CloneFall 2: Game Design Document

1. Introduction

1.1. Our Plan

For this assignment, we plan on cloning Respawn Entertainment's 2016 first-person shooter known as Titanfall 2 (Respawn Entertainment, 2016). Our goals are to replicate the basic movement systems, before moving on to building a multiplayer arena in which the players can interact with each other in a simple deathmatch, inspired by one of the game's modes, known as 'Pilots vs Pilots'. Furthermore, we are going to try to replicate UI elements from the game along with the map design language of Titanfall 2 to create a richer environment for the mechanics to be built in and tested.

1.2. What is Titanfall 2?

Titanfall 2 is a first-person shooter video game that was developed by Respawn Entertainment and published by Electronic Arts. It was released in October 2016 for Windows, PlayStation 4, and Xbox One. The game is notably built upon a heavily modified version of the Source engine, which influences the game's feel.

"Titanfall 2" is the sequel to "Titanfall" (2014) and expands on the original game's concepts. The game combines fast-paced action with the strategic elements of controlling large robots, known as Titans. These Titans have various abilities that can be deployed in combat, including powerful weaponry and defensive systems. Mostly, the game focuses on quick, high-lethality firefights in an arena environment, where mastery of game mechanics (especially movement) is vital to victory.

On the multiplayer front, "Titanfall 2" offers a variety of modes and customization options, including the ability to customise both the pilot and the Titan. Gameplay centres around the dynamic between Pilots and Titans, with Pilots being agile and capable of parkour, while Titans are less mobile but significantly more powerful.

2. Interrogation of genres

2.1. Main Genre Analysis

Titanfall 2 is first and foremost a First Person shooter. The game is almost exclusively played from the perspective of the player character, and moving the mouse moves the player's view. A great deal of the game is built around handling guns from this first-person perspective, with mastery over these weapons being a key focus as players compete with each other to show superior skill by scoring more points across various game modes. Something to note about these modes is that with one exception, players are rewarded the most for targeting other players, with the other objectives either serving as something to do between firefights or a way for

worse players to catch up. This puts the emphasis on competing with others directly and forcing intense clashes where the player with the most skill comes out on top, which are the lifeblood of the game.

2.2. Subgenre Analysis

When it comes to Titanfall 2, it is seen to incorporate various subgenres, here are a few key subgenres seen within the game, they are:

- Parkour: Titanfall 2 incorporates parkour elements extensively into its gameplay. As pilots, players can run along walls, double jump, mantling, power-sliding and perform various acrobatic manoeuvres. This gives the game a high mobility that drastically influences the game's pace and how players navigate the environment. These mechanics are an iteration upon Titanfall 1's unique style of highly movement-centric arena-based first-person shooting that integrates high levels of lethality and movement potential with a fast respawn time, creating a fast-paced shooting environment. It also drastically raises the skill ceiling with the added dimensions of mobility, although this is at the cost of raising the skill floor as well.
- Multiplayer Online Battle Arena (MOBA): Although Titanfall 2 is primarily an FPS, some aspects of its multiplayer modes resemble the MOBA sub-genre. For example, in the Attrition mode, along with fighting against other players, both teams also face off against AI-controlled grunts and other NPC enemies, similar to MOBA games where players combat both player-controlled and AI-controlled characters. Titanfall 2 also features a separate set of mechanics in its main sandbox in the form of large mechs known as Titans. These mechs provide a fun change of pace mid-round, using mechanics closer to those of a MOBA or a Hero shooter than an arena shooter while players control them, but because of how isolated they are from the core gameplay systems mechanically -only being connected by some specific mechanics built for interaction- they have been deemed to be out of the scope of this project.

3. Inspiration

The main reason we chose this game is that it should provide something that is challenging to replicate but still possible within our current skill bracket. The game's complex movement systems are ripe for study, and building a multiplayer environment will provide a chance to learn vital skills that are highly sought after in the broader games development space. The game also offers somewhat of a safety net for our development, since we should be unable to implement multiplayer, pivoting to providing a skill trial for a single player in the form of the gauntlet (a single-player mini-game accessed within Titanfall 2's campaign) is also a viable option since it would use similar mechanics without the complexity of implementing them in a multiplayer environment.

4. Hypothesis / Design Goals

We intend to recreate a small portion of the FPS, Titanfall 2. Specifically, we intend to:

- Clone the unique movement mechanics of Titanfall 2.
- Create a simplified version of the gunplay (only a few weapons for prototyping and presentation purposes) since the game is foremost, a shooter game.
- Attempt to facilitate some form of multiplayer - to gain some experience with networking within games.
- Recreate as many UI elements within the game as possible.

5. The Team - Clone-Group-One-4

Team Member	Roles
Bruce Tonkin	● The multiplayer component of the game ● Gunplay within the game ● Work on Technical Documents and Game Documents
Erin Harper 2445245	● UI components within the game ● Presentation ● Work on Technical Documents and Game Documents ● Game Pitch Document
Jean-Francois Retief	● Character controller ● Level Design ● Work on Technical Documents and Game Documents

6. Title of Game

Seeing that our game is a clone of Titanfall 2 we have decided to name it ‘Clonefall 2’

7. Game Overview

Our game is planned to feature a small multiplayer environment, built using a similar design language to Titanfall’s more close-quarters levels. This distinction is important because the maps in Titanfall have to cater to both regular-sized humans and giant robots, but since we do not have titans in our list of planned features due to how

complex they would be to implement, we have to instead shift our focus to purely being on the core gameplay experience.

To create this core experience, we have found three mechanics and two other elements we need to focus on. Those mechanics are:

- Titanfall 2's iconic movement. We plan on building the main mechanics of sprinting, sliding, jumping and wall-running, before tuning them to match Titanfall 2's unique Source Engine-based character controller as closely as we can.
- Gunplay. Titanfall 2's gunplay is relatively simple since the complexity comes from the movement instead of any kind of highly complex weapon handling, but recreating the interesting combination of Precision Aiming and Random Bullet Deviation that the game uses should still prove an interesting challenge to undertake.
- Multiplayer. Our goal is to build a basic net code that allows somewhere between 4 and 6 players to enter the same environment and compete against each other in tense skill-based gunfights. We chose this because Titanfall 2 is, at its core, an arena shooter, in which the player's main goal in every game mode is somehow related to the act of expressing superior skill by defeating other players in combat.

The two secondary requirements that we have to focus on are as follows:

- Building a Level. Titanfall 2 is built on strong level design, and while we don't aspire to those same levels, creating a similarly interesting environment for players to learn and adapt to is vital for showing off the depth of the movement we are hoping to create.
- User Interface. Titanfall 2 has a simple user interface, but it's vital for conveying information about the player's condition and the conditions around them. We plan on replicating the ammo counter, health indicating border (lovingly referred to as "Raspberry jam") and the minimap.

8. Level Metrics

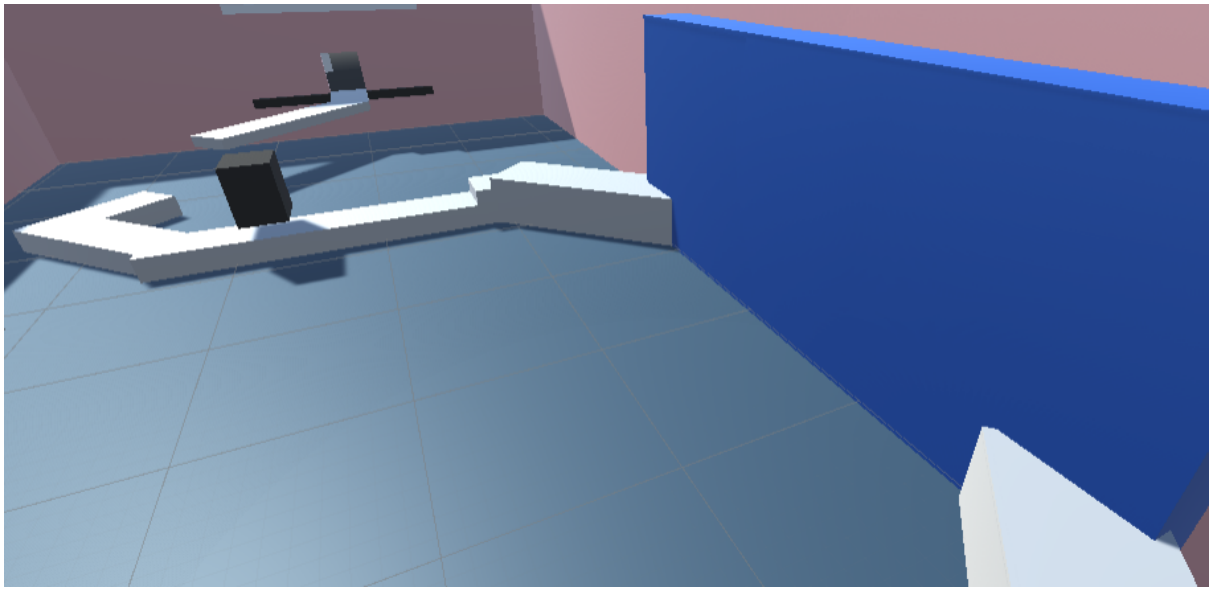

Figure 1: Tutorial section screenshot

There is a section within the game that gives room for the player to get used to the controls and character controller. It allows the player to experiment with wall-running, powerslide, etc.

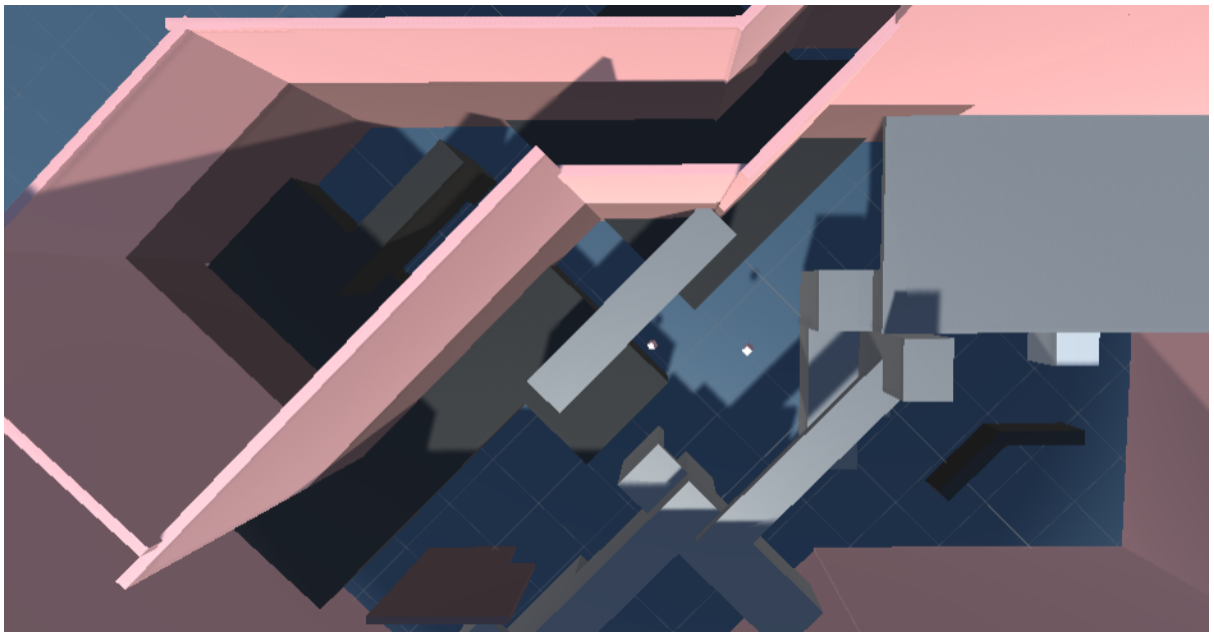

Figure 2: Main Multiplayer Arena screenshot

The main arena is split into two parts:

- An open area filled with geometry that allows for interesting wall-running chains
- A semi-closed-off hallway to allow for interesting combat scenarios.

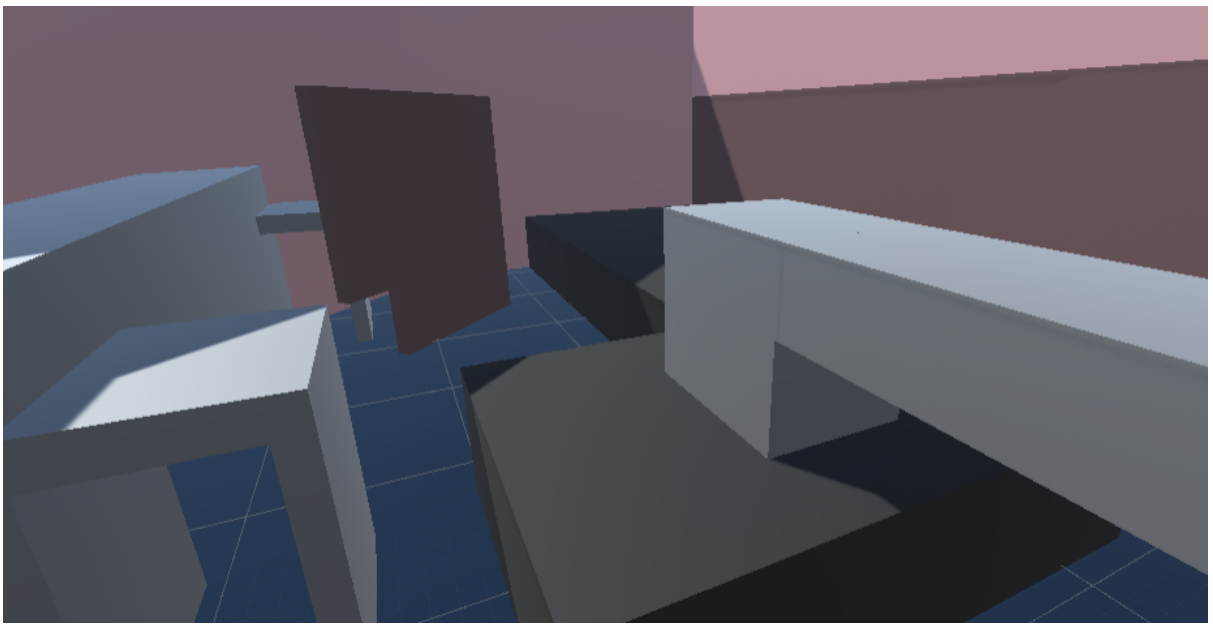

Figure 3: Screenshot displaying wall-running opportunities

The geometry in the main arena allows the player to jump from wall to wall, keeping their momentum to avoid being shot.

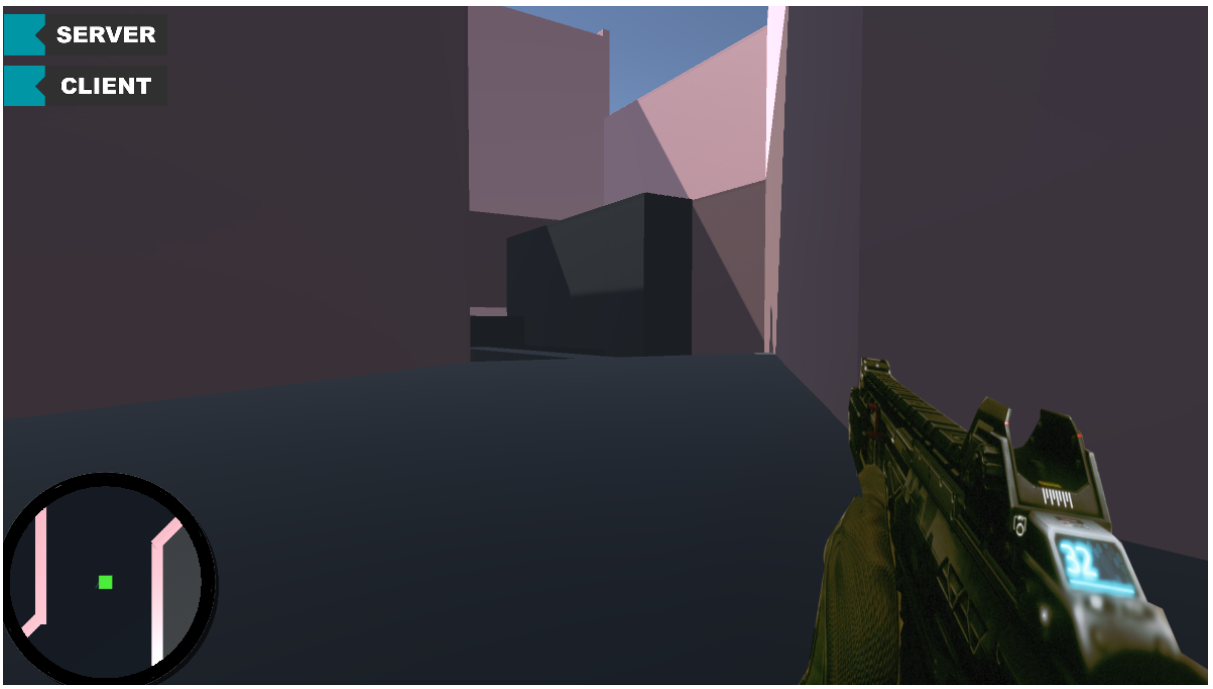

Figure 4: Gameplay screenshot (notice minimap)

The minimap gives the player more information about their surrounding area, as well as the other players' locations if they are close by. The player is indicated by the colour green, so that it stands out from the environment and that the player is able to understand who is on their side vs the enemies.

Diving further into the level Metrics of the game, we will be touching on how the player will make progress/fail within the game.

How the player will succeed within the game:

- Avoid getting shot at by other players
- Manoeuvring their way through the obstacle course
- Shooting other players/enemies within the game

How the player will fail within the game:

- Being killed within the game.

The primary measure of which player is doing best is number of deaths, with the player with the least depths being the player who is out performing everyone else.

9. Difficulty and Design Philosophy

In the beginning, we aspired to work on multiplayer networking, knowing full well how difficult it would be. We wanted to use this opportunity (where the game itself does NOT count the majority of the marks) to investigate and get experience with things such as networking. The desire to try areas of game design and development that we personally haven't had much experience in formed the base of many of our initial decisions. Some of these decisions include: deciding to make a first-person shooter with random-bullet-variation and gun recoil, deciding to focus on the fast-paced "pilot" movement of Titanfall 2 (with wall-running), rather than the mech/Titan gameplay, and small additions like a minimap. All of these were challenges to the members of our team that we decided to take on because we didn't have the answers before we started to develop the game.

10. Design Notes and Processes

- Not every aspect of our game is one-to-one with Titanfall 2. The reasons for some of the differences are:
    - Initially limited scope for prototyping purposes
    - Time constraints
    - Networking constraints
    - Internal decisions, when something in our game seems enjoyable to us, even if it was not exactly the same as Titanfall 2.
    - Assets are too advanced to recreate in a short space of time
- This game is mostly mechanically similar to Titanfall 2, although one deliberate deviation was our reload mechanics. We implemented a tactical reload -where the player reloads faster when they have an empty magazine- to make up for the inability to have the extremely polished animation checkpoints that make reloading in Titanfall 2 an active decision instead of a formality.
- For additional notes on specific features, please refer to the Technical Document (Harper et al., 2023)

11. Reflections

10.1. Bruce Tonkin's Reflection

Multiplayer proved to be both easier and harder than I could have expected. Initially, setting up the basics was really easy thanks to using a premade architecture, but as time went on I was reminded time and time again why I'm so stubborn about using my own code: it's hard to work with things you don't fully understand. Throughout the project, there were increasingly more and more situations where something that I or someone else would change would have massive unintended ripple effects, which took a great deal of time to fix and caused a lot of frustration. That said, I think these challenges did make me stronger as a game designer and forced me to spend more time with the game, leading to an increased appreciation for why Titanfall 2 is so good in the first place. So, multiplayer is never to be undertaken lightly, unless you're planning on doing something incredibly simple. Still, I'm glad that I did this. Getting started is almost always the hardest part when learning a new skill, so having this basic background means that no matter what I am getting something out of this project.

Titanfall 2 is a very interesting game to study because of how simple its base mechanics are and how densely they become layered once you get into the game. The complex interactions between movement and gunplay create an environment where high skill is strongly rewarded, albeit this does cost the game overall. Titanfall 2 has a notoriously high skill floor, not helped by a raft of un-tutorialized advanced techniques that are basically required knowledge to be competitive, and it hurt the game at launch, likely even being the reason that the franchise died as quickly as it did, despite the rave reviews. Titanfall 2 aimed to resolve these issues by adding titans as a catch-up mechanic, but we chose to exclude those since we were aiming for a minimum viable product, not a fully functioning game that we could take to market.

The way that gunplay in Titanfall 2 interacts with movement is arguably as important as the movement itself. I spent some time puzzling out exactly why it felt the way it did, and my conclusion was that the handling of the weapon operates differently based on whether the player is aiming or hip firing. While hip firing, the game uses a simple randomised spray similar to other games built upon the Source Engine. However, when aiming, the game switches to a more modern recoil-based system. The game also mainly uses hitscan for weapons, which made my life a lot easier since networking physical bullets would likely have been nigh impossible. To replicate this unique gunplay, I built two separate systems for aiming, one that uses random bloom and another that replicates recoil by physically moving the camera, to recreate this dichotomy. I felt the second was especially important to replicate since

my personal bias for recoil mechanics heavily leans towards simulated solutions instead of gamified ones because I feel those add a lot more skill than the latter. I also made extensive usage of the Titanfall wiki (*Titanfall Wiki*, no date) while tuning the gun, since it provided enough exact data to get the gun feeling almost exactly like its in-game counterpart.

The last thing we sought to imitate from Titanfall 2 is the level design. Titanfall 2's level design is built around facilitating movement, but also the Titans. This created complications at first, but then we realised that the live fire maps, which are small indoor arenas for a titan-free mode, would serve as a better reference for a titan-free movement space. Once we had our reference, the goal became replicating the way that the maps in Titanfall 2 flow. Movement in the maps comes in two forms from my experience: the smooth movement through the intended routes, which flow together beautifully and direct the pace of play, and the rougher unintended routes, which is happens when the player needs to get somewhere in an unintended way and don't have a tool available to bypass the space. Titanfall 2's bigger maps struggle from far too much of the latter due to the need to design around Titans (Colony is probably the most afflicted by this), and we needed to avoid it as much as possible since it's the least enjoyable part of the game. Fortunately, we were able to make our lives easier by referencing mainly Drydock and Live Fire maps, which are the best at maximising the amount of time spent on intended routes.

I want to briefly reflect upon the work we did as a group. I think this was one of the areas where I realised I still have a way to go because I feel as though we were only able to get through this assignment through raw competency as opposed to efficient teamwork. Our communication was good thanks to using discord and whatsapp, but we struggled with working in parallel due to our combined inexperience with GitHub and -more damagingly- our lack of good coding practices when it comes to making our code easy to understand and work with for teammates. I was the main person guilty of this since my focus was more on building than documentation and it meant that there were issues trying to integrate my code into other functions (especially because of the complicated networking involved with what I was doing).

So, to conclude, the two things I learnt the most about in this were networking and gunplay. My networking still has a way to go, but I'm actually pretty satisfied with my gunplay. The system I built is robust and responsive, and I'd actually really like to dig into gunplay more in the future since I really enjoyed the little bit I did here. I also learnt through this that my ability to work in a team still has a long way to go, which is a tough pill to swallow but also a chance to do better in future.

10.2. Erin Harper's Reflection

To reflect upon the creation of the group assignment, my areas of focus were the UI components of the game, such as the main menu, minimap, raspberry screen, font,

etc. With this, I started by creating a demo scene on my own to work on the mindmap along with the Raspberry screen before finally integrating it into the game, see attached images below.

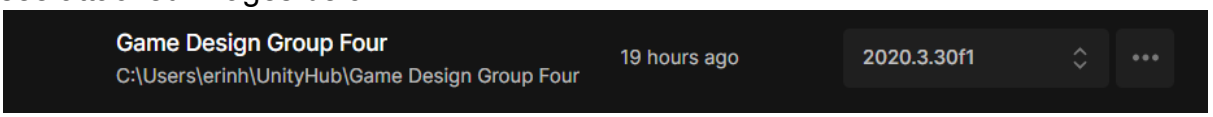

Figure 5: represents what I saved for the test project on my side

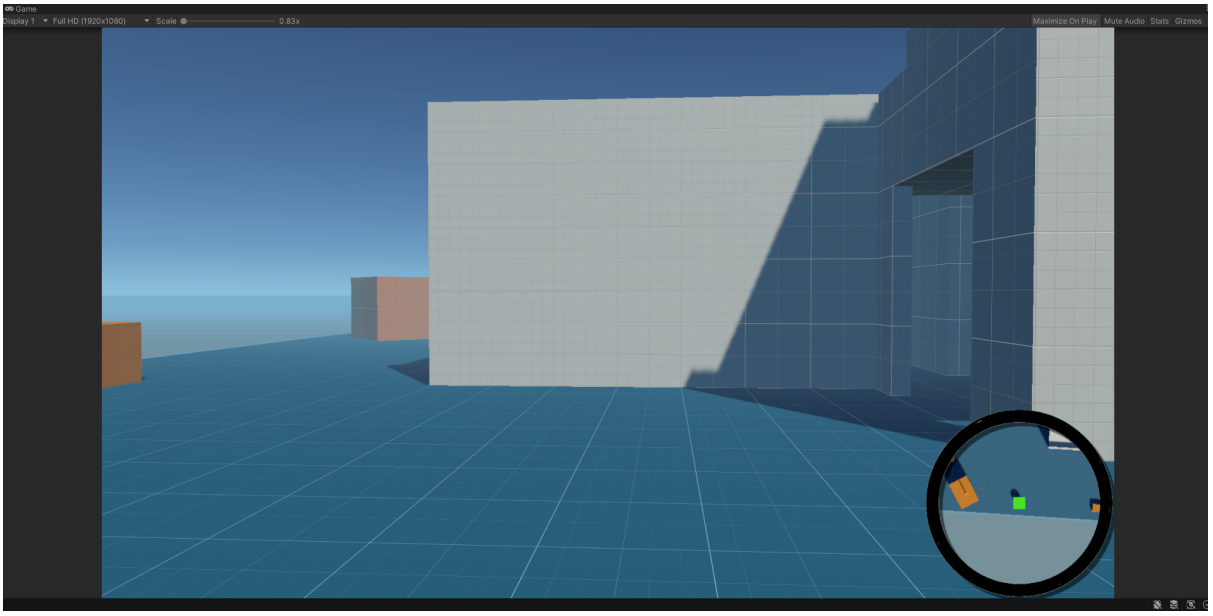

Figure 6: represents the demo of my minimap- which as you can see differs from the final one

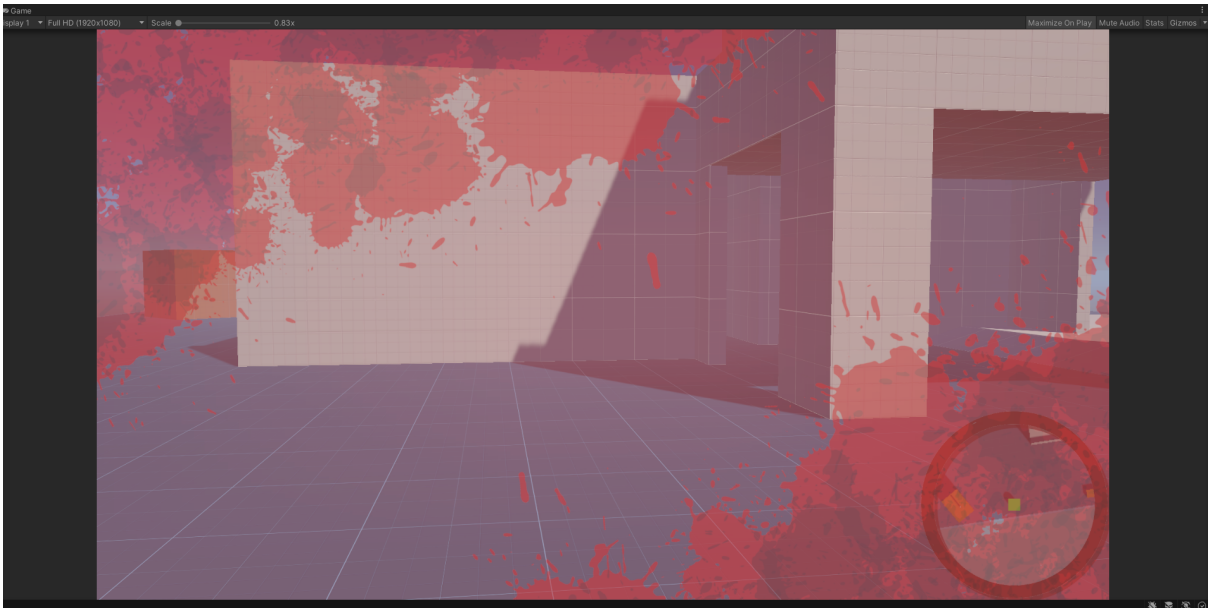

Figure 7: represents the Raspberry screen that will pop up every time the player receives damage

From these three images displayed above, I found the process of creating them quite tricky at times, for my initial thoughts was that they were going to be simple however, that was not the case, for the cameras needed to be positioned correctly, with enough information displaying so that players are able to see both views. At first, the cameras were not working simultaneously, but with enough trial and error, I managed

to get it to work, allowing me to integrate it into the final game. However, the problem came with integrating the Raspberry screen into the final gameplay due to the different canvas created wherein the user is to engage upon entering the gameplay, it clashed and did not allow the player to enter the game mode. Hopefully, before the time comes to submit this error can be worked upon.

When it came to creating the main menu of my game, I really enjoyed adding that feature to make it look like the actual game, I use adobe photoshop to edit and tweak a few things in the images - such as removing the wording that was displayed on the initial image as I could not find one appropriate on the web.

Overall what I can say from working as a group, especially to try and clone a game such as Titanfall 2, has shown me how important it is to work together, share ideas, and help each other and that nothing is impossible as this was my first time creating a game such as Titanfall 2, especially 3D as I am more used to creating games in 2D it allowed me to step out of my comfort zone which I am grateful for. I am really happy with the end result of the game, I think my team members and myself did a great job.

10.3. Jean-Francois Retief's Reflection

My areas of design and development were the movement system and the level design. I started with a checklist of goals for movement: basic FPS movement, wall-running, double jump, wall jumping, power sliding, mantling and grappling. Then I looked online to see other implementations, and this taught me a valuable lesson: do not underestimate the "simple" mechanics if those mechanics are one of the main focuses of the game.

I expected to see raycasting and some physics in online tutorials, but I was surprised to see the complex maths needed to make mechanics such as wall-running feel good to use. The only tutorials that were remotely acceptable for our project, had at the very least cross-products to get the wall-tangent vector and some other physics calculations that, even as an engineering student, I still don't fully understand.

Overall, I enjoyed the experience of researching solutions for problems I'm having, and using those ideas and techniques for different problems. For example, the code snippets I used to implement wall-running and wall jumping, I also used to inform my own code for mantling, double jumping and power sliding.

In terms of level design, I really enjoyed the experience of this project especially. The movement system influenced my level, and my level influenced my movement system. Throughout the process of building the level, I constantly had ideas like: "I can add a wall in the centre here, to give players access to that higher area" or "I can add another pathway here where the players have to powerslide inorder to get to this area faster".

The level had to showcase the movement possibilities, and still allow players to become more experienced, either by learning the nuances of the movement system itself by experimenting within the level or learning the layout and pathing of the level. The verticality of the level design of Titanfall 2 was also an interesting thing to clone. While designing any section, I had to take into account that the player could be entering or exiting these scenarios, in a multitude of ways. What follows is a few visual examples:

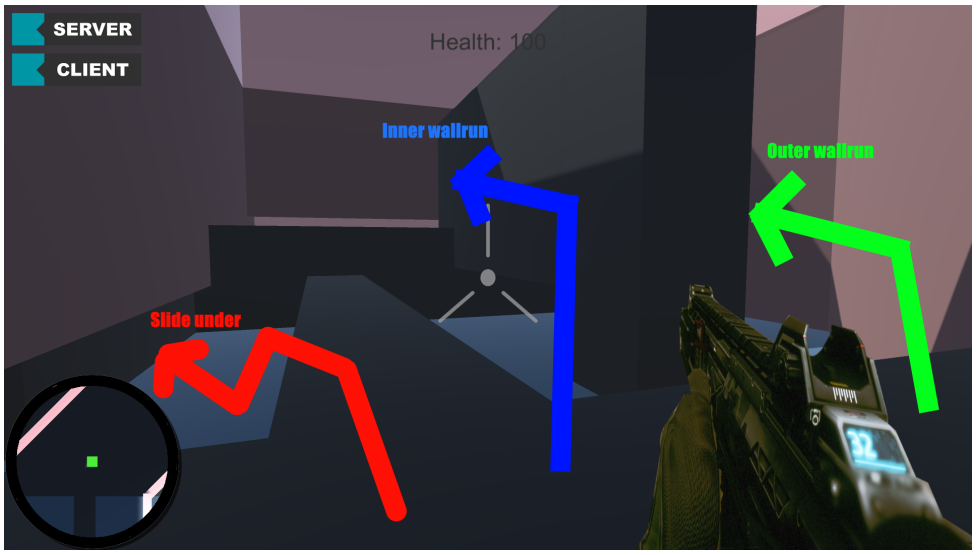

Figure 8: Scenario where the player can decide to wall run to the next section or take a shortcut by sliding underneath a wall

Figure 9: Another scenario where verticality is key to gaining the advantage in combat

Quite often, the high ground is preferred in a combat scenario, however, the two shortcuts between the hallway area and the open area are at the lowest level, which offers up an interesting trade-off which I found interesting to design for.

There are many additional trade-offs that I did not specifically design for, but instead are inherent elements of the genre of game or are emergent during gameplay. The player constantly makes meaningful decisions in any given scenario. For example:

while being chased by another player, one might try to get to the aforementioned high ground, but while parkouring away from your opponent, you leave yourself exposed on the side of a wall or in the middle of the air. These emergent scenarios are very similar to those found in Titanfall 2. So it can be concluded that, while cloning Titanfall, we got at least some of our design decisions right.

We chose Titanfall 2, to challenge ourselves and to learn about areas of game design we had little experience with, and this was the most difficult game I had to work on so far, and I'm glad we attempted. It was a fun challenge. I'm more proud of this game than I am of many of the other games I've made so far.

References

- [1] Harper, E., Tonkin, B., & Retief, J.-F. (2023). CloneFall 2: Technical Document.
- [2] Respawn Entertainment. (2016). Titanfall 2. Available online:
    - https://store.steampowered.com/app/1237970/Titanfall_2/
- [3] *Titanfall Wiki* (no date). Available at: https://titanfall.fandom.com/wiki/Titanfall_Wiki (Accessed: 4 August 2023).