Solent University

School of Media, Arts and Technology

BA (Hons) Computer Games (Design)
2021

Final Project Report – Imperium Games

Supervisor	:	James Terkeurst
Date of submission	:	05/2021

1 Acknowledgements

James Terkeurst and Kostas Dokos who have played a big part in preparing and guiding me towards this project, I would also like to acknowledge constructive feedback from Joshua Hardy and Joshua Keene who have been vital during both the testing and production of this project.

2 Abstract

This report acts as a post-mortem of my final major project, discussing its production as well as evaluating whether it successfully achieved the goals set out in its plan and reflecting on both strengths and weakness. Overall, the project is found to be a success with All aims being met, however, some areas warrant further development or polishing.

3 Contents

4 Definition of Project

This project requires the completion of a portfolio piece, reflecting and demonstrating the candidate’s interest and personal skills towards to goal of displaying an ability to plan, execute and deliver a product within a specific timeframe. After taking account of my interests and skillset, I decided that the best way for me to showcase such skills was through the creation of a game that displays a range of systemic and mechanical gameplay.

After researching games that meet this set criteria and taking personal preference into account, I decided to create a first-person survival game, that includes mechanics commonly associated with the genre: Crafting, Looting, Inventory systems, Combat, Puzzling, Artificial intelligence, and Level design. The objective of creating a piece such as this will be to provide a cornerstone to my portfolio, showing my ability to design gameplay mechanics and systems.

The aims of this project were:

- Produce a range of systemic mechanics demonstrating technical and mechanical design skills.
- Develop a sandbox environment to support gameplay.
- Complete a full gameplay experience from start to finish with narrative.

Research and inspiration for this project were taken firstly from Rust (FacePunch, 2021), a game that involves similar mechanics and systems in a multiplayer environment. Secondly, for references on certain aspects of gameplay UI, I also used Ark Survival Evolved (Ark, 2021) and Dayz (Bohemia Interactive, 2021).

Finally, issues and problems encountered during the project were most often dealt with through research on forums such as Unity Answers (2021), Stack Overflow (2021), and YouTube (2021).

5 Project Plan

Solent University

The Nexus Games

Project Proposal

“Project Re-named to Imperium Games”

Nathan Tubb
2-5-2021

Table of Contents

1 Project Overview

The Nexus Games is a first-person survival game featuring crafting, combat, and puzzle exploration, dropping the player into a dystopian world as the contestant of new entertainment. These games place willing contestants into a deadly arena where they must work from nothing to survive, using only what they can scavenge and craft to defend and feed themselves. For more details on the theme of this project see Appendix A.

2 Core Project Aims

This project aims to demonstrate technical and mechanical design with the aims of producing a product that utilises them in a short narrative level. The goal is to have a fully playable experience with the following mechanics:

2.1 Combat

Combat will consist of the following animations for the following items:

Weapon	Attack (same each tier)	Aim (Shoot/ Throw)	Shoot	Block	Throw	Reload	Draw	Total
Fists	Yes	No	No	No	No	No	Yes	2
Stone	Yes	Yes	No	No	Yes	No	Yes	3
Spears (tier 1-3)	Yes	Yes	No	Yes	Yes	No	Yes	5
Knife (tier 1-3)	Yes	Yes	No	No	Yes	No	Yes	4
Crafting Axe	Yes (same as harvest)	No	No	No	No	No	Yes	2
Crafting Pickaxe	Yes (same as harvest)	No	No	No	No	No	Yes	2
Bow (after melee combat)	No	Yes	Yes	No	No	Yes	Yes	4
Pistol (stretch goal)	No	Yes	Yes	No	No	Yes	Yes	4
Total	5	5	2	1	3	2	8	26

Basic attack animations are the initial goal, with blocking and throwing being based on the success of the initial implementation. Combat will consist of the following parameters for weapons:

Weapon	Melee Damage tier 1-3	Ranged Damage	Draw Speed From Inventory	Rate of use (fire)	Reload (New arrow/Mag)	Durability Tier 1-3
Fists	10	0	0.4	1s	NA	NA
Stone	15	0	0.5	1s	NA	10
Spears (tier 1-3)	20-34-50	40-60-80	1.5s	1s	Draw Speed	70-120-150
Knife (tier 1-3)	15-25-35	25-40-55	1s	0.8s	Draw Speed	30-60-100
Crafting Axe	18-25-38	NA	1.2	1.5	NA	30-50-80
Crafting Pickaxe	20-34-50	NA	2s	2s	NA	40-60-90
Bow	NA	20-80 based on draw	2s	2s	2s	100
Pistol (stretch goal)	NA	80	1.5s	0.2	2s	120

2.2 Crafting & Inventory system

Crafting will use the basic attack animations seen in section 2.1 for each of the crafting tools used harvest:

Tool	Harvest
Stone	Yes
Axe (Tier 1-3)	Yes
Pickaxe (Tier 1-3)	Yes

Crafting will consist of basic 3D harvest animations for the following Stats:

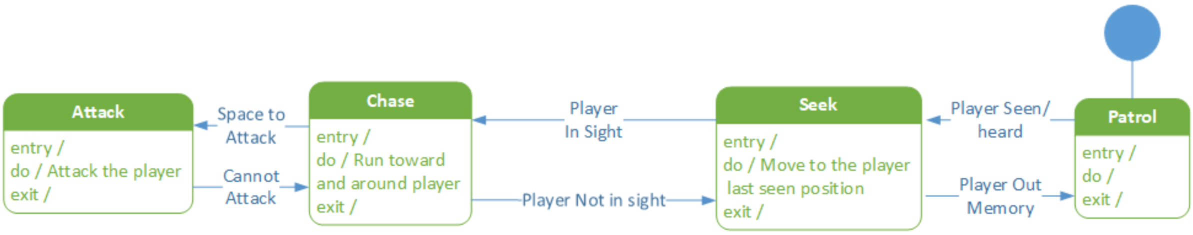
Tool	Material Harvested (Amount per hit)
Stone wood	1
Axe (Tier 1-3) Wood	(1-3) – (4-8) – (9-12)
Pickaxe (Tier 1-3) Metal	(1-3) – (4-8) – (9-12)

The games inventory system will act similarly to other games in the genre with items being picked up/harvested and stored, the player can also place certain items into a hot bar which can then be used to place items in their hands.

2.3 Ai Enemies

The ai in the game is meant to be more of a nuisance than a main threat or antagonist (appendix B), they are there to give the player something extra to think about whilst they are traversing the level.

Figure 1 Zombie State Machine Diagram



2.4 Puzzle/Challenges

Puzzles will take some of the games core concepts and turn them into challenges in which the player must master certain mechanics to pass. The aim is to have around three puzzles based on certain mechanics and then a final challenge that incorporates all three. The main premise is that puzzles revolve around the following:

- Jump Puzzles
- Ranged Puzzles
- Combat Challenges
- Crafting Puzzles

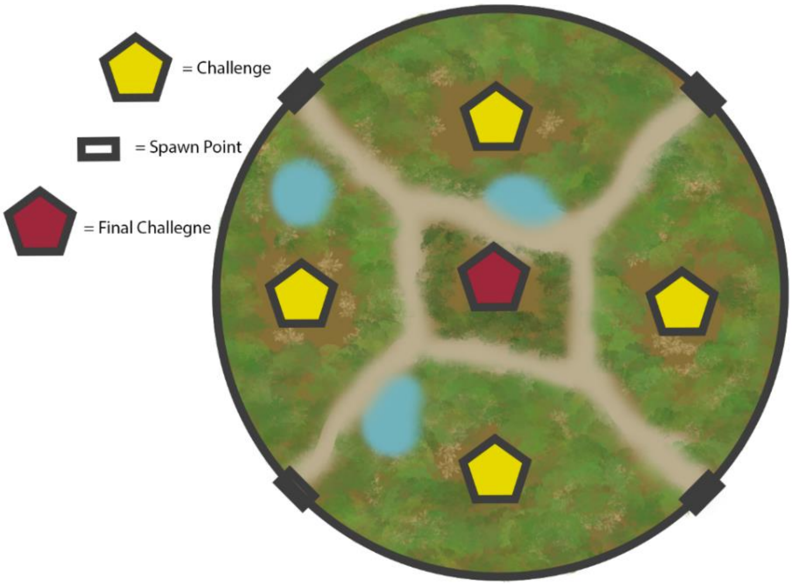
2.5 Respawn

The player will be able to respawn at campfires they have crafted, placed, and saved at. An alternative to this is that upon death, the player respawns as a new competitor, who enters the arena at a chosen location to pick up where the last left off, this could be done through either saving the current game state and reloading the scene, or simply teleporting the player to a new location and dropping their inventory at their last location.

3 Level and Environment

The main environment for the game is a wooded arena constrained inside a forcefield dome, see Appendix D, with areas of water and rock formations. Research for this was mainly drawn from the Hunger games, see Appendix E, using the centre of the dome as a focal point, see Appendix F. The main guidance for the player will be the use of vistas, with challenge points being in open spaces, and having objects and structures that would not normally be expected in forests.

Figure 2 Mock Level Sketchup



4 UI/UX

Ui/UX is inspired by research on games such as Rust (FacePunch, 2021), Dayz (Bohemia Interactive, 2021), and Ark (2021), which are all games within the survival genre.

4.1 Inventory and Crafting

Based on Appendix F, G and H

Figure 3 Inventory WireFrame Design 1

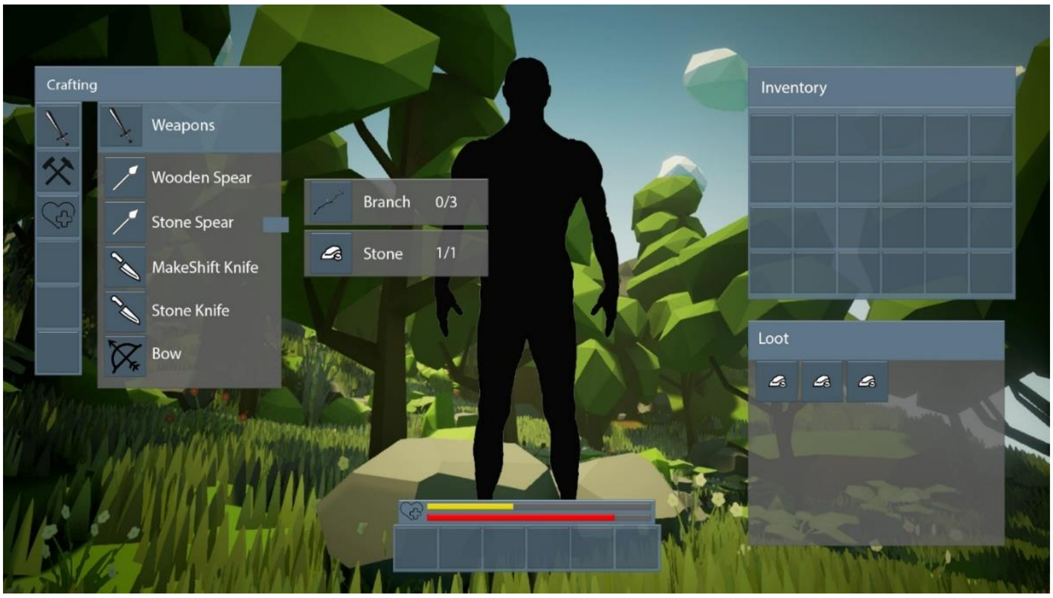
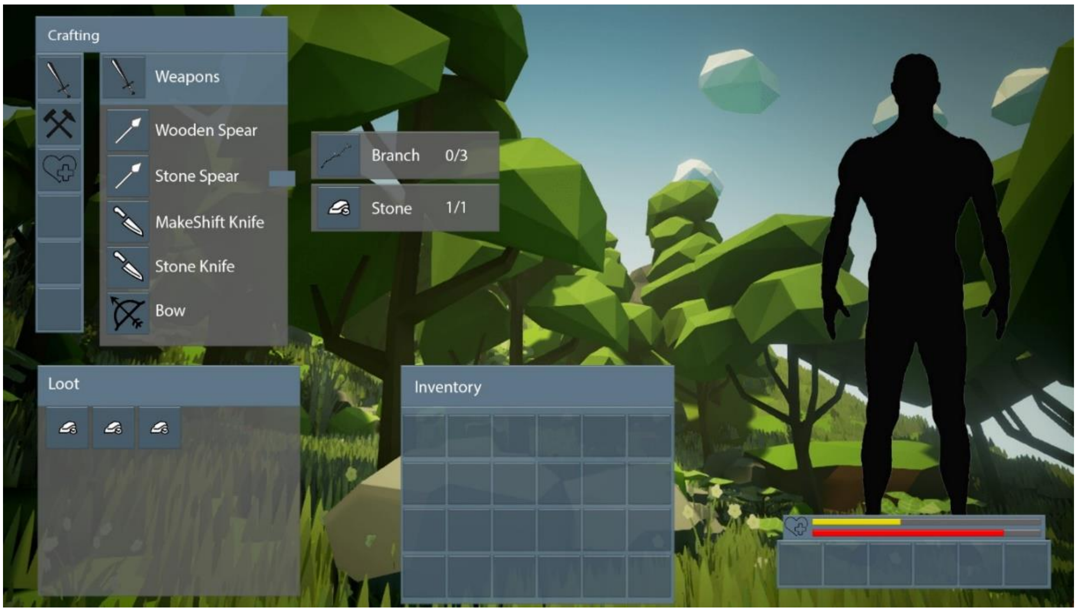


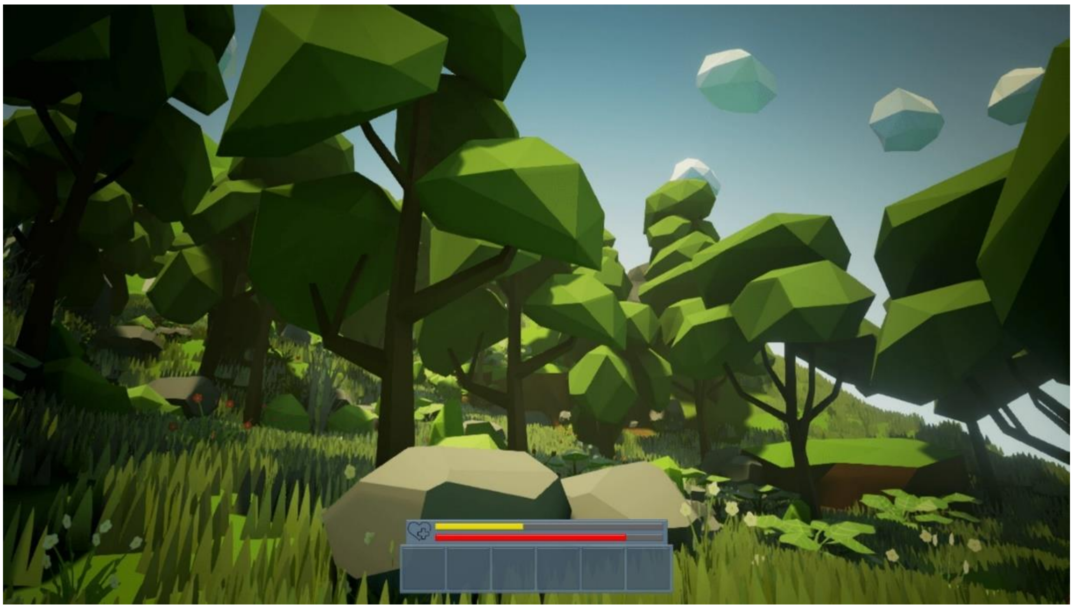
Figure 4 Inventory WireFrame Design 2



4.2 HUD

Based on Appendix I, J and K

Figure 5 Heads Up Display WireFrame 1



5 Dynamic events

Dynamic events are a positive/negative reinforcement system that will reward and punish the player in certain circumstances, some examples being:

- Airdrops for killing multiple zombies.
- Position revealed to attract enemies if the player is avoiding too many.

- Areas of the world become dangerous or hazardous to force the player away from them if they stay there for too long.

6 Sound FX & Music

6.1 Sound FX

Sound FX will be sourced externally from websites such a Freesound.org (2020) and OpenGameArt (2020), however, voice-overs will be done by me. The general sounds needed for the game will be:

Sound	Description
Footsteps	Footsteps used when running or sprinting and jumping
Human Grunts	Attacking, taking damage
Zombie movement	Chasing, attacking, death, footsteps
Weapon thrust and slash	Generic thrush and slash sound of melee weapons
Ranged Weapon sounds	Drawing, reloading, aiming, firing
Crafting Sounds	Ui crafting pings and dings
Harvesting sounds	Hitting trees, rocks, picking things up
Inventory	Selecting an item, equipping and item, dropping items
Puzzle noises	Interactions, moving parts
Ambient sounds	Trees, wind
UI menus	Main Menu, Pause Menu
Voice Lines	Dialoge and Cinematic Voice Overs

6.2 Music

All music used in this project will be from a selection of royalty music packs, purchased as part of a Humble Bundle (2021) package, the overall theme is intended to be heroic and dangerous.

7 Development Plan

As the focus of this project is to showcase gameplay and mechanic design. The development will focus on producing the player and Ai based features first, with a focus on level and environmental factors coming towards the end. The Gantt chart (Appendix B) details the intended work cycle of the project from start to finish. For a breakdown of the Gannt chart, see Appendix M which shows the tasks for each sprint.

Phase 1: Develop core features discussed in section 2 within a white box environment to ensure core project goals are met.

Phase 2: Develop the level and environment ready for implementation of core features, focus on building the player experience within this phase. Stage private testing to obtain feedback on bugs and balancing.

Phase 3: Develop dynamic events (section 4), as a stretch goal, based on the current state of the project.

Phase 4: Utilise slack period over easter break to either, catch up on work over-schedule or polish/juice features with no further development changes. Begin final playtesting with external players for both bug testing and final balance/polish. Being post-mortem 2 weeks before the deadline.

8 Bibliography

Humbe Bundle, 2021. *Humble Software Bundle: Big Music 2*. [online] Humble Bundle. Available at: <https://www.humblebundle.com/software/big-royaltyfree-music-2-software?hmb_source=&hmb_medium=product_tile&hmb_campaign=mosaic_section_1_layout_index_1_layout_type_threes_tile_index_3_c_bigroyaltyfreemusic2_softwarebundle> [Accessed 13 February 2021].

FacePunch, 2021. *Rust | Facepunch*. [online] Rust.facepunch.com. Available at: <https://rust.facepunch.com/> [Accessed 13 February 2021].

Bohemia Interactive, 2021. *DayZ | Bohemia Interactive*. [online] Bohemia Interactive. Available at: <https://www.bohemia.net/games/dayz> [Accessed 13 February 2021].

Ark, 2021. *ARK News*. [online] ARK - Official Community Forums. Available at: <https://survivetheark.com/> [Accessed 13 February 2021].

9 Appendices

Appendix A: Game Overview

Narrative

In a dystopian future, where money matters more than lives and the main forms of entertainment have devolved into primitive shows of combat, instinct and the will to survive, in most cases, death. With many families on the brink of total starvation and squalor, social and economic divides have generated rifts in society that echo back to the dark ages where gladiator combat and life-threatening danger have become the norm. The upper echelons of society take advantage of this and have created the Nexus games, a range of challenges that are televised across the planet to desensitized viewers who bet and wager on the success of competitors.

The player steps into the shoes of Connor, a competitor in the Nexus Survival game, one of the most anticipated and watched games of the year. These games pit individual opponents against a dangerous and hostile environment, rewarding them for putting on a show and punishing them if they do not. This is all to attain the golden sphere and reach the exit with it in hand.

Gameplay

- Crafting
- Combat
- Puzzle/Challenge Solving
- Dynamic Events

Gameplay Is focused on the use of objects found and harvested in the world to craft gear that allows the player to get past numerous enemies and obstacles to complete a range of puzzles/tasks within the world, all whilst responding to dynamic events that will throw them off or present them with new challenges. The main incentive for this will be to gain rewards and one-off items needed to eventually escape with each sector or challenge giving a unique item to help in the player's survival, and one piece of the final puzzle.

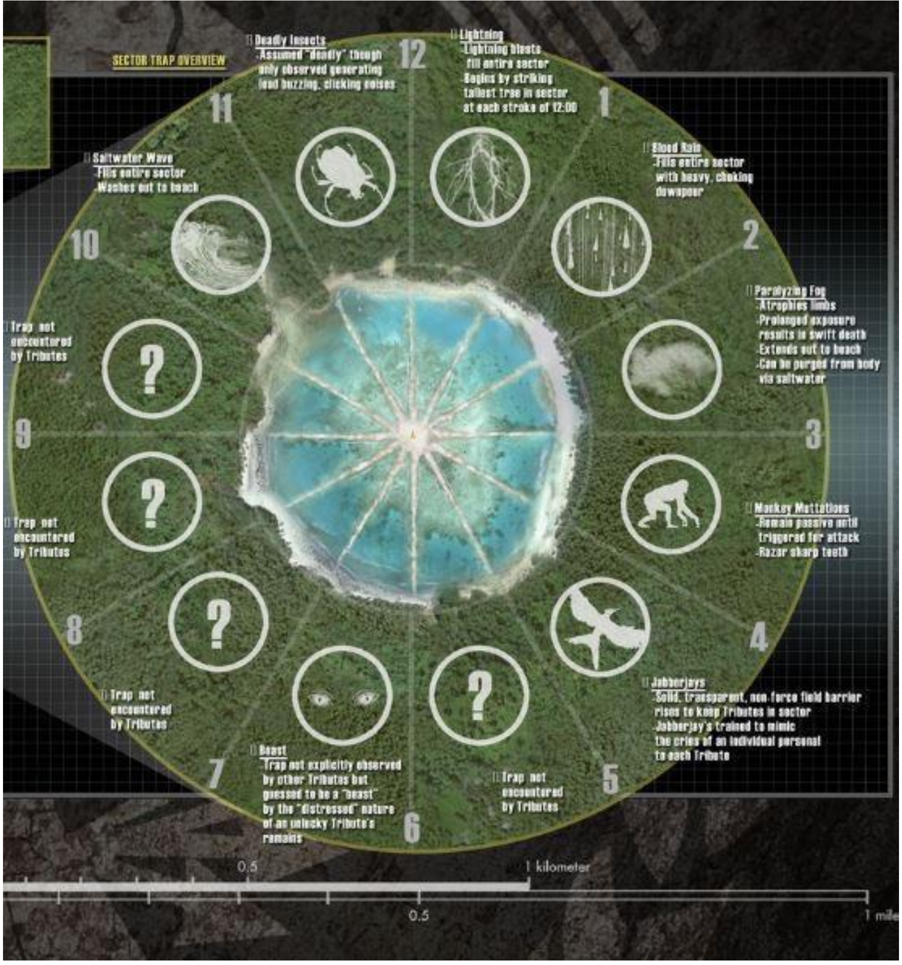
Art

Art for the game is entirely externally sourced from asset packs and stores, the style is low poly due to its abundance but also, it is performance and file size allowing me to focus more on the gameplay systems that I wish to develop over the art style.

Appendix B: Main enemy for the game

Appendix C: Forcefield Example

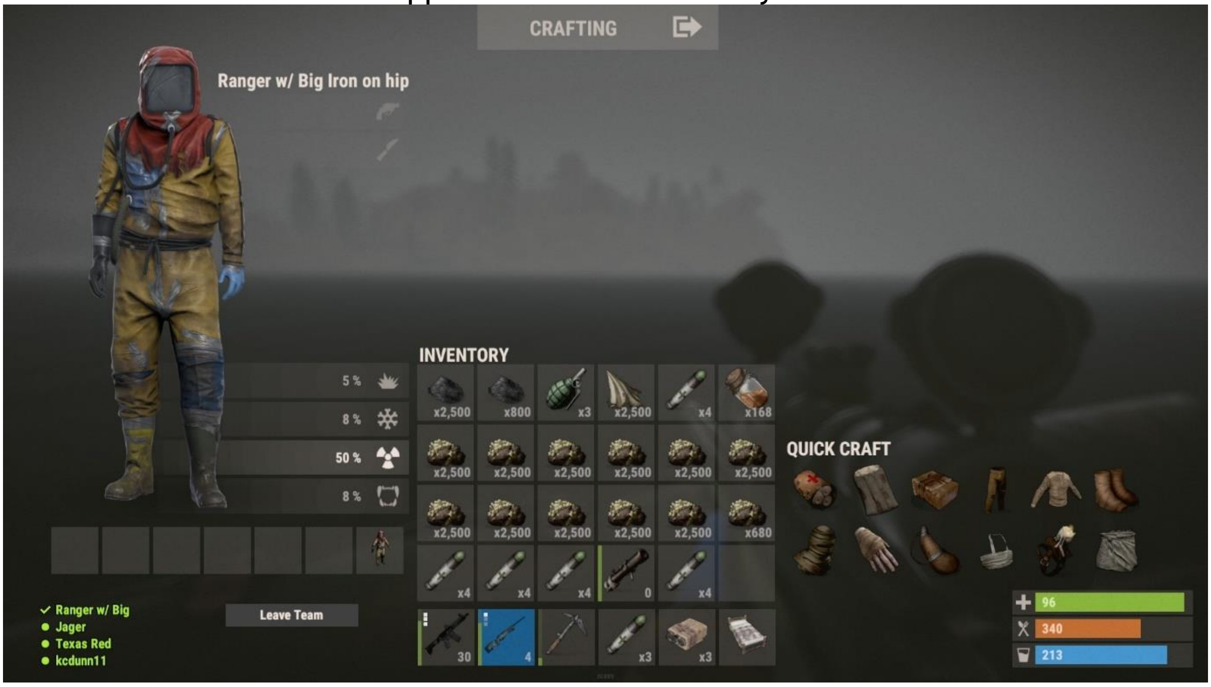
Appendix D: Hunger Games World Layout


SECTOR TRAP OVERVIEW
12
Lightning
-Lightning blasts fill entire sector
-Begins by striking tallest tree in sector at each stroke of 12:00
1
Blood Rain
-Fills entire sector with heavy, choking downpour
2
Paralyzing Fog
-Atrophies limbs
-Prolonged exposure results in swift death
-Extends out to beach
-Can be purged from body via saltwater
3
Monkey Muttations
-Remain passive until triggered for attack
-Razor sharp teeth
4
Jabberjays
-Solid, transparent, non-force field barrier rises to keep Tributes in sector
-Jabberjay's trained to mimic the cries of an individual personal to each Tribute
5
Trap not encountered by Tributes
6
Beast
-Trap not explicitly observed by other Tributes but guessed to be a "beast" by the "distressed" nature of an unlucky Tribute's remains
7
Trap not encountered by Tributes
8
Trap not encountered by Tributes
9
Trap not encountered by Tributes
10
Saltwater Wave
-Fills entire sector
-Washes out to beach
11
Deadly Insects
-Assumed "deadly" though only observed generating loud buzzing, clicking noises
?
?
?
?
0.5
1 kilometer
0.5
1 mile

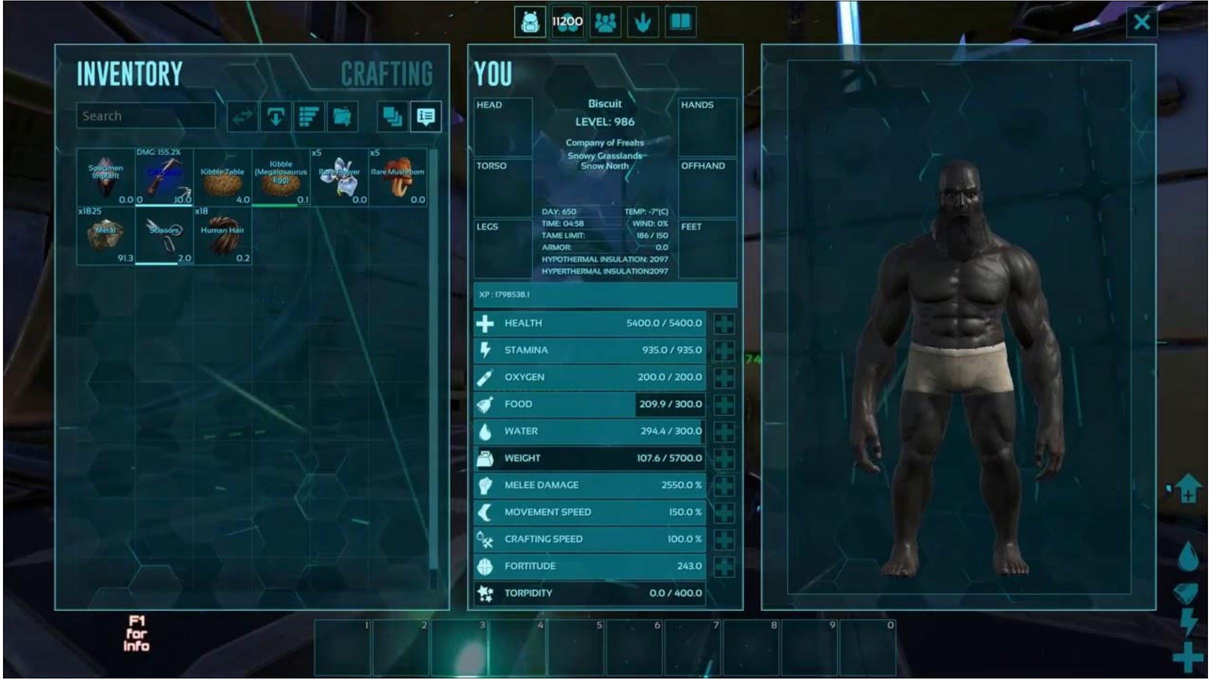
Appendix E: Focal Hunger Games Centre

CatchingFire_Cornucopia3d_DB_07.24.12.jpg

Appendix F: Rust Inventory



Appendix G: Ark Inventory



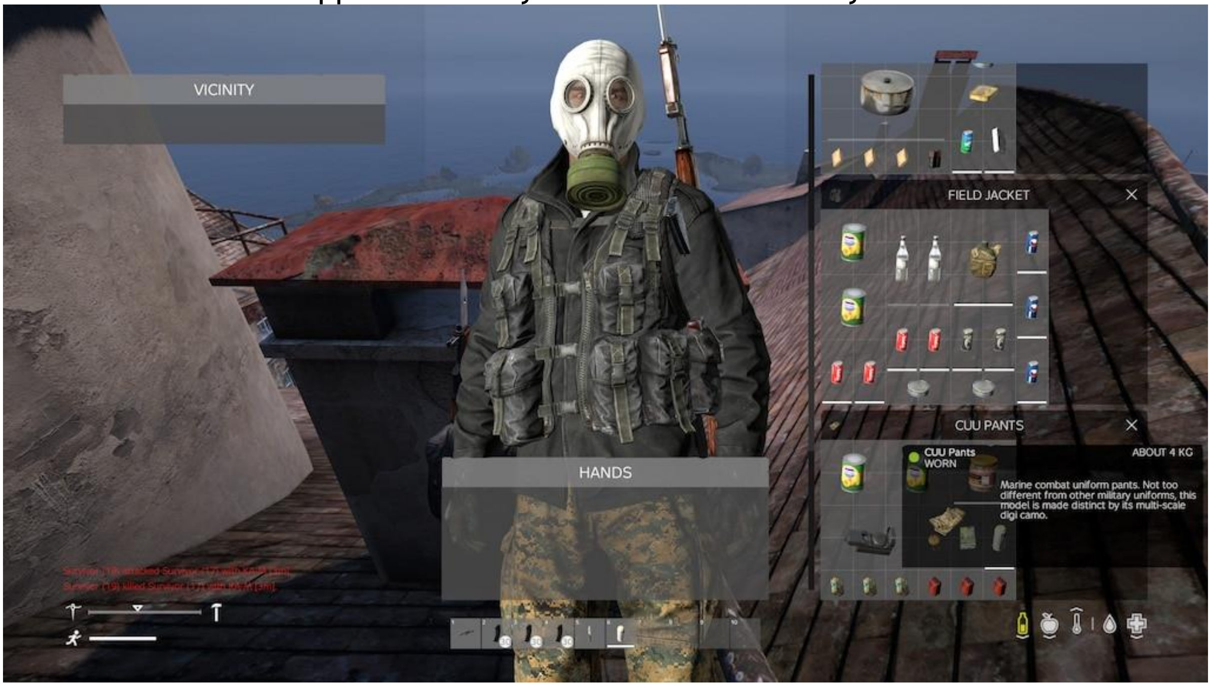
Biscuit
LEVEL: 986
Company of Freaks
Snowy Grasslands-
Snow North

DAY: 650 TEMP: -7°(C)
TIME: 04:58 WIND: 0%
TAME LIMIT: 186 / 150
ARMOR: 0.0
HYPOTHERMAL INSULATION: 2097
HYPERTHERMAL INSULATION2097

XP : 1798538.1

HEALTH	5400.0 / 5400.0
STAMINA	935.0 / 935.0
OXYGEN	200.0 / 200.0
FOOD	209.9 / 300.0
WATER	294.4 / 300.0
WEIGHT	107.6 / 5700.0
MELEE DAMAGE	2550.0 %
MOVEMENT SPEED	150.0 %
CRAFTING SPEED	100.0 %
FORTITUDE	243.0
TORPIDITY	0.0 / 400.0

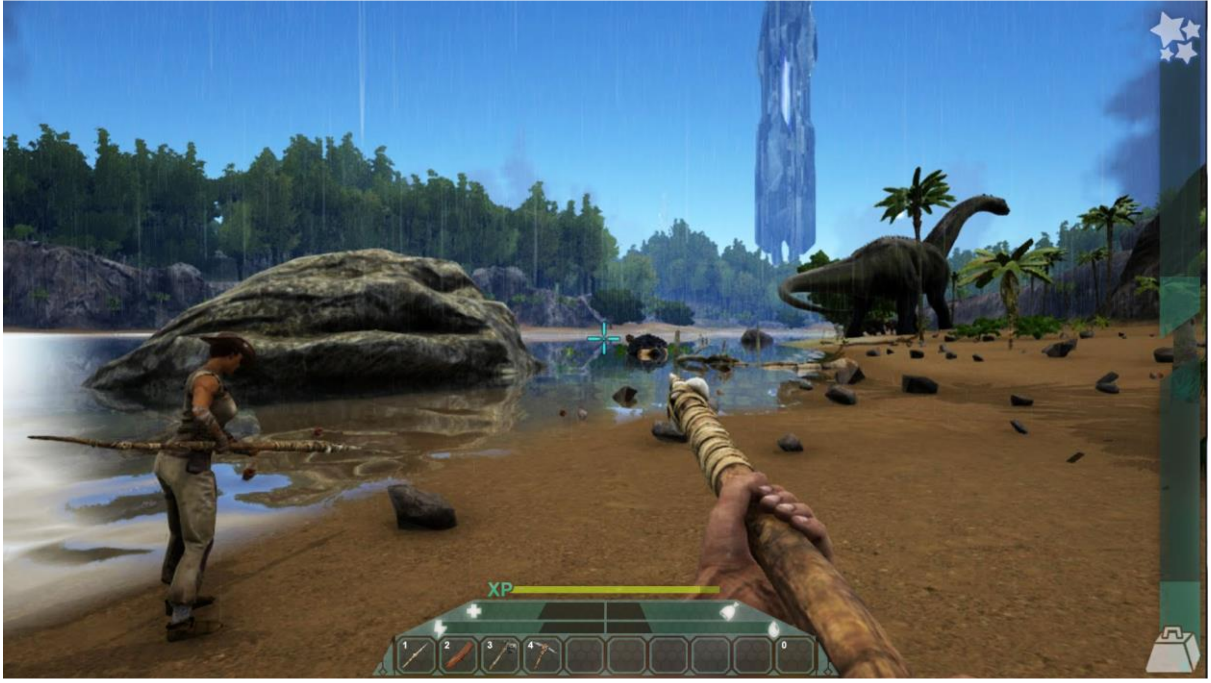
Appendix H: Dayz Standalone Inventory



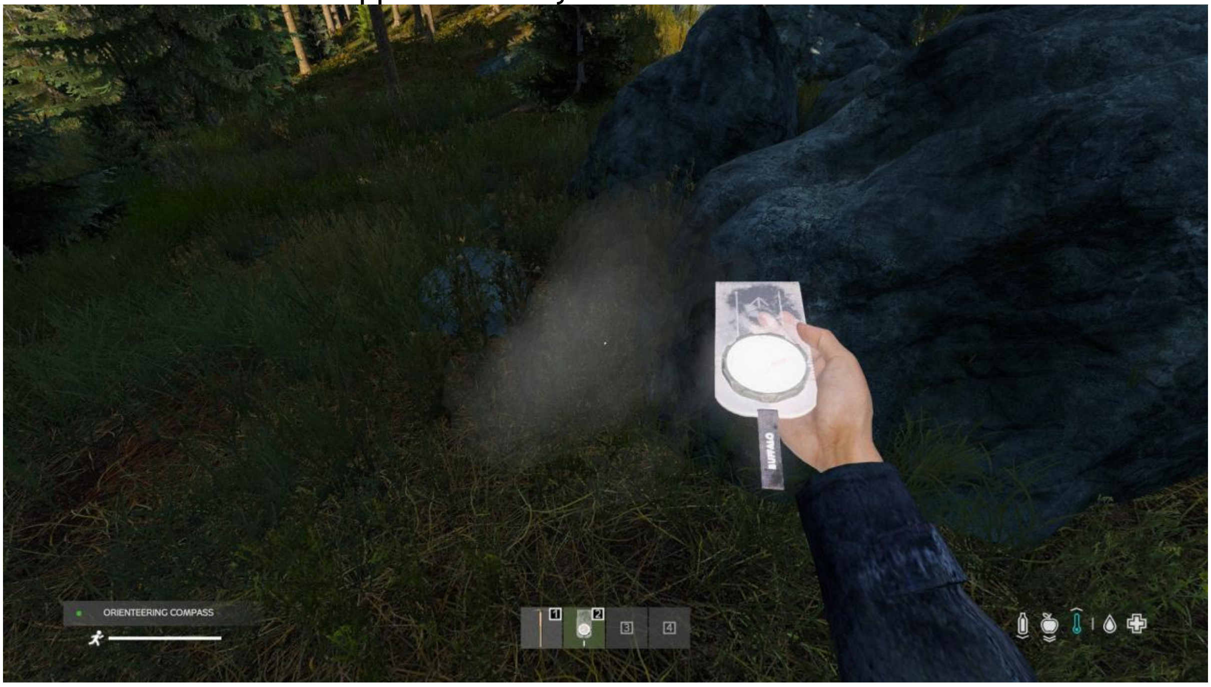
CUU Pants
WORN
ABOUT 4 KG

Marine combat uniform pants. Not too different from other military uniforms, this model is made distinct by its multi-scale digi camo.

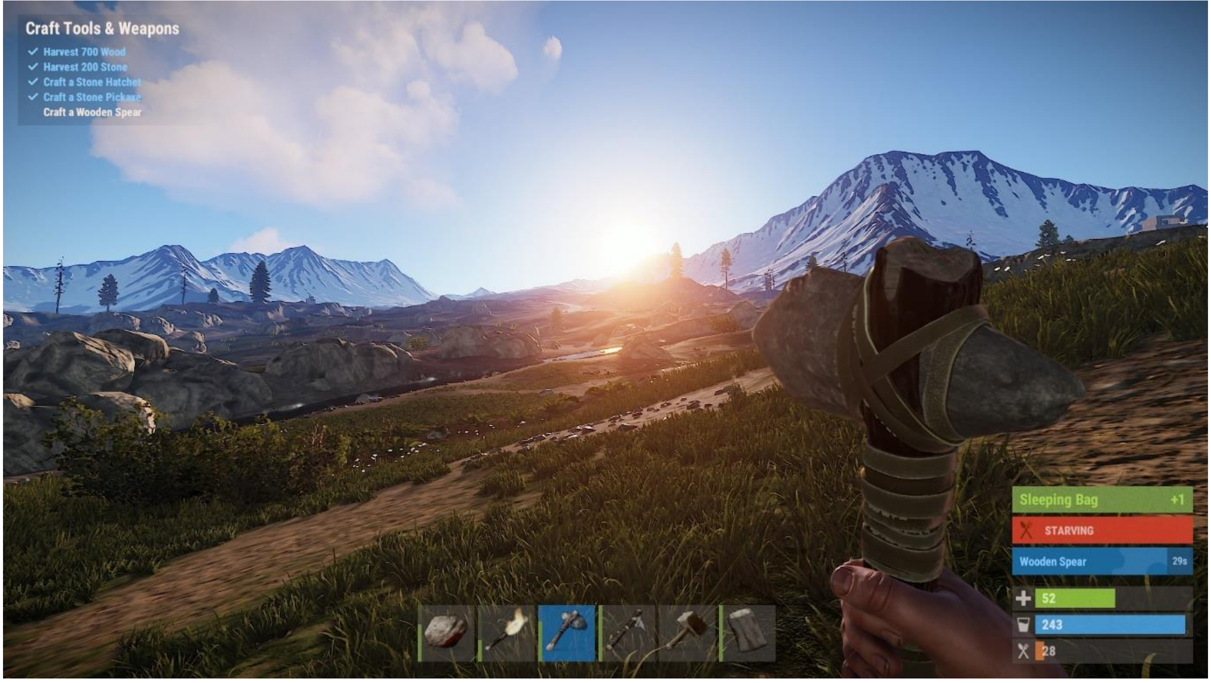
Appendix I: Ark HUD



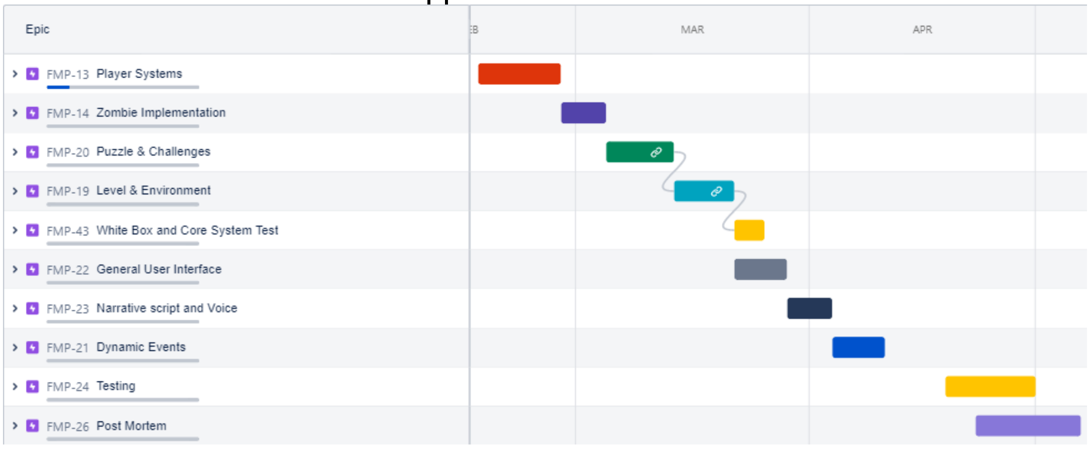
Appendix J: DayZ Standalone Hud



Appendix K: Rust HUD



Appendix L: Gantt Chart

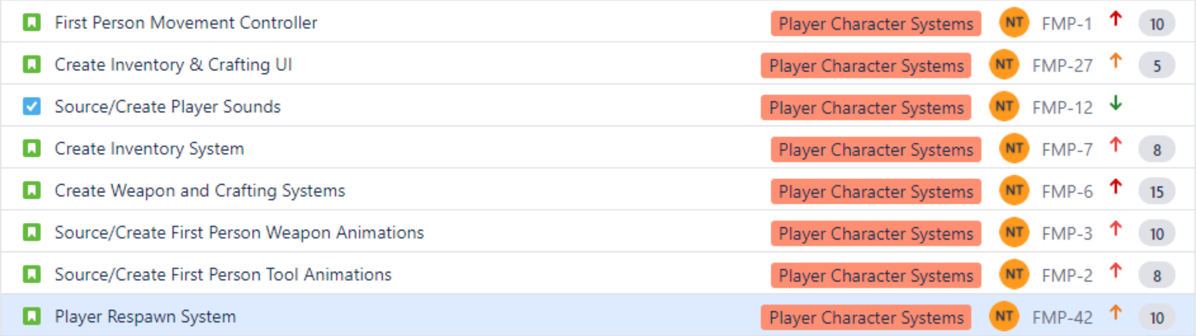


Epic
FMP-13 Player Systems
FMP-14 Zombie Implementation
FMP-20 Puzzle & Challenges
FMP-19 Level & Environment
FMP-43 White Box and Core System Test
FMP-22 General User Interface
FMP-23 Narrative script and Voice
FMP-21 Dynamic Events
FMP-24 Testing
FMP-26 Post Mortem

Appendix M: Task Breakdown

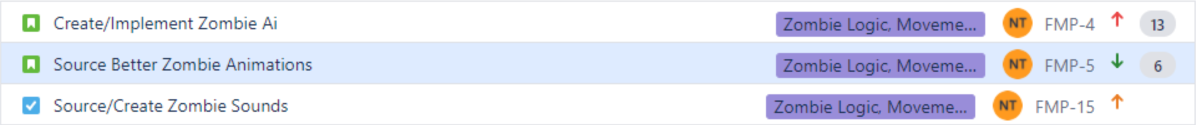
Presented in the same order as the Gannt Chart.

Player Systems:



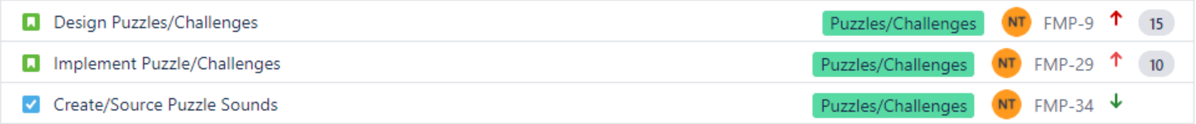
🟩	First Person Movement Controller	Player Character Systems	NT	FMP-1	↑	10
🟩	Create Inventory & Crafting UI	Player Character Systems	NT	FMP-27	↑	5
☑	Source/Create Player Sounds	Player Character Systems	NT	FMP-12	↓	
🟩	Create Inventory System	Player Character Systems	NT	FMP-7	↑	8
🟩	Create Weapon and Crafting Systems	Player Character Systems	NT	FMP-6	↑	15
🟩	Source/Create First Person Weapon Animations	Player Character Systems	NT	FMP-3	↑	10
🟩	Source/Create First Person Tool Animations	Player Character Systems	NT	FMP-2	↑	8
🟩	Player Respawn System	Player Character Systems	NT	FMP-42	↑	10

Zombie Implementation:



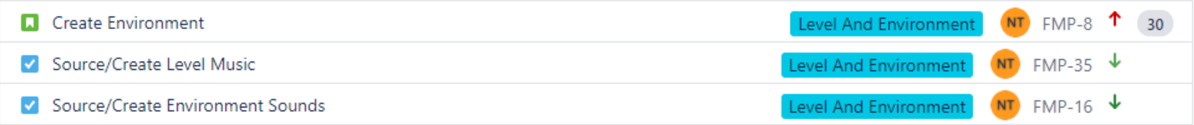
🟩	Create/Implement Zombie Ai	Zombie Logic, Moveme...	NT	FMP-4	↑	13
🟩	Source Better Zombie Animations	Zombie Logic, Moveme...	NT	FMP-5	↓	6
☑	Source/Create Zombie Sounds	Zombie Logic, Moveme...	NT	FMP-15	↑	

Puzzle & Challenges:



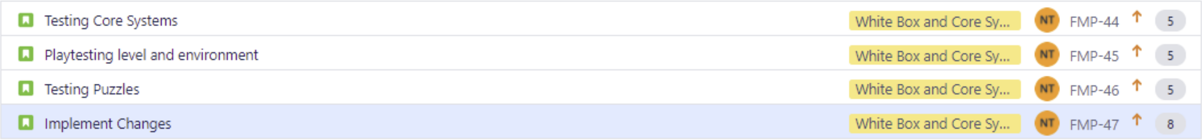
🟩	Design Puzzles/Challenges	Puzzles/Challenges	NT	FMP-9	↑	15
🟩	Implement Puzzle/Challenges	Puzzles/Challenges	NT	FMP-29	↑	10
☑	Create/Source Puzzle Sounds	Puzzles/Challenges	NT	FMP-34	↓	

Level & Environment:



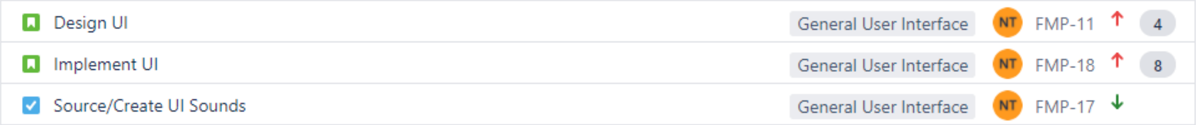
🟩	Create Environment	Level And Environment	NT	FMP-8	↑	30
☑	Source/Create Level Music	Level And Environment	NT	FMP-35	↓	
☑	Source/Create Environment Sounds	Level And Environment	NT	FMP-16	↓	

White Box & Core Systems Test



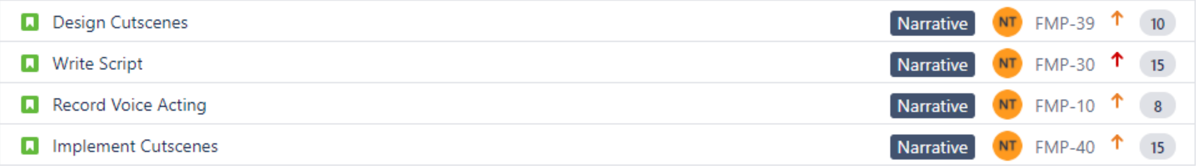
🟩	Testing Core Systems	White Box and Core Sy...	NT	FMP-44	↑	5
🟩	Playtesting level and environment	White Box and Core Sy...	NT	FMP-45	↑	5
🟩	Testing Puzzles	White Box and Core Sy...	NT	FMP-46	↑	5
🟩	Implement Changes	White Box and Core Sy...	NT	FMP-47	↑	8

General User Interface:



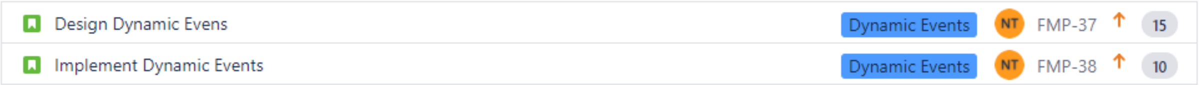
🟩	Design UI	General User Interface	NT	FMP-11	↑	4
🟩	Implement UI	General User Interface	NT	FMP-18	↑	8
☑	Source/Create UI Sounds	General User Interface	NT	FMP-17	↓	

Narrative Script and Voice:



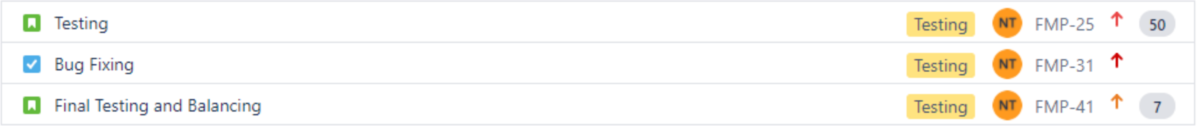
🟩	Design Cutscenes	Narrative	NT	FMP-39	↑	10
🟩	Write Script	Narrative	NT	FMP-30	↑	15
🟩	Record Voice Acting	Narrative	NT	FMP-10	↑	8
🟩	Implement Cutscenes	Narrative	NT	FMP-40	↑	15

Dynamic Events:



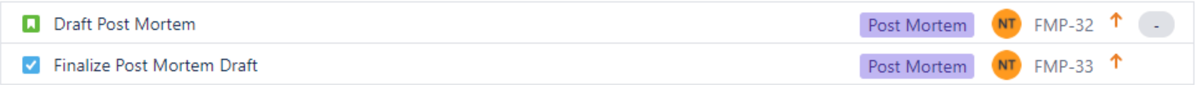
🟩	Design Dynamic Evens	Dynamic Events	NT	FMP-37	↑	15
🟩	Implement Dynamic Events	Dynamic Events	NT	FMP-38	↑	10

Testing:



🟩	Testing	Testing	NT	FMP-25	↑	50
☑	Bug Fixing	Testing	NT	FMP-31	↑	
🟩	Final Testing and Balancing	Testing	NT	FMP-41	↑	7

Post-Mortem



🟩	Draft Post Mortem	Post Mortem	NT	FMP-32	↑	-
☑	Finalize Post Mortem Draft	Post Mortem	NT	FMP-33	↑	

6 Project Development

This section of my report servers to detail the progression of my project as well as all changes made. I will cover each area of the project, discussing their basic implementation as well as any issues/success' I had with them. Succeeding this is an evaluation and reflective account of how production on this project went. Finally, you can see a week-by-week breakdown of feature implementation in Appendix A.

6.1 Player Character Systems

6.1.1 Character Controller and First-Person View

For the character's movement, I started with a basic guide (Brackeys, 2019) and sourced all animations from an open-source library called Mixamo (Adobe, 2021). However, I made my edits to it as well as adding a crouch feature, clamping sprint directions, and adding stamina. I did this because, like other survival games, I wanted the movement to feel realistic, as well as making interactions with enemies in the game force players to think about how they are going to approach.

Secondly, to enhance realism, I decided the character's body should be visible to the player, however, this proved to be harder to achieve than I expected. The problem began when trying to simply place the camera in front of the character model's head and parenting it to the rig. Whilst this some-what worked there were issues with other body parts such as the chest, shoulders and head clipping into the camera's frustum. To solve this issue, I found that I could shrink certain parts of the rig which in turn would shrink the model, preventing and clipping.

The second problem came when I tried importing some punch animations. I realised that the arms would always punch at a set rotation since the animations were baked and that I could not change this to follow the direction of the camera (The height the player is looking). This problem would have persisted with any animation I tried to use. The only solution for this issue was to take the character model into 3ds max and delete all parts of the mesh excluding the arms, and then reimport the model whilst preserving the rig. I could then set the arms rotation to follow the camera without disturbing the legs. You can see the result of these two fixes in figure 6.

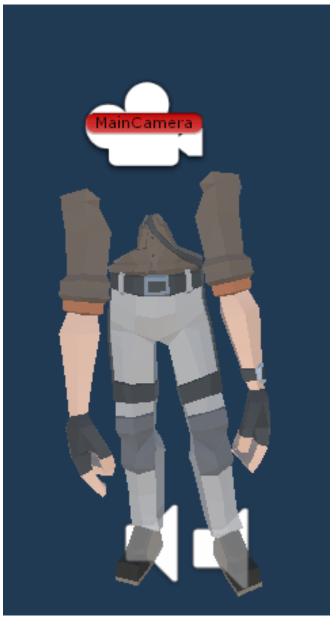


Figure 6 Separate rigs for 1st person arms and legs

Finally, adding two separate rigs and shrinking the chest resulted in the shadow missing half a body, the only solution for this was to create a third rig that would copy the animations played by the two rigs, whilst only rendering shadows, and not it is mesh, in their place.

6.1.2 Inventory & Crafting System

The main tool for survival is the inventory and crafting system, used for creating, storing, and using weapons, tools, and medicines, it is heavily inspired by Rust (FacePunch, 2021). The original plan for this system was to have an all-in-one inventory crafting UI screen that could be toggled for easy and quick access, the reasoning for this, rather than using separate screens or menus, was due to the pace of the game with the player being attacked frequently. It required quick fluid access with as few inputs as possible. As can be seen in figure 6, all main elements regarding looting, crafting and what the player is currently holding are compacted into a single screen.

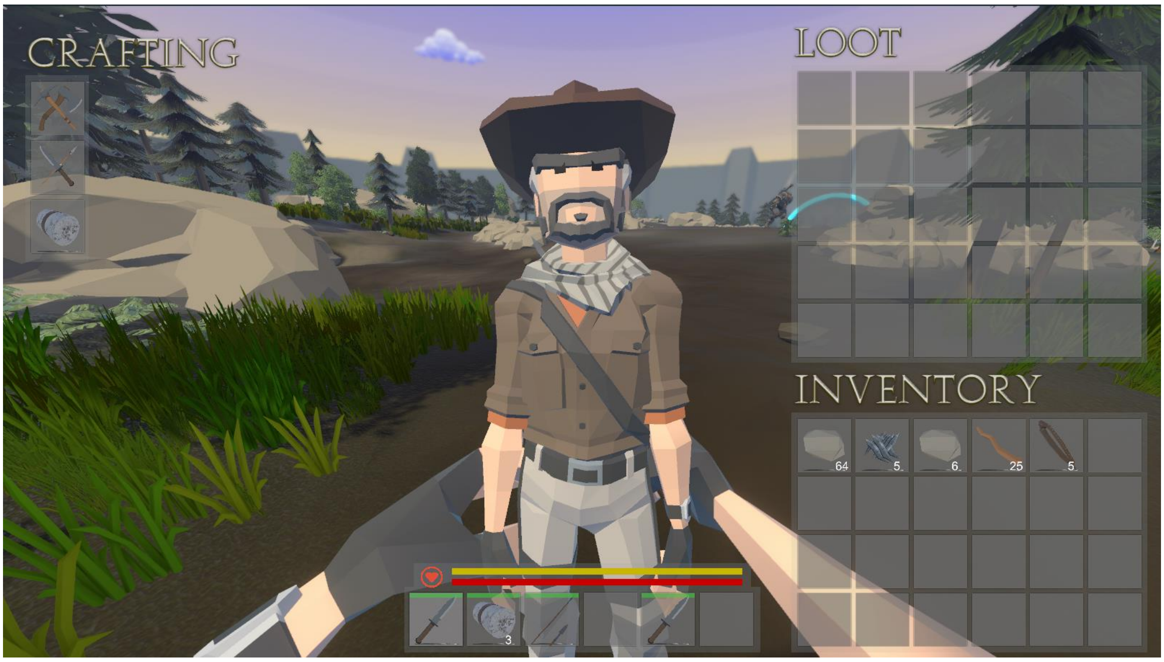


Figure 7 Inventory Screen

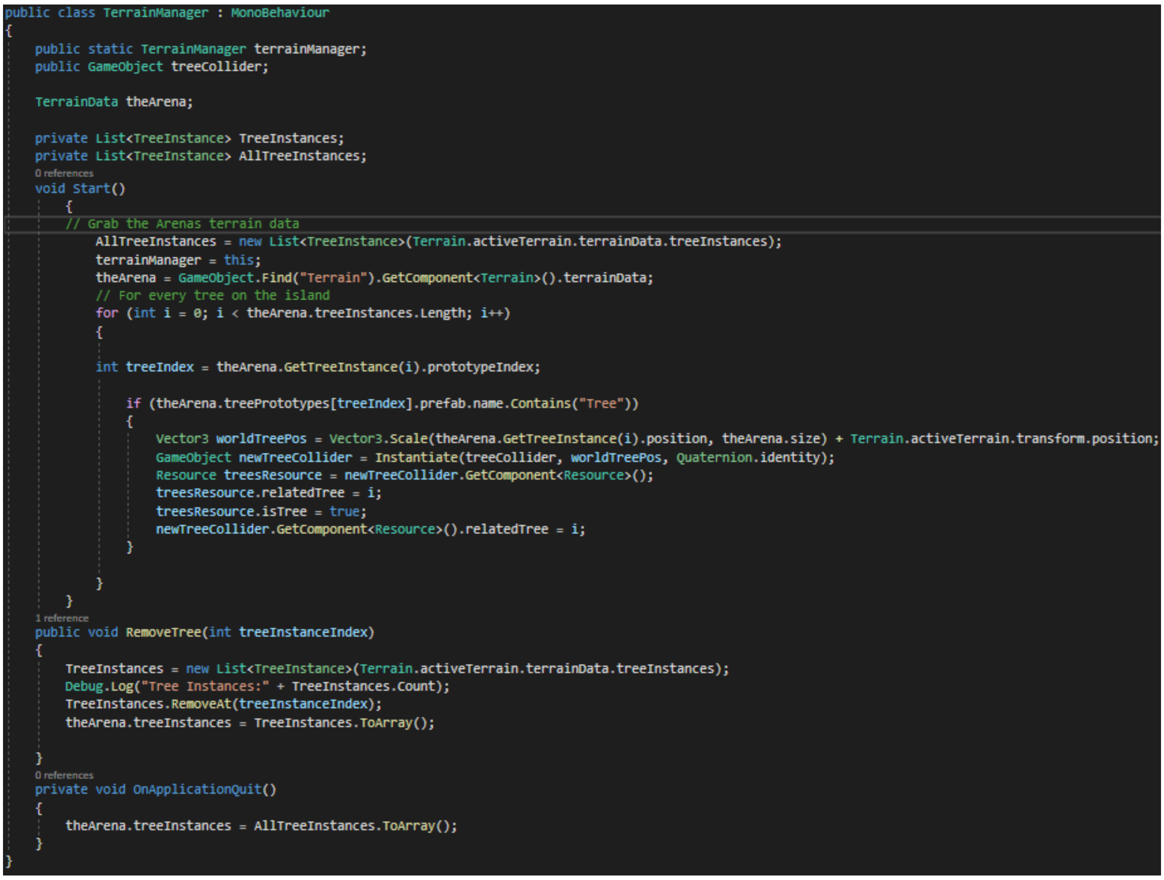
Crafting works with an item tier system, (Wood, Stone, Metal) the player must gather resources from the environment, combining them into the items they will need. The basic resources are Twine, Wood, Stone and Metal Fragments, which are then combined in various crafting recipes.

To make reaching higher tiers more challenging, I chose to make twine and metal fragments are more of a rarity within the world, being a random secondary resource from harvesting though mainly being found as lootables from high-risk areas on the map and challenges. The intention of this is to entice players towards interaction with these parts of the level whilst also giving them a greater sense of accomplishment when crafting to higher tiers.

6.1.3 Resource Gathering

Without being able to gather resources the player would not be able to get the gear needed to help them fight the undead and complete some of the challenges. Gathering works through using correct tools on certain elements within the environment, such as trees and mineable rocks. The system works by detecting hits and then returning the appropriate resource from that source (amount based on the tier of a tool). Once a resource deposit or tree has been used up, it will be destroyed, dropping additional resources on the ground for the player to pick up.

A big issue I faced with this system was during level creation and the use of Unity’s tree painter terrain tool. Whilst painted trees are automatically given colliders, they do not carry over any script or other data from their prefabs. After researching the problem (Unity Answers, 2021), I came up with the following solution for the problem (seen in figure 8. The script takes all placed trees saved in terrain data at the start and allocates them with a prefab containing the appropriate resource scripts, allowing them to be accessible and removed from the scene when harvested.



```
public class TerrainManager : MonoBehaviour
{
    public static TerrainManager terrainManager;
    public GameObject treeCollider;

    TerrainData theArena;

    private List<TreeInstance> TreeInstances;
    private List<TreeInstance> AllTreeInstances;
    0 references
    void Start()
        {
        // Grab the Arenas terrain data
            AllTreeInstances = new List<TreeInstance>(Terrain.activeTerrain.terrainData.treeInstances);
            terrainManager = this;
            theArena = GameObject.Find("Terrain").GetComponent<Terrain>().terrainData;
            // For every tree on the island
            for (int i = 0; i < theArena.treeInstances.Length; i++)
            {

            int treeIndex = theArena.GetTreeInstance(i).prototypeIndex;

                if (theArena.treePrototypes[treeIndex].prefab.name.Contains("Tree"))
                {
                    Vector3 worldTreePos = Vector3.Scale(theArena.GetTreeInstance(i).position, theArena.size) + Terrain.activeTerrain.transform.position;
                    GameObject newTreeCollider = Instantiate(treeCollider, worldTreePos, Quaternion.identity);
                    Resource treesResource = newTreeCollider.GetComponent<Resource>();
                    treesResource.relatedTree = i;
                    treesResource.isTree = true;
                    newTreeCollider.GetComponent<Resource>().relatedTree = i;
                }

            }
        }
    1 reference
    public void RemoveTree(int treeInstanceIndex)
    {
        TreeInstances = new List<TreeInstance>(Terrain.activeTerrain.terrainData.treeInstances);
        Debug.Log("Tree Instances:" + TreeInstances.Count);
        TreeInstances.RemoveAt(treeInstanceIndex);
        theArena.treeInstances = TreeInstances.ToArray();

    }
    0 references
    private void OnApplicationQuit()
    {
        theArena.treeInstances = AllTreeInstances.ToArray();
    }
}
```

Figure 8 Terrain Manager Script

Finally, for stone deposits, I decided to have them respawn after a set delay, as they were not as common as trees and throughout a session could easily be fully used up. To allow for greater playability having them respawn was the best option and has worked well.

6.1.4 Player Combat & Equip-able Items.

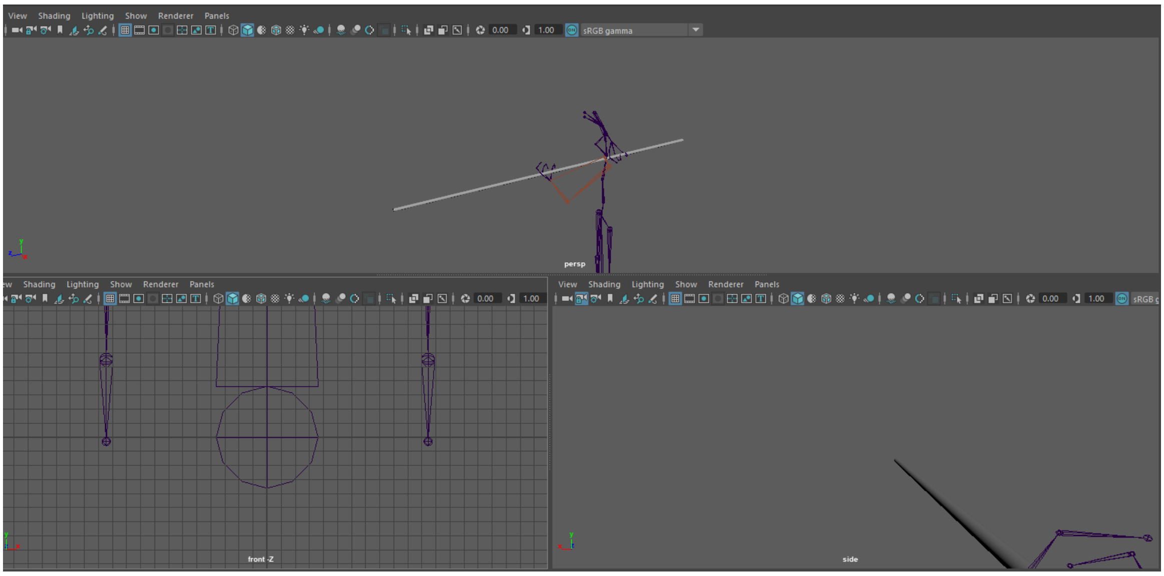
Combat in the game is used for fighting off the zombies that spawn periodically across the map and in certain challenges. It consists of key mechanics such as attacking and throwing, with the later decision to add blocking and parrying.

During each attack animation, the equipped items collider is set active for a brief period. Upon collision with an object, be it an enemy or environmental piece, an associated particle effect is played at the hit position. A sound is also

played that is either chosen from a default library, or through accessing a hit sound script attached to any object that requires a unique sound.
Throwing weapons can be used to hit targets at range and works differently based on aerodynamic and non-aerodynamic objects such as spears and axes, the first flies true through the air and embeds in its target and the second bounces off.

Whilst testing combat with zombies and getting feedback from course mates, I noticed that situations in which the player is outnumbered could prove difficult. With multiple zombies attacking at once combat felt very defenceless, if the player ran out of stamina and could not run away their only option was to try and attack. Based on these results the only option was to add a blocking feature allowing defence from the front against multiple attackers. Blocking works by holding the right mouse button, I added a short parry window at the beginning of the animation that causes zombies to stagger backwards. Blocking will take durability off the equipped weapon.

One issue I have had from the very beginning of this project was sourcing the first-person animations for the different items and objects used in the game. When it came down to it the only solution was for me to take the rig from the asset pack and animate it myself in Autodesk Maya. The problem with this was that whilst I have previously used Maya for animating I could not successfully generate an inverse kinematic controller for the joints on the model. This resulted in animations created for the game being done using forward kinematics, which was very time-consuming. I also struggled to get a good reference on what animations would look like in-game from the camera's perspective so had to keep going back and adjust it.



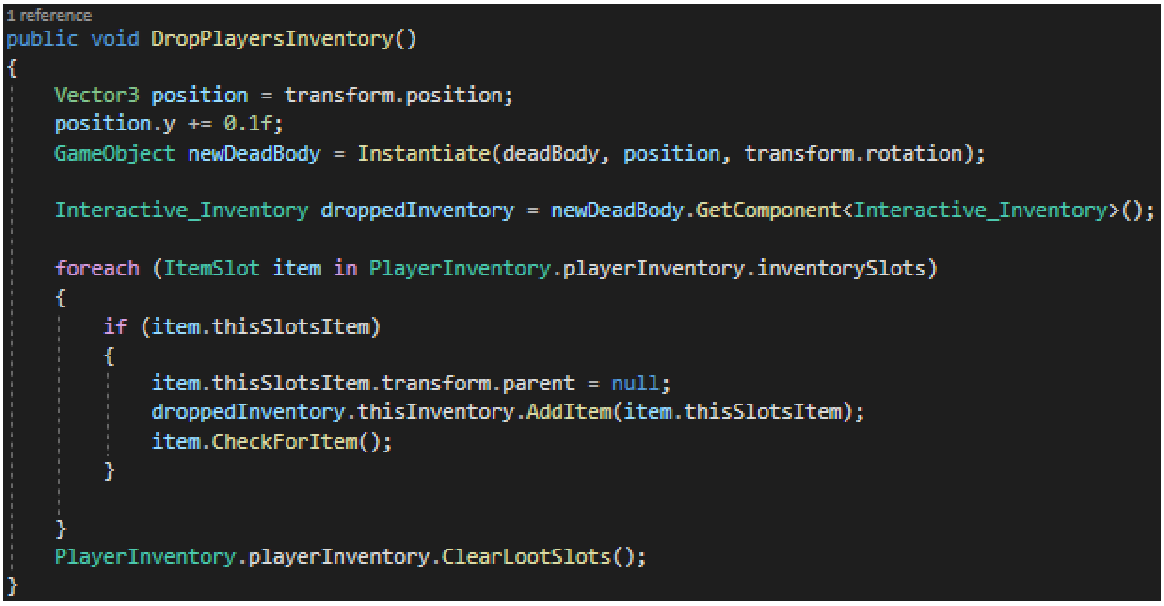
Later, when animating I found a way to include the model along with the original rig, this helped a lot in being able to judge how animations were looking by eye.

A decision I tool to save time was to take advantage of Unity’s interpolate feature. Using the animator's transition features which it automatically calculates joint movements from one animation to another. This saved time on not having to create animations such as transitions from holding to throwing or holding to blocking and worked with switching weapons and saved me a lot of time.

6.1.5 Player Death

With the number of Zombies that inhabit the arena and the giant Iron Golem that needs to be killed at the end of the game, some players may likely fall victim and be killed.

Aligning this with the narrative of the game I needed to create a respawn system that would allow players to re-enter the arena as a new competitor whilst also being able to access their old loot. To this end I made use of the lootable dead bodies, instantiating a prefabbed version, and passing the players inventory into it.



```
1 reference
public void DropPlayersInventory()
{
    Vector3 position = transform.position;
    position.y += 0.1f;
    GameObject newDeadBody = Instantiate(deadBody, position, transform.rotation);

    Interactive_Inventory droppedInventory = newDeadBody.GetComponent<Interactive_Inventory>();

    foreach (ItemSlot item in PlayerInventory.playerInventory.inventorySlots)
    {
        if (item.thisSlotsItem)
        {
            item.thisSlotsItem.transform.parent = null;
            droppedInventory.thisInventory.AddItem(item.thisSlotsItem);
            item.CheckForItem();
        }

    }
    PlayerInventory.playerInventory.ClearLootSlots();
}
```

Once this has happened the player is then moved back to the entrance pipe, which repeats its initial entrance from the beginning of the game, seen in the video at the start of this post.

One thing I struck lucky with when creating this feature was the use of the death animation. Since the player character is just a pair of arms and legs, I was worried that when they died and any animation played, seeing missing body parts would be easy and would break any immersion. Secondly, as the camera was not attached to the rig it would not move with the body as it fell. However, after going through a few different death animations found on Mixamo I found one that worked perfectly, which is used in the game now.

To fix the camera issue I created a death camera which is attached to the head in the rig, it is switched to seamlessly as the player dies so the player cannot tell.

6.1.6 Future work

Most of the future work I plan to do for the game will mainly come for the player systems, with improvements to the polish of animations, quality of life improvements for the Inventory systems, and the possible addition of new equipment items and resource types. The reason for this would be to make the game feel more professional and complete, as well as fulfilling the project proposal in its entirety.

6.2 Zombies

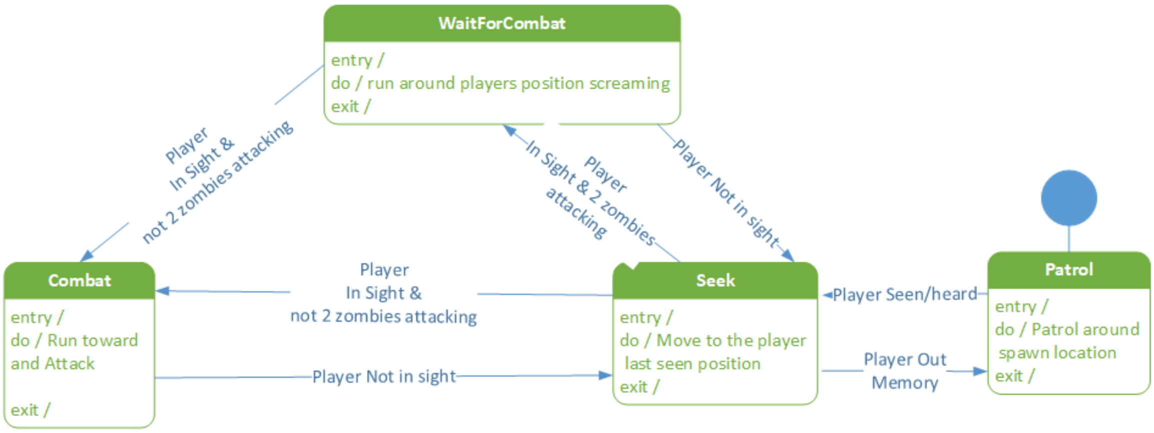
6.2.1 Zombie Finite State Machine

Zombies are the main antagonist of the game, they are docile however become enraged at the sight of the player, chasing, and attacking them ferociously.

The ai runs on a Finite State machine Algorithm (Millington, I. and Funge, J. 2009), relying on a sensory system to react and attack the player as and when they come into contact. I decided that a state machine was the best design choice for the zombie ai based on two factors. Firstly, I never intended for the zombies to be super intelligent, they are in the world to be a nuisance and to force the player to constantly think and make sure that they are stocked with equipment, this meant that the algorithm on which they ran did not need to provide hugely complex functionality. State machines are known to be a simplistic algorithm, by comparison, working best at a smaller scale allowing for a quicker production process (Millington and Funge, 2009; Sweetser and Wiles, 2002).

Secondly, state machines are considered very efficient and effective (Sweetser and Wiles, 2002; Gill, 2014). This was a big factor for me as I needed to keep each agent's computational cost as low as possible since there would be many of them active in the scene at any given time.

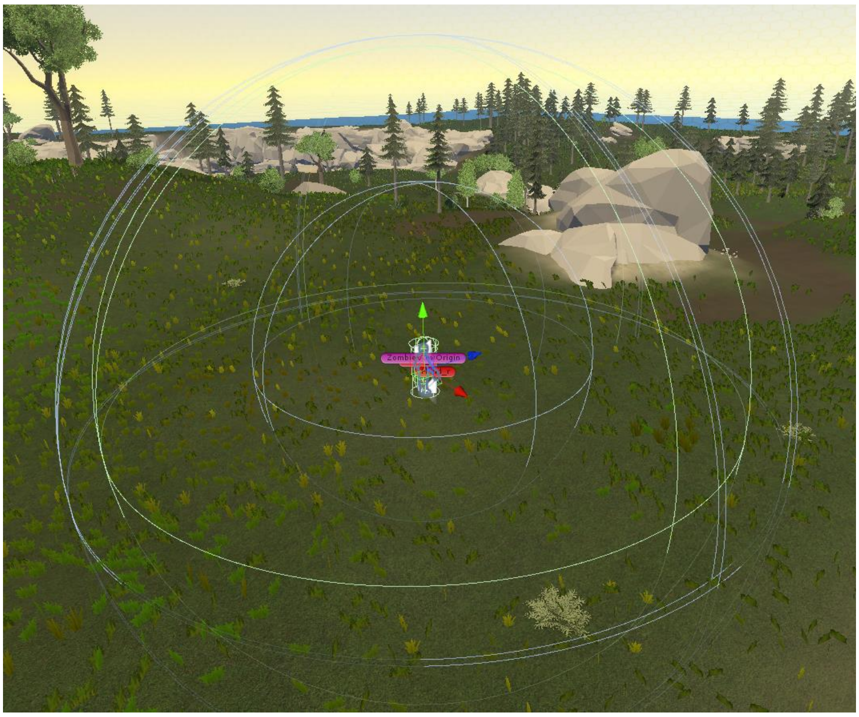
Figure x shows the final design of the Zombie ai they have access to four different states based on the stimuli they receive.


WaitForCombat
entry /
do / run around players position screaming
exit /
Player In Sight & not 2 zombies attacking
In Sight & 2 zombies attacking Player
Player Not in sight
Combat
entry /
do / Run toward and Attack
exit /
Player In Sight & not 2 zombies attacking
Player Not in sight
Seek
entry /
do / Move to the player last seen position
exit /
Player Seen/heard
Player Out Memory
Patrol
entry /
do / Patrol around spawn location
exit /

6.2.2 Sensing and Data

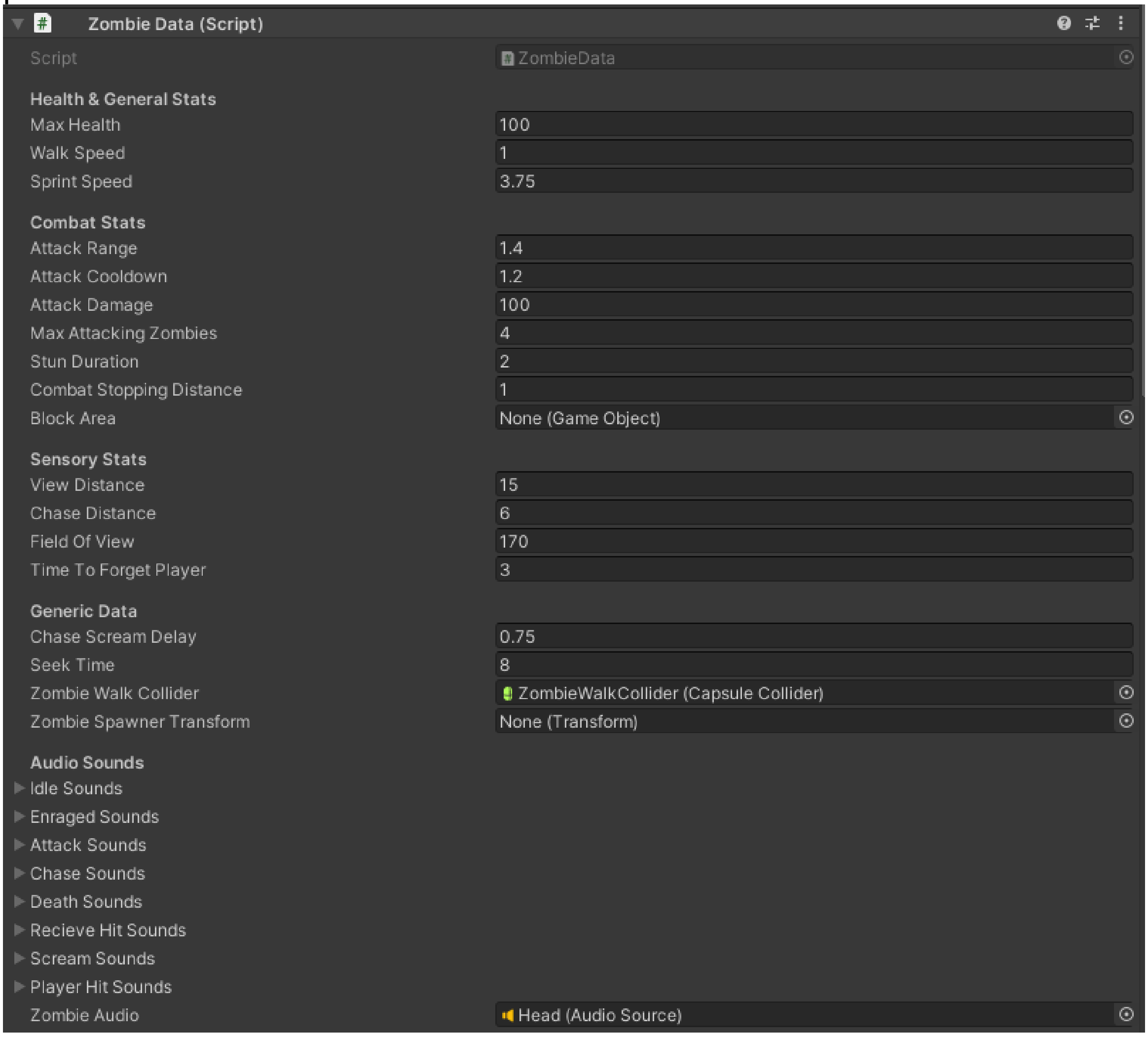
The state machine algorithm is supported by a sense and data system that houses all the information they use when executing actions.

The sensory system allows agents to both hear and see the player based on set circumstances. Figure x shows a green sphere, that works as a trigger collider if the player enters this area the ai will begin checking if they are within their field of view and unobstructed. It also checks if the agent can hear the player, based on whether they are walking or sprinting, crouch is unheard allowing for a stealth element. Seeing the player will cause them to transition straight into a combat state.



When working with Ai in the past I had issues with them forgetting information on an enemy as soon as it went out of sight, for this project I made sure to avoid that issue by creating a basic memory system that allows them to continue following the player for a short period even if they are out of view. This prevents them from seeming stupid, forgetting about the player as soon as they go around a corner.

The second side to designing and tweaking the AI is their data system. The easiest way to control how the zombies acted without having to edit their scripts each time was to use an inspector panel that exposed the main elements they used in decision making and combat. This allowed a more flexible and fluid design process as I could change factors with ease, it also made it easy when writing the state machine scripts to have access to everything in one place.

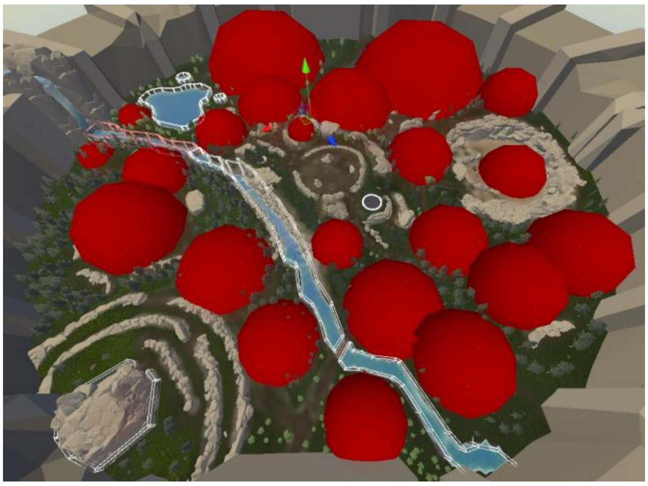


Zombie Data (Script)

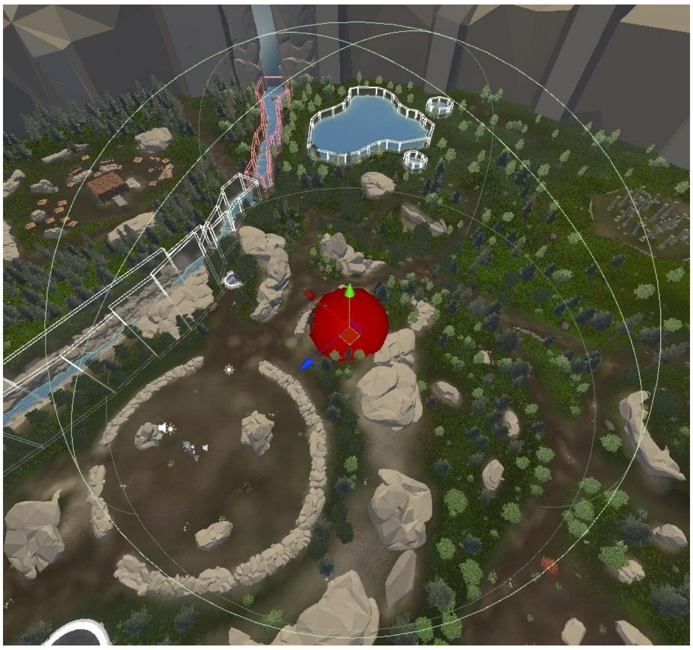
Script	ZombieData
Health & General Stats	
Max Health	100
Walk Speed	1
Sprint Speed	3.75
Combat Stats	
Attack Range	1.4
Attack Cooldown	1.2
Attack Damage	100
Max Attacking Zombies	4
Stun Duration	2
Combat Stopping Distance	1
Block Area	None (Game Object)
Sensory Stats	
View Distance	15
Chase Distance	6
Field Of View	170
Time To Forget Player	3
Generic Data	
Chase Scream Delay	0.75
Seek Time	8
Zombie Walk Collider	ZombieWalkCollider (Capsule Collider)
Zombie Spawner Transform	None (Transform)
Audio Sounds	
▶ Idle Sounds	
▶ Enraged Sounds	
▶ Attack Sounds	
▶ Chase Sounds	
▶ Death Sounds	
▶ Recieve Hit Sounds	
▶ Scream Sounds	
▶ Player Hit Sounds	
Zombie Audio	Head (Audio Source)

6.2.3 Zombie Spawning

To make sure that zombies are evenly distributed across the map, I chose to create several spawn locations that work throughout the game to replenish and spawn/despawn zombies as and when the player comes within their active area, they also ensure zombies do not wander too far away from their intended location.

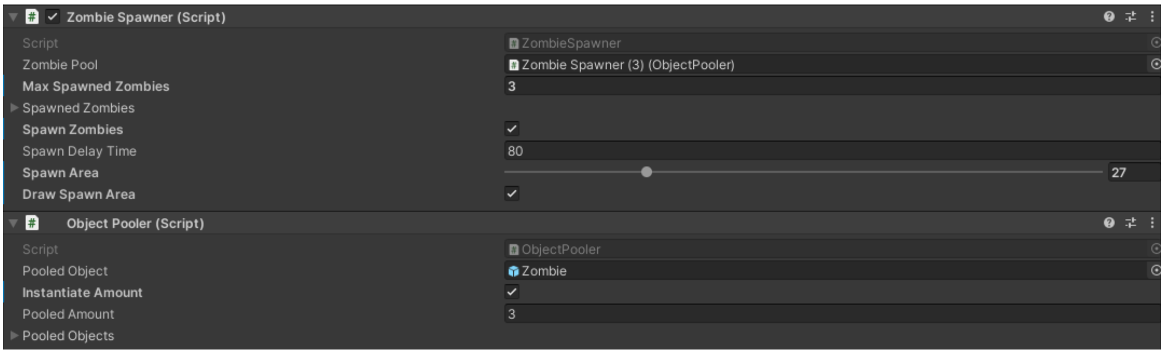


Zombies that are spawned within these areas are set to patrol within the designated spawn area. Seen individually as red spheres in figure x. They are triggered active or inactive based on whether the player has entered a sphere trigger, indicated by the green lines seen in figure x. The main benefit of this system is to increase performance as only certain zombies are active at any given time, so less Ai is running.



Zombies that die are permanently dead on the map though have all scripts running on them disabled. The spawner then begins a new spawn process to replenish zombies over time once the player has left the area, being cancelled if they return.

To control and edit each spawner with ease I set up a system in the inspector seen below.



Zombie Spawner (Script)	
Script	ZombieSpawner
Zombie Pool	Zombie Spawner (3) (ObjectPooler)
Max Spawned Zombies	3
Spawned Zombies	
Spawn Zombies	✓
Spawn Delay Time	80
Spawn Area	27
Draw Spawn Area	✓
Object Pooler (Script)	
Script	ObjectPooler
Pooled Object	Zombie
Instantiate Amount	✓
Pooled Amount	3
Pooled Objects	

6.2.4 Future Work

Future work might see zombies spawn by burrowing out from the earth. This would allow for zombies to spawn whilst players are in the area and will give a logical explanation for where they are coming from. It might also add to their fear factor.

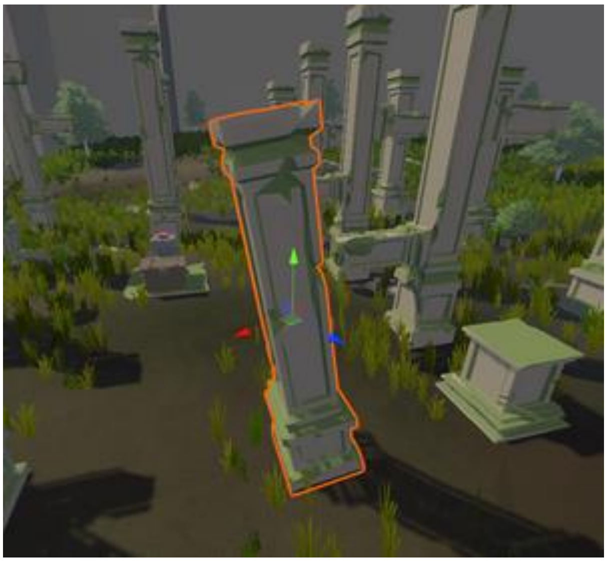
6.3 Puzzles & Challenges

Making up the main objectives of the game these features took the largest period of development to perfect, their design goals were to promote the use of the game's mechanics whilst also giving players a challenging and entertaining environment to use them in.

6.3.1 Parkour Challenge

This area forces the use of movement mechanics forcing the player to climb their way up to the top of a central pillar, via the surroundings. On approach, the pillar is raised from ground level up to a height that is only accessible via

The design of the challenge promotes the maximum use of the space at hand, with pillars being used at multiple levels. During white box testing, I found that due to the length of the route taken it was very frustrating to fall later in the challenge, as the player would have to restart from scratch. To reduce any frustration, I added a checkpoint in the form of a falling pillar, see figure x. This allows any player who falls after this point to begin from a halfway point so that they may not need to restart from scratch again.



For the animation of the two moving pillars in the challenge, I found a great tool called "Itween" (Berkebile, 2021), it allows for real-time object animation whilst being quicker and easier to set up than Unity's animation system. I used this tool throughout the rest of my project to help maximise productivity and time.

6.3.2 Rune Puzzle

Working as a more logical challenge, this puzzle forces the player to align certain runes together, then lock them into a list displayed on a vertical wall. Upon completing the puzzle, a pillar with a chest will be activated and the player can loot its contents.

The rune circles can be activated by standing on a pressure plate at the centre of the puzzle. This allows the player to then move individual circles using the Q and E keys. They must then use a second pressure plate located below the vertical wall to lock in any runes they have activated.

Initially, this pressure plate and the one used to activate the rune circles was very simplistic, they had no markings and did not stand apart from their surroundings. The feedback I received in playtesting showed that players were failing to identify them for what they were and therefore struggled to complete the puzzle.

To deal with this issue I re-designed them using brighter more eye-catching bases, I made sure that the design for the two plates was identical, unlike before, so that players could easily recognise one, had they found the other. The runes surrounding now also light up to show the plate has been activated. The change in design can be seen in figures x and x

Upon lining up a set of runes with an outer symbol they will activate it. It can then be locked into the main rune list by standing on the pressure plate beneath it, seen in the image above. If needed, already activated runes can be overwritten to complete different selections. Whilst beginning simple, the challenge becomes more difficult as the player completes the list, where the player must then line up two runes at once.

The biggest challenge I faced with this puzzle was determining how to track each of the three rune circles. I decided that creating an array that would shift with each rotation of a circle was the best approach, to find out how to do this I researched methods for array shifting on code forums, finding a few methods on stack overflow (2021, B).

After looking at these methods, I created my own, see Appendix B. When each run circle turns there is a check to see if all three arrays have the same rune in a specific position if so, they are lit, if an outer run is also aligned it is activated and can then be locked in if needed.

6.3.3 Ranged Challenge

This area is designed to both give the player an entertaining objective, whilst also helping to train them on the throwing mechanics of the game. The challenge has three stages, starting from left to right, with a new type of target introduced in each stage. There is a vat of target balls that can be collected so that the player need not craft endless items to throw, these items get destroyed on use and do-little damage to zombies so discourage the player from trying to overuse them.

Secondly, to stop a player from activating a challenge, and then running outside to hit targets from point-blank range, the hut's door becomes locked until they have either completed the challenge or reset it by pressing the button a second time. Completing all challenges, will like the other major challenges, unlock a chest filled with higher tier loot.

Finally, to keep with the theme of the arena already being attempted by previous competitors, a dead body is placed at the window of the first challenge, and two of its targets have already been knocked down.

6.3.4 Minor Challenges

Partway through development of the map, I realised that the original design would just funnel players from one object to the other without much need or reason to travel anywhere else. To that end, I decided to scrap one major challenge as their design was taking up too much time due to a misjudgement in pre-production.

Instead, I created minor challenges that scout the corners of the map and advertised them to the player through using vistas that encourage exploration such as swamps, caves, and clearings. This has helped make the map feel much more alive and entertaining to play in.

These smaller challenges include areas with increased zombies protecting a loot area and a quid pro quo puzzle with chest and key.

6.3.5 Future development

One thing I would like to possibly add is a narrator that gives the players hints and tips during or on approach to new puzzle and challenge areas. Testing showed that players need further guidance on what their objectives are, and in some cases, how they work. Whilst some of this guidance could be given in the tutorial, tips on what they must do would be better given at the time. An alternative to this may be to give narrated information on the objectives in the tutorial and give real-time information through tooltips or environmental guidance like signs or posters.

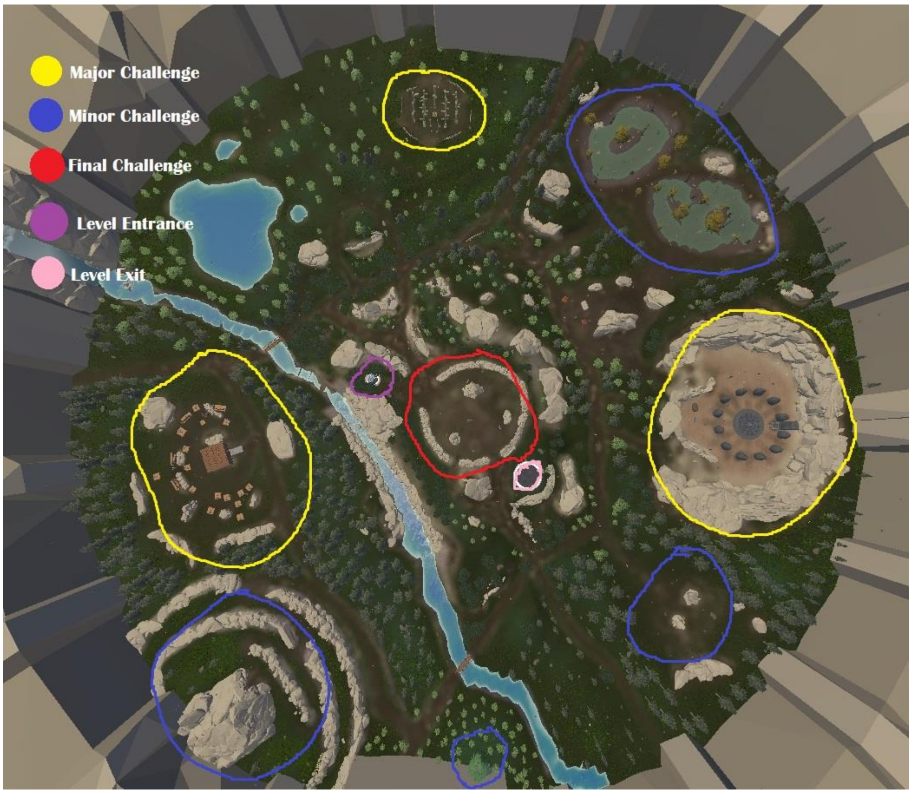
6.4 Environment

6.4.1 Development

The difficulty with creating the environment for this game was that it had to both look and feel like the game's theme, whilst also working mechanically with open-world, sandbox type gameplay.

To this end I originally planned for a map where the player could spawn from four corners, allowing the player to enter a place closest to where they might have died. This would save them having to cross the map if their current objective were on the opposite side of any single spawn point. Secondly, I had planned to exclusively have four challenges that culminated into a final one however that has changed in the final version.

I decided to change the four spawn points into a single entrance tube, outlined in purple. I placed this near the centre of the map so that the player could then travel in any direction without needing to cross the whole map to return to their point of death. This also made much more sense narratively, with the player entering from the tutorial scene. whilst also being easier to implement in the long run.



Overall, I finished the environment ahead of schedule. To some degree, this made up for some time I had gone over the puzzles.

6.4.2 Future development

Due to the success of the environment in its current form, I do not feel at this time that there are any extensions needed, I will review this in future after any other changes have been made.

6.5 Tutorial

Throughout development, I began to realise that there were several systems and features that players would not be able to just pick up and play. For this reason, I decided to create a tutorial scene. By end of the tutorial, the objective is to have players ready to be injected into the arena knowing all the basic features they will need to survive.

The tutorial contains the following rooms:

6.5.1 Starting Corridor:

Where the player is first introduced to the game, the guide makes remarks about how they are one of many competitors competing in a competition and then prompts them to make their way down the corridor, producing a tooltip teaching the movement mechanics.

6.5.2 Melee Room:

Where the player is taught about picking up items and equipping them. Upon entering the room, they are asked to pick up a spear on the floor and then use it to attack the target dummy. Completing this they are then taught how to block.

6.5.3 Ranged Room:

In this room the player learns about the ability to throw items, they are given a tooltip explaining how to switch to ranged mode and then must pick up some target balls to knock two targets down. Regardless of whether the player hits the targets or not they can still proceed as the point is for them to learn how to access and use the mechanic.

6.5.4 Item Room:

The final room of the tutorial is the item room, where the player learns about the basic resources and have an opportunity to interact with the two lootable inventory types they will interact with within the map. They can if they choose, take the resources in the two inventories, and craft some items, however, everything is taken away from them at the end of the tutorial.

6.5.5 Ending Corridor:

This last corridor preps the player for the entrance into the main scene, they can walk towards the entrance pipe as the screen fades out and their transition begins.

6.5.6 Future Developments:

As it stands, I am not completely happy with the tutorial, I think it could do better to explain to the player what their objectives in the level are, and a more in-depth account of the narrative surrounding the game. I feel currently this is a sore point for the game so will aim to rectify it in future development.

6.6 Boss

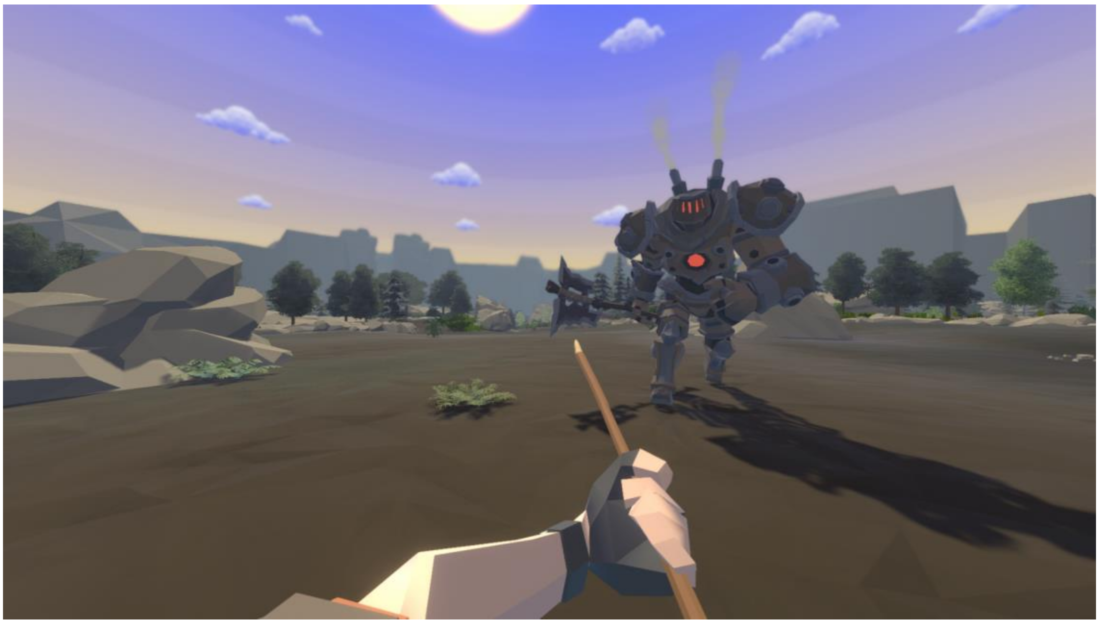
6.6.1 Concept and Reasoning

The final challenge for the game, bringing together all the skills and features used throughout the game. It is the product of several ideas I had been through and serves what I see as the best solution.

Originally, I had planned a final challenge that would see elements of the three major challenges repeated, however, I realised that this would allow the player to complete the game without needing to craft items, which would have made developing the whole system quite pointless. To solve this solution, I instead decided to implement a boss enemy that could believably only be damaged by metal items, due to its armour. This would force the player to craft up and make use of the system, making it a far more prevalent part of the game. It is hinted at in the Golems cutscene with the rhyme "sticks and stones can't break his bones.".

6.6.2 Implementation

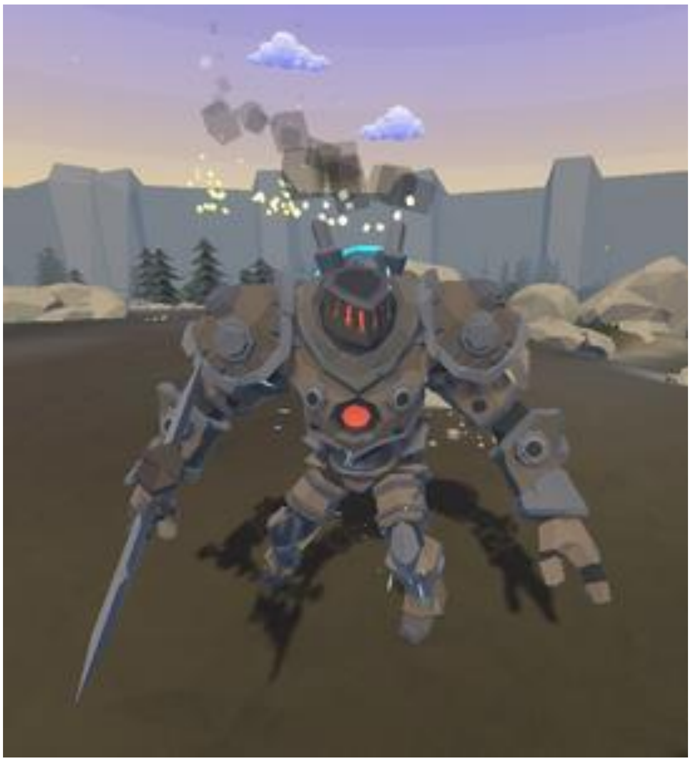
The boss runs on the same state machine as the zombies; however, his combat state is greatly developed in comparison.



One issue I wanted to make sure would be dealt with are instances where the player might outrun or get into a position that the Golem cannot. To solve this, I added a fireball attack, which the Golem would choose to use when the player was passed a certain distance. To ensure that this did not cause the golem to always sit at the range, a cooldown is also added for this magic attack, to give the boss time to close any distance if the player moves.

Finally, I did not want to use a health bar as I felt it would detract from its feel and not match with the overall style of the game. Considering this I decided a better approach was to use spatial elements to show the player how well they are doing. My implementation for this was to use Unity’s particle system and adjust the visuals of the Golem as he takes damage at different percentages of max health.

Health:
100%: Gray Smoke
75%: Black Smoke
50%: Black Smoke + Sparks
25%: Black Smoke + Sparks + Flames
0%: Dead + Sparks



This has proven to work very well, with playtesters commenting on how clear and informative it is to them during the fight.

6.6.3 Future Developments:

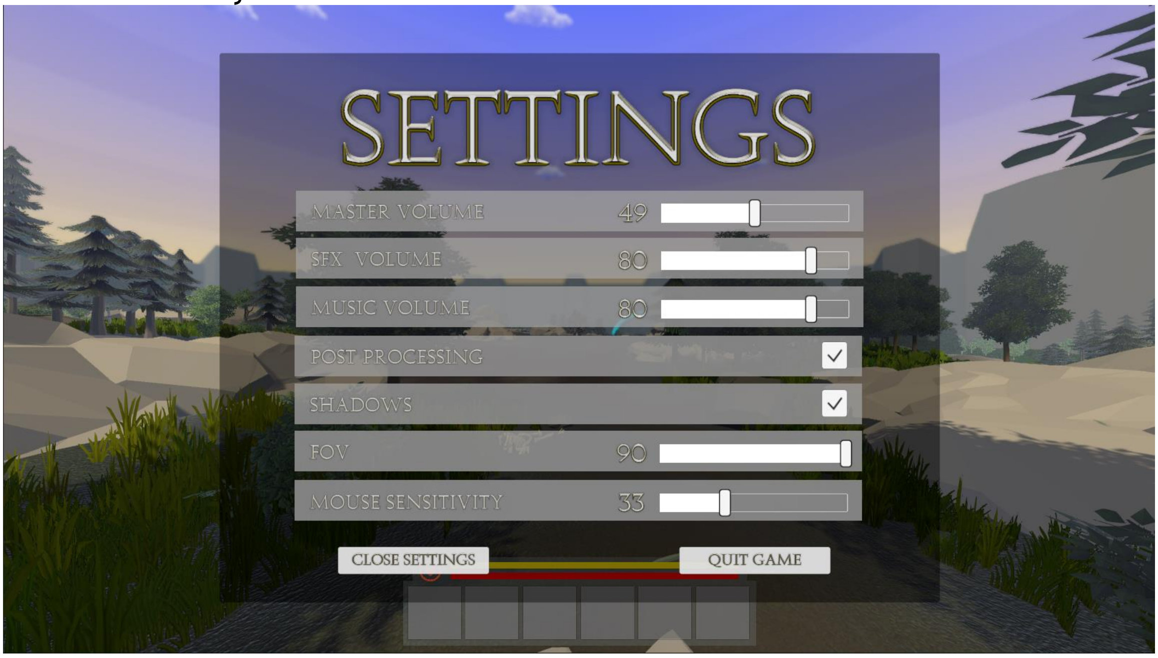
Whilst I am happy with the core implementation of the boss at current, I would like to look at increasing his move set and perhaps improving his state machine to allow him to make more complex decisions. He has a weakness in struggling to hit the player when they get in close under his axe, so I would like to implement an Aoe type attack that he can use to stop the player from abusing this too often.

Secondly, I may adjust his height as it is difficult to find animation for combat against a smaller opponent.

6.7 UI and Menus

6.7.1 Core Concepts and Design Choices

Whilst not a focus of my project, I still wanted to allow players to customise features that they would commonly expect to be able to change in conventional games. Primarily this meant adding an FOV slider and more importantly, a Mouse Sensitivity slider.



SETTINGS

MASTER VOLUME 49
SFX VOLUME 80
MUSIC VOLUME 80
POST PROCESSING ☑
SHADOWS ☑
FOV 90
MOUSE SENSITIVITY 33

Pausing the game was easy as I simply had to pause the games time scale, which worked as effectively stopping time completely.

6.7.2 Future Developments

In future, I would like to improve the artistic quality of UI elements as they are quite basic at current. As Ui is not a priority of mine compared to other areas, I felt that getting the menu functional was most important at this stage in development. The same could be said for other areas of UI in the game such as the inventory system and main menu.

6.8 Major Unity Bug

I have recently encountered a very frustrating issue within Unity's build system that prevents me from creating a built version of the game to send to testers. It has something to do with Unity's iL2CPP compiler that converts C# code into C++ which then becomes Binary. After Looking at several forum posts detailing other people's issues with this (Unity Forum, 2021 B, Unity Forum, 2021 C).

I attempted several fixes such as:

- Downloading the most recent Microsoft SDKs
- Reinstalling Unity Packages for Visual Studio
- Installing extra packages for Visual Studio
- Upgrading the unity projects editor version
- installing new unity editor versions and reinstalling current ones

However, none of these fixes worked and I had this same problem. By this point, I had spent 2 and a half days on the issue with no progress and I was lost. Fortunately for me, I found that unity has a second compiler called Mono, which is slightly older however let me build. After checking that there were no massive issues with Mono (Unity Forum, 2021 D). Which there did not seem to be. So, I continued work.

6.9 Playtesting

6.9.1 Internal Testing

For testing, I used a method called Adhoc testing, which is defined as a testing method with no pre-planning or documentation, where "a tester is free to test the system/application to execute any scenarios, what he thinks." (Chhabra, M. N. (2012). Using this method, I started multiple playthroughs, listing down all the Issues that occurred in each playthrough that needed fixing. This worked very well for me and I felt very productive in this period. The list that I had created by the end of testing can be seen in Appendix B.

6.9.2 Public Testing

Upon finishing internal testing, I released the game for others to play and provide feedback. To log feedback, I used a google form that allowed testers to both make suggestions and discuss any bug that they had encountered, the responses to which, can be found in a spreadsheet document attached to this submission. This period gave me some great insights into issues regarding balancing, bugs and areas of polish that were still needed. I will cover the main of which in the next section.

6.10 Balancing and Additional Addons

6.10.1 Resource Balancing

After publishing to some private testers, the biggest response was that resources and crafting recipes were wholly unbalanced. To rectify this, I lowered the number of resources needed to craft certain items and increased Tool harvests amounts making each tool effectively level up, (a stone hatchet now harvests the same as a metal hatchet) and so on.

Secondly, players found it too difficult to gather resources such as twine and metal fragments as they were only lootable from bodies and chests whilst also being dropped when destroying resource deposits and trees. I solved this issue by adding the two resources to trees and stone deposits respectively as a secondary resource, whereby the player has a chance to gain them each time they harvest. This greatly improved resource gathering and allowed players to easily craft basic items as was intended.

This process of balancing will continue in future developments however I am happy with its current state.

6.10.2 Inventory and Crafting Polish

After further testing two big issues have been raised, firstly many players struggled to identify when they were successfully harvesting, as well as how much they were getting. To fix this I added text that would flash on the corner of the screen each time you successfully harvest a resource, it tells the player how much of the resource they harvested and the type of resource they got. It also shows any extra resources they might have gotten such as twine or metal fragments.

The second issue was that players had no idea what certain items were or what their intended use was and asked for a tooltip that would show some basic information about the selected item. Implementing this was difficult at first because I wanted a system that would show the tooltip near the mouse, and check if it was correctly displayed on the screen and not off it (moving it back on screen if it was). However, with the time I have left and the time this approach was taking I decided to switch to a singular position that would show item details instead.

6.11 Evaluations, Reflections and Future Development

6.11.1 Evaluation - A critical discussion of the project:

This project aimed to show my skills in systemic and gameplay design, through displaying a range of completed features within a fully playable level with narrative.

6.11.1.1 Systemic and Mechanical Design

Regarding player systems, I have achieved the core functionality planned in the original project proposal. I planned for a set tier system of both weapons and tools that would allow for progression throughout the game whilst also having noticeable differences. At the time I had taken inspiration from Rust (Facepunch, 2021) to decide on how items would scale and interact with the environment. This proved useful as I did not have to balance the system from scratch or look too heavily into what constituted a successful item tier system.

The biggest drawback with player systems is a lack of polish, I did not have time to create the full suite of animations that I had planned, mainly movement animations when holding items. I also failed to implement the bow and pistol though the latter was only a stretch goal. Overall, I am pleased with the functionality and completion of this area. Testing played a big part in ironing out bugs whilst also providing me with great feedback on additional features that were missing such as item descriptions and resource gathering feedback.

Secondly, the puzzles created for the game have gone far better than I envisioned. Originally, I had planned for more simplistic challenges that could be completed quickly with ease. However, I soon found in testing that my designs were taking far longer for players to complete and were found more difficult than expected. I consider them a strong point for the project overall. On the other hand, puzzle design came at a cost in development. Where I had planned in the Gantt chart for puzzle implementation to take just over a week, the process took closer to two and a half, with me creating the environment in-between as a break from my struggles. The reason for this is down to a lack of planning in pre-production. My project plan briefly mentions a plan for puzzles to reflect gameplay mechanics, however, no designs or plans were made in this period to scrutinize how long or developed these puzzles would be, resulting in me entirely under scoping their size within the project. This resulted in my overshooting my window to work on the dynamic events which I had planned, though as mentioned in the plan, these features were a stretch goal based on me not running into issues during development, so ended up being used to finish off the puzzle implementation.

Enemies developed for this project were intended primarily as a platform to promote to use of player systems and force the player to interact with them. In this regard, I feel they have been a success as the player simply cannot survive without crafting weapons and medicines to defend themselves. Development of these features also went smooth, with the core creation of the zombie Ai and final boss being done ahead of schedule, only needing minor polish and bug fixing later. The development of spawning zombies across the map was also a great addition to the project as it allowed for play to be continued rather than the arena becoming somewhat of wasteland on the players first pass.

Overall, I would say that this aim has been met, though I believe there is always room to polish gameplay systems and features, the majority of core goals have been met, and due to testing have been polished and fixed in areas as well. The allowance of stretch goals and slack periods in the plan meant the over scope of puzzles was managed and dealt with rather than causing the project to sink.

6.11.1.2 Developed Environment

The environment was very challenging when I began developing it. With little experience creating open-world sandbox levels I had to think hard on how the player would interact with all areas. I again used Rust (FacePunch, 2021) to establish how the placement of trees, rocks and landscape can affect gameplay, often using it as a reference. It is also where I got the idea for biomes to help separate the map and assist navigation.

As mentioned in 6.4 It was during the map-making process that I decided more atomic and low-level challenges were needed throughout the map to promote exploration, as, without them, the game would just be about running to a few key locations with no reason to explore elsewhere. This decision has most definitely paid off as when observing testers play through the game it was apparent that they used these areas to help gear up to fight the final boss.

Finally, since there were no negative comments about the environment from testers, I feel that this was a success in its entirety.

6.11.1.3 Fully Playable Narrative Level

I felt the best way to showcase the systems that I had designed was by showcasing them at a complete level with a narrative, as it would show my ability to see a full-scale project through from start to finish.

To this end, I implemented a tutorial scene that doubles as part of the narrative, as well as cutscenes and dialogue that help guide the player and inform them on what their objectives are. Unlike the other two aims of this project, I do not feel as though I entirely met this one. This became most apparent to me in testing, where some mentioned that they either had no idea or were unsure of what their objectives were. I feel first, as though this is because I did not implement enough dialogue into the game scene, regarding the completion of tasks or commentary on the player's success. More importantly, however, I feel as though this is due to a failure in the tutorial. Whilst it does a good job of explaining the based mechanics and showing players how systems work, there is an obvious gap in how it builds the story and informs the players of their actual objectives.

I believe that this aim has only been partially met on this project as it does contain a fully playable level from start to finish. However, I believe that more work needs to be done on narrative elements to make them feel better connected to this experience.

Whilst this important I feel it is the lesser of the three aims on this project so will look to resolve it in future work rather than immediately as there are more important things I wish to improve first. Such as the player systems polish.

6.11.2 Reflections - A post-mortem discussion of the project as a whole:

6.11.2.1 What went right?

From my evaluation, I feel that the biggest positives from this project were the competition of core mechanics and features as well as an environment to use them within. I am very pleased with the fact that the game has so many different areas to interact with and learn.

Secondly, I believe my method of Adhoc testing was very efficient on this project, being a single developer, I did not see the need for detailed defect reports as they would take time to write, and I would be the only one viewing them. Also, many testers are not experienced with QA so would struggle to provide accurate or usable information such as this in detail, so having them simply describe any bug in the feedback form was easier.

Finally, the use of my risk assessment allowed me to quickly put recovery plans into action, saving time and preventing greater fallout on the project due to my issue with puzzles and time spent for animations. Specifically, this refers to risks 1 and 11.

6.11.2.2 What went wrong?

The biggest issue with this project was scope. Whilst it was most prevalent with puzzle design, I also feel as though it was an issue across the board. It has resulted in me having to work for longer than planned, just to implement the core features. The result is that I have had less time to polish than planned whilst also missing out on stretch goals I had planned. The reason that this happened was because of a lack of planning in the project proposal, where I had failed to develop certain features of the game and had left them as ideas. The ramifications of this have meant the project is not as polished as I would have liked. There are still areas that I would like to improve and some features that I would like to add.

Secondly, I should have investigated how I was going to source first-person animations for the player character in more depth. The solution I came to in development, using forward kinematics, is the reason that I had to revert to my risk assessment and get by with what was functional. Had I spent more time researching other avenues to complete the task, I may have found a more efficient means to completing and task.

The final issue I had with this project was finding the time to keep a detailed log of what I was working on, regarding the keeping of a detailed logbook with regular entries. I believe this was caused by the rate I was working to try and keep to my scope, often finishing late and not having the motivation to then write an account of my day's work. My solution for this was to use my Github commit history, in which I made sure to make regular comments on the feature I was working on, as well as noting any major design decision I had made. This at least has allowed me to accurately view what was completed each week on the project, though has made the post-mortem process slightly more difficult as I have had less comment to go on when looking back to work, I had done.

6.11.2.3 What was learned and What could have been done differently?

Firstly, the biggest lesson for me was the importance of prior planning, not only to fulfil a proper project plan but also to better gauge the feasibility of a project. My failure to do this has resulted in this project paying a price. Had I done this differently it would have forced me to scope the project down, resulting in me having more time to complete what I had planned.

Secondly, I have come to respect the importance of time management and making sure you have a watertight solution for when everything in your project will be completed by. Whilst I had to deviate in places from my original Gantt chart, by whole I followed it closely, and have found that it very much put things into perspective and increased my productivity due to seeing how much time certain elements were costing. I feel that had I not done this I could have become complacent, feeling as though I had plenty of them to complete certain features when I did not.

Finally, I have had the importance of testing reinforced to me as without making time for it I would have released a project with a multitude of bugs and issues. Furthermore, my project would also have needed many areas of extra polish. This has enforced my planning on future projects to include ample time for this process.

6.11.3 Future Development - A discussion of the potential continuation of the project:

6.11.3.1 Continuation

The main reason I see for future development is to improve the game for the graduate show in June. This would initially involve adding polish to the features currently in place within the game, most of which have been discussed previously in this report. The benefit I see from doing this would be to present a more professional and complete project to potential employers.

Past this, I do not see a continuation of development on this project. The reason for this is that I have plans with other students to begin working as a studio, my focus would then pass to whatever project that team begins. Furthermore, although I worked the project into a playable level with narrative, it intends to be more of a showcase/tech demo than a fully developed game, this is also the reason I do not intend to publish on steam.

6.11.3.2 Other platforms

As for porting this game for console or mobile, I do not see any feasibility. Firstly, the game is too costly to run on a phone, and not only do I lack the skills to optimise the game to this level, but I am not sure it would be possible.

As for Console, I lack access to the development kits needed to port the games, I do not see this changing for the foreseeable future so will not pursue it.

7 Reference list / Bibliography

1. Adobe, 2021. *Mixamo*. [online] Mixamo.com. Available at: <https://www.mixamo.com/#/> [Accessed 24 April 2021].

2. Answers.unity.com. 2021. *The best place for answers about Unity - Unity Answers*. [online] Available at: <https://answers.unity.com/index.html> [Accessed 19 April 2021].

3. Ark, 2021. *ARK News*. [online] ARK - Official Community Forums. Available at: <https://survivetheark.com/> [Accessed 13 February 2021].

4. Berkebile, B., 2021. *iTween for Unity by Bob Berkebile (pixelplacement)*. [online] Pixelplacement.com. Available at: <http://www.pixelplacement.com/itween/index.php> [Accessed 25 April 2021].

5. Bohemia Interactive, 2021. *DayZ | Bohemia Interactive*. [online] Bohemia Interactive. Available at: <https://www.bohemia.net/games/dayz> [Accessed 13 February 2021].

6. Chhabra, M. N. (2012) 'Introduction To Adhoc Testing', *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, 1(7). Available at: www.ijstr.org (Accessed: 25 April 2021).

7. FacePunch, 2021. *Rust | Facepunch*. [online] Rust.facepunch.com. Available at: <https://rust.facepunch.com/> [Accessed 13 February 2021].

8. *FIRST PERSON MOVEMENT in Unity - FPS Controller*. 2019. [video] Directed by Brackeys. https://youtu.be/_QajrabyTJc: Youtube.

9. Gill, S., 2014. Visual Finite State Machine AI Systems. [online] Gamasutra.com. Available at: [Accessed 7 December 2020].

10. *How To Make An Inventory Like Rust In Unity Tutorial! - Part 1 - Setup and Adding Items*. 2019. [video] Directed by NikNax. https://youtu.be/Cf-3SPL1IMI: Youtube.

11. Millington, I. and Funge, J. (2009) *Artificial Intelligence For Games (Second Edition)*. 2nd edn, *Morgan KaufMann*. 2nd edn. Morgan KaufMann. doi: 10.1017/s0263574700004070.

12. Stack Overflow, 2021 B. *C# quickest way to shift array*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/2381245/c-sharp-quickest-way-to-shift-array/2381397> [Accessed 25 April 2021].

13. Stack Overflow. 2021 A. *Stack Overflow - Where Developers Learn, Share, & Build Careers*. [online] Available at: <https://stackoverflow.com/> [Accessed 19 April 2021].

14. Sweetser, P. and Wiles, J. (2002) 'Current AI in Games: A review', *Australian Journal of Intelligent Information Processing Systems*, 8(1), pp. 24–42. Available at: http://cs.anu.edu.au/ojs/index.php/ajiips.

15. Unity Forum, 2021 A. *Checking for the name of a tree being chopped from a terrain*. [online] forum.unity. Available at: <https://forum.unity.com/threads/checking-for-the-name-of-a-tree-being-chopped-from-a-terrain.332839/> [Accessed 24 April 2021].

16. Unity Forum, 2021 B. *https://forum.unity.com/threads/exception-il2cpp-exe-did-not-run-properly.659323/*. [online] Unity Answers. Available at: <https://forum.unity.com/threads/exception-il2cpp-exe-did-not-run-properly.659323/> [Accessed 26 April 2021].

17. Unity Forum, 2021 C. *Build - IL2CPP errors - Please, how fix it?*. [online] Unity Answers. Available at: <https://forum.unity.com/threads/build-il2cpp-errors-please-how-fix-it.492556/> [Accessed 26 April 2021].

18. Unity Forum, 2021 D. *Mono vs IL2CPP*. [online] Unity Forum. Available at: <https://forum.unity.com/threads/mono-vs-il2cpp.533416/> [Accessed 26 April 2021].

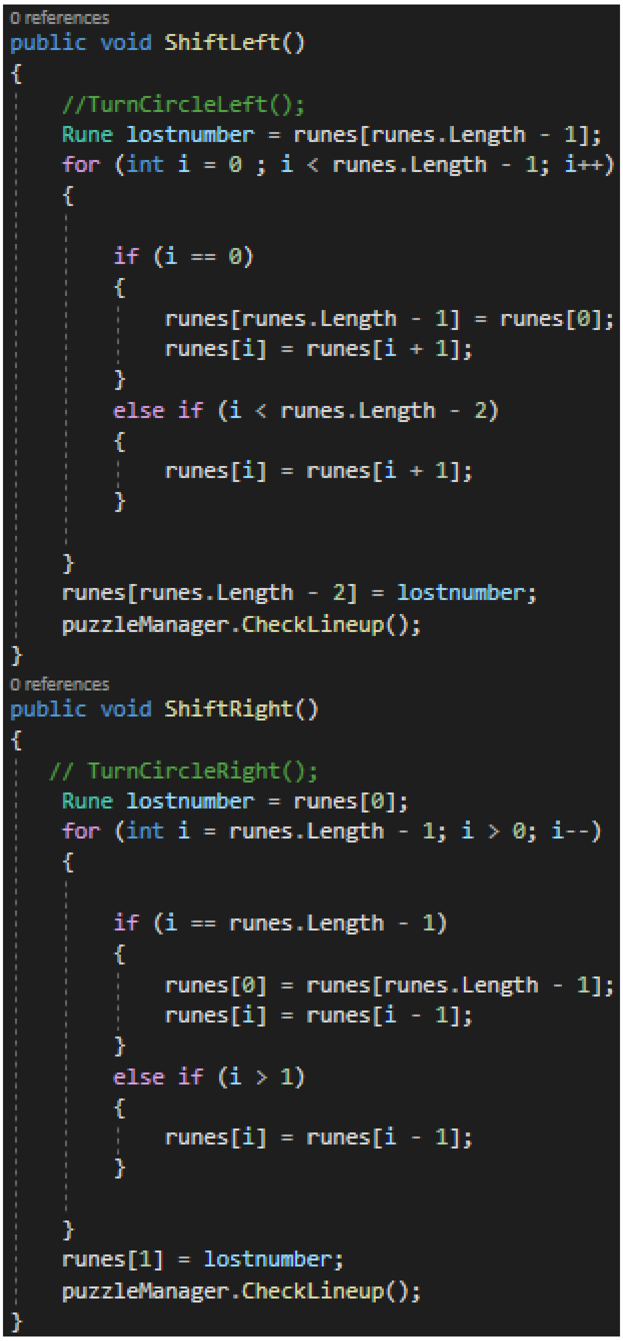
8 Appendices

8.1 Appendix A: Work Week Breakdown

Week:	Features Implemented
Week 1 (Feb 1st)	Player Movement and First-person controller (decided to add a body to the character and separate the arms for easier animation).
Week 2 (Feb 8th)	Work on another unit
Week 3 (Feb 15th)	Began adding weapon animations. Inventory systems
Week 4 (Feb 22nd)	Started crafting system. Added remaining animations Throwing.
Week 5 (Mar 1st)	Zombie Ai Blood Splatters
Week 6 (Mar 8th)	Added loot able Inventories. Added first iteration of forcefield. Added Parkour Puzzle Started Rune Puzzles Started Terrain and Environment
Week 7 (Mar 15th)	Finished first draft of the map. Finished Rune Puzzle Created Base Zombie Spawners
Week 8 (Mar 22nd)	Added Range Puzzle Added Medical Items Added Respawn System
Week 9 (Mar 29th)	Started Work on the tutorial scene. Added more crafting recipes. Added Game Scene Spawn and transition from tutorial. Started Final Boss
Week 10 (Apr 5th)	Added Settings and pause menu.

	Added Cutscene for shields closing. Added Zombie Respawning Fixed general puzzle issues added some environmental storytelling. Changed Game font and ui images.
Week 11 (Apr 12th)	Major unity bug First playtest Adhoc Bug testing
Week 12(Apr 19th)	Final Release Balancing and minor patching
Week 12 (Apr 26th)	
Week 13 (May 3rd)	

8.2 Appendix B: Function created to shift array elements in the rune puzzle.



```
0 references
public void ShiftLeft()
{
    //TurnCircleLeft();
    Rune lostnumber = runes[runes.Length - 1];
    for (int i = 0 ; i < runes.Length - 1; i++)
    {

        if (i == 0)
        {
            runes[runes.Length - 1] = runes[0];
            runes[i] = runes[i + 1];
        }
        else if (i < runes.Length - 2)
        {
            runes[i] = runes[i + 1];
        }

    }
    runes[runes.Length - 2] = lostnumber;
    puzzleManager.CheckLineup();
}
0 references
public void ShiftRight()
{
   // TurnCircleRight();
    Rune lostnumber = runes[0];
    for (int i = runes.Length - 1; i > 0; i--)
    {

        if (i == runes.Length - 1)
        {
            runes[0] = runes[runes.Length - 1];
            runes[i] = runes[i - 1];
        }
        else if (i > 1)
        {
            runes[i] = runes[i - 1];
        }

    }
    runes[1] = lostnumber;
    puzzleManager.CheckLineup();
}
```

8.3 Appendix C: Testing Results List

Test ID	Results
1	1. Exiting Settings does not hide mouse. 2. Footsteps too quiet. 3. Blocking sounds. 4. Twine Item Id missing. 5. Can punch door triggers. 6. Sound is very quiet after changing scenes. 7. Sharp rock breaks instantly on tree hit (durability check). 8. Navmesh need baking. 9. Zombies bug when crouching. 10. Zombie footsteps not working. 11. Outline. 12. Larger amount of lootable bodies. 13. Item balancing. 14. Painkiller still plays use sound when you cannot use it.
2	1. Add Waterfall Smoke. 2. Master Vol. 3. Moving item from equipped slot does not unequip it. 4. Outlines still bugged. 5. Tool tip 0 in chat at ranged tutorial tooltip. 6. Stones near entrance are floating.
3	1. sharp rock attack cancels. 2. crafting images. 3. zombie footsteps.

4	1. Trees not dying correctly. 2. Crafting requirement arrows do not disappear. 3. Sensitivity randomly setting itself. 4. Thrown items full durability refresh. 5. Axe melee. 6. drag and drop no item slot error. 7. tree drop items all spawned.
5	1. Metal axes equip causes animation loop. 2. Check chests lootable. 3. target ball pickup outline. 4. target ball not destroyed on zombie hit.
6	1. Build Problems 2. Tutorial Lights dim 3. Cursor at tutorial start. 4. Meshs Mouse sense 5. Last room remove "perhaps" from line 1. 6. Outlines. 7. Drag and drop quick move fix. 8. Golem nav agent broken. 9. Shield impact 10. Zombies not reacting to player.
7	1. Swap from spear to spear. 2. Target tutorial blocker. 3. Sub behind menu screen. 4. Ambient wind.

8	1. Update Crafting Requirements. 2. Whilst Spamming Throw Balls you can punch in between.