be to collect a given number of key items to access a boss fight to end the game.

Additionally, I am planning to implement a skill acquisition/shop mechanism, which will allow the player to access previously unreachable areas. (Double jump, gliding, shooting switches etc…) to add re-playability to the levels and make the player return to complete the game 100%.

Relevance

I am creating this project to demonstrate all the skills learnt studying for the Games and Multimedia Environments BSc (Hons) Degree at the University of Sussex, including writing clean and efficient code, user experience, visual effects, 3d modelling and animations.

Appendix C Project Gantt Chart



Bow-Bow and Bum-Bum	start	end	8%
Reports	12/10/21	10/05/22	33%
Project Proposal	12/10	12/10	100%
Interim Report	11/11	11/11	0%
Final Report	10/05	10/05	0%
Game Development	12/10/21	31/03/22	7%
Main Characters	12/10/21	31/12/21	43%
Modelling	12/10	12/10	100%
Textures	12/10	12/10	100%
Rigging	12/10	12/10	100%
Animations	31/12	31/12	0%
Controls	31/12	31/12	0%
Abilities	31/12	31/12	0%
Health	31/12	31/12	0%
Levels	31/12/21	31/03/22	0%
Level 1	01/02/22	31/03/22	0%
Prototype	01/02	01/02	0%
Enemies	01/02/22	01/02/22	0%
Models	01/02	01/02	0%
AI	01/02	01/02	0%
Health	01/02	01/02	0%
Modelling	01/03	01/03	0%
Polish	31/03	31/03	0%
Level 2	01/02/22	31/03/22	0%
Prototype	01/02	01/02	0%
Enemies	01/02/22	01/02/22	0%
Models	01/02	01/02	0%
AI	01/02	01/02	0%
Health	01/02	01/02	0%
Modelling	01/03	01/03	0%
Polish	31/03	31/03	0%
Level 3	01/02/22	31/03/22	0%
Prototype	01/02	01/02	0%
Enemies	01/02/22	01/02/22	0%
Models	01/02	01/02	0%
AI	01/02	01/02	0%
Health	01/02	01/02	0%
Modelling	01/03	01/03	0%
Polish	31/03	31/03	0%
Main Boss Level	31/12/21	31/03/22	0%
Main Boss	31/12/21	31/01/22	0%
Get Model	31/12	31/12	0%
Animations	31/12	31/12	0%
AI	31/01	31/01	0%
Prototype	01/02	01/02	0%
Modelling	01/03	01/03	0%
Polish	31/03	31/03	0%
Collectibles	31/12/21	31/01/22	0%
Main Bone	31/12/21	31/01/22	0%
Modelling	31/12	31/12	0%
Logic	31/01	31/01	0%
General Bone	31/12/21	31/01/22	0%
Modelling	31/12	31/12	0%
Logic	31/01	31/01	0%
Letters	31/12/21	31/01/22	0%
Modelling	31/12	31/12	0%
Logic	31/01	31/01	0%
Health	31/12/21	31/01/22	0%
Modelling	31/12	31/12	0%
Logic	31/01	31/01	0%
Ammo	31/12/21	31/01/22	0%
Modelling	31/12	31/12	0%
Logic	31/01	31/01	0%
UI	31/03/22	31/03/22	0%
Game UI	31/03	31/03	0%
Main Menu	31/03	31/03	0%
Settings Menu	31/03	31/03	0%
Story	28/02/22	28/02/22	0%
Dialog	28/02	28/02	0%
Cutscenes	28/02	28/02	0%
Testing	30/04/22	30/04/22	0%
User testing	30/04	30/04	0%