3D Game Development

Exploring Accessibility in a Competitive Online Multiplayer Video Game

---

Department of Informatics
University of Sussex
Candidate Number 246535
Supervised by Dr Kingsley Sage
Year of Submission 2024

Statement of Originality

This report is submitted as part requirement for the degree of computer science at the University of Sussex. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged. I hereby give permission for a copy of this report to be loaned out to students in future years.

Candidate: 246535

Date: May 2024

Acknowledgements

I would like to thank my supervisor, Kingsley, for his incredible guidance and patience throughout the course of writing this dissertation, as well as the playtesters who gave their time to provide feedback. And of course my family, for their unconditional love and support.

Contents

List of Abbreviations

1 Introduction

1.1 Project Overview

In this project, I have developed a 3D competitive first-person online multiplayer game with a focus on accessibility. The report is an overview of the different resources, methodologies and evaluation practices used to facilitate the game's development. In Section 2, I have written about the inspiration for the game's concept and how looking at real-life sports as a source can help improve the accessibility of video games. In Section 3, there is a discussion on the use of software architecture and programming patterns. Section 4 establishes what accessibility means in games as well as how a game can be more accessible. After this, Section 5 highlights the various mechanics built into the Player Character (PC), whilst Section 6 focuses on the inclusion of explicit accessibility features through a settings menu. Next is Section 7, which describes the various Unity Gaming Services (UGS) used to develop the game's online functionality. Finally, Section 8 evaluates the results of User Test 1 and User Test 2, before Section 9 concludes the report and offers cases where future work could improve the final prototype.

1.2 Motivation

I have had a keen interest in video games since my early childhood, attributing such interest to studying for a computer science degree at university. I believe everyone should be able to access the fun and enriching experiences that come with playing games, whether that be alone, with friends and family, or even with strangers.
A stronger focus in User Experience (UX) has been seen in recent years as the number of people regularly playing video games continues to rise. In the online, 'live-service' sector, we are seeing such developments. Cross-platform technology allows people to connect regardless of the platform they are on. Aim assistant allows players who have trouble with hand-eye coordination to attack their opposition and win matches. Improved matchmaking systems with 'anti-smurfing'[1] practices ensure players can enjoy competing against opponents of a similar skill level. All these improvements make games accessible to a wider audience.
This project will give me the unique opportunity to understand the challenges associated with researching, designing, implementing and testing systems and mechanics in the game to improve accessibility, as well as the complexity of online functionality.

1.3 Findings and Achievements

In this project I implemented accessibility features into an online multiplayer game built from scratch, through my own design, and assistance from package tools used during the implementation stages of development. Through user testing, I discovered that the mechanics of the PC were intuitive and easy to use, and the ability to change settings within the game made the experience better for players. However, the online functionality could be polished in future versions of the game to remove bugs that disrupt the player's experience.

[1] smurfing is the act of a high-ranked player playing against low-ranked opponents

1.4 Professional Considerations

This project involves the development of computer software and anonymous user testing for evaluative purposes. As such, I have familiarised myself with the Chartered Institute for IT's Code of Conduct[2] and I can certify that I will abide by Sections 1 (Public Interest), 2 (Professional Competence and Integrity), 3 (Duty to Relevant Authority) and 4 (Duty to the Profession), which are relevant to any IT project.
I have also signed with my project supervisor the University of Sussex's User Testing Compliance Form, which can be found in Appendix D.

[2] https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf

2 Game - Overview

2.1 Overview

The final product I aim to create through this project is a game that has online functionality, as well as explicit (can be changed by the player) and implicit (built into the game by design) accessibility features. The player will interact with the game environment through a first-person perspective, using a minimalist control scheme to control the PC. As the title of this project implies, the game environment will be 3D. Furthermore, the player will be able to compete against other players over the Internet. In this section, I will provide an outline of the game's concept, followed by a requirements analysis, and then a discussion on the most appropriate game engine to achieve said requirements.

2.2 Game Description

To provide support in conceiving the game's design I chose to research what a competitive multiplayer game is, looking at similar titles and comparing them against each other to see which ones stand out the most in terms of accessibility.

2.2.1 Similar 3D Competitive Online Multiplayer Games

Trying to provide a general definition of competitive multiplayer games, which may also be known as 'eSports', is a challenging feat, as games under this sub-genre can be vastly different to each other. However, in the majority of instances, they will involve Player-versus-Player (PvP) combat, with teams competing against each other to try and achieve a target/objective. In *Rocket League*(Psyonix), that's scoring the most amount of goals by getting the ball into the opponent's net; in *League of Legends*(Riot Games), that's destroying the opponent's Nexus. I use Rocket League and League of Legends as examples here because, whilst both games are considered eSports, they differ significantly in terms of the mechanics and systems you will find.
Rocket League offers a key implicit accessibility feature in its clear inspiration from football. As football is widely regarded as the most popular sport in the world[7], the instinctive knowledge - to score more goals than your opponent - reduces the amount of learning required to play Rocket League for the vast majority of new players.
League of Legends, on the other hand, appears to take no direct inspiration from real-world examples. The game has an extensive tutorial section to teach the player its many components, which may be bothersome to those who are unable to grasp the game and thus have to replay subsections of the tutorial.

2.2.2 Game Concept

Inspired by most popular ball games/sports, the game that this report focuses on will involve two teams competing to score more goals than the other. As a deviation from conventional games, which have a global direction of gravity, the PC will have a 'gravity control' mechanic, thus making gravity a local system based on the player's orientation. More on this mechanic is explained in 5.7. A team may contain between 1 to 5 players.

Figure 1: Blue team attacking the red team's Nexus in *League of Legends*(2009). Source of image: `https://gigazine.net/gsc_news/en/20170929-league-of-legends/` [accessed 16/04/2024]

Figure 2: Screenshot of Rocket League gameplay. Top-centre of the screen's Heads-Up Display (HUD) shows the score

.

2.3 Requirements Analysis

I will now outline the functional and non-functional requirements of the project. Note that this list is not comprehensive and deviations from the game concept (2.2.2) may occur during later stages of development. Here I will focus on the requirements that will make the game more accessible as well as what is needed for online functionality.
In the non-functional requirements I have mentioned the use of the Xbox Accessibility Guidelines (XAG): more information on these guidelines can be found in 4.4.3.

2.3.1 Functional Requirements

These requirements are related to the systems within the game that the player will interact with.

Mandatory

- The game shall support the use of keyboard-only, keyboard and mouse, and gamepad input devices.
- Players shall be able to modify the game's visual and audio feedback.
- Players shall be able to change the input bindings for their chosen input device.
- The game shall provide a re-playable tutorial for learning the mechanics of the game.
- The game shall allow players to host and join online lobbies for competing against other players over the Internet.
- Players shall be able to create 'public' and 'private' lobbies.
- If a lobby is 'private', the game shall generate a password that other players use to join the lobby.
- A single online lobby in the game shall be able to connect a minimum of 2 players together.

Desirable

- The game should include a player agent with Artificial Intelligence (AI) for players to compete against in offline matches.
- The player should be able to customise the graphical quality of the game, to meet the processing capabilities of the player's computer.
- The game should provide an option for the player to save a preset of their customised settings to their computer.
- A single online lobby in the game should be able to connect a maximum of 10 players together.

2.3.2 Non-Functional Requirements

These requirements are related to the 'design' of the game's systems.

Mandatory

- The game shall meet the following XAG goals[8]:
  - *101 - Text display*
  - *102 - Contrast*
  - *103 - Additional channels for visual and audio cues*
  - *104 - Subtitles and captions*
  - *105 - Audio accessibility*
  - *112 - UI navigation*
  - *113 - UI focus handling*
  - *117 - Visual distractions and motion settings*
  - *118 - Photosensitivity*
- The game shall be compatible with macOS and Windows operating systems.
- The player shall interact with the game through a keyboard and mouse, or a gamepad compatible with their computer.

Desirable

- The game should meet the following XAG goals[8]:
  - *106 - Screen narration*
  - *110 - Haptic feedback*
  - *114 - UI context*
  - *115 - Error messages and destructive actions*
  - *116 - Time limits*
  - *119 - Speech-to-text/text-to-speech chat*
  - *120 - Communication experiences*
- The game should offer all functionality on computers with the Ubuntu operating system.

2.3.3 Domain Requirements

The requirements reflect the environment in which the game operates in.

Mandatory

- The game shall be accessible to players new and experienced with video games.

Desirable

- The game should provide controls that feel intuitive to the player.
- The game's interface should be visually appealing.

2.4 Game Engine

Before development can begin, I must choose the most appropriate game engine. To do this, I will use the following questions to analyse an engine's usefulness:

- Is support available for implementing accessibility features?
- Is support available for connecting players over the Internet (i.e. online functionality)?
- What are the licensing terms associated with using the engine?
- What are the engine's minimum hardware/software requirements?
- Which programming language(s) are used for writing classes/scripts?

With these points in mind, there are many game engines which could be considered to facilitate the game's production. In the interest of time, I have decided to shortlist my options to three currently available game engines. The engines in mind are: Unity[11], Unreal[12] and Godot[13].

2.4.1 Unity

The Unity game engine was released on 8th June 2005, and has since grown into one of the most popular game engines of modern times[14]. The web contains a vast collection of 1st and 3rd party advice available for free, from Unity-funded courses to YouTube tutorials. The *Practical Game Accessibility*[15] course offers over 12 hours of information related to implementing accessibility into games made with Unity. In addition, Unity provides a suite of online services as part of their UGS platform, including (but not limited to) Netcode for GameObjects (NGO), Lobby, Relay, and Vivox chat.
Unity offers a range of licensing agreements, mostly revolving around the amount of revenue generated from games made with the engine. A Personal plan is available for 'hobbyists' where any game generating less than 100,000USD in the last 12 months does not require the developers to pay a licensing fee to keep the game legally on sale[16]. With this said, the engine is considered proprietary - meaning developers are unable to freely modify the source code of the engine.
The system specifications for the Unity Editor include: Windows, macOS or Linux operating system, an X64 architecture CPU, and an appropriate graphics API in relation to the operating system in use[17]. Unity supports visual scripting, or the C# programming language for writing scripts.

2.4.2 Unreal

The Unreal game engine has innovated the industry by giving developers with limited resources the potential to produce AAA-quality titles. A limited amount of accessibility-related[18] and a substantial amount of online functionality-related[19] information is available on Unreal's website.
To use the Unreal Editor, a developer is recommended to have a Windows, macOS or Ubuntu (Linux) operating system, and competent hardware requirements[20](see reference for detailed system requirements).
Unreal provides a Standard License for projects that generate less than 1,000,000USD[21]

- no fee is attached to this license.
Unreal provides intuitive 'Blueprint' visual scripting technology[22] as well as support for writing scripts using the C++ programming language.

2.4.3 Godot

Godot is a 2014 game engine which has since found a base of supporters over the web as well as use by independent game studios[23]. Godot provides documentation for online functionality[24] and 3rd party accessibility plugins, although this information is limited. There is no official information on the system specifications, except that it can be ran on Windows, macOS and Linux[25].
Godot is free and open-source, meaning developers can customise the engine's source code to suit their needs. It also implies that there is no licensing fee attached to producing games with the engine[26].
Scripts can be written with the engine's native GDScript programming language[27], or with the C# programming language.

2.4.4 Summary and Decision

Out of these three game engines, my decision was to develop the game with Unity version 2022.3 LTS (Unity's latest Long-Term Service version).
Unity's investment in teaching accessibility was significantly appealing to me. Unity also has a dedicated asset store which provides resources made by 3rd party developers, and tight integration with the engine itself means that assets can be imported with ease. Furthermore, Unity's Netcode for GameObjects has excellent documentation as well as extensive community support through its forums and online tutorials.
I will admit there is some bias in this decision, as I have used Unity in the past to complete game-related projects. However, I believe this to be a benefit, as I can spend more time focused on the core sections of the project, rather than losing time getting onboard with the engine (as would be the case with Unreal or Godot).

3 Game - Software Architecture and Programming Patterns

3.1 Overview

In this section I describe the software architecture and programming patterns used to organise the writing (as well as the purpose of) the program's many classes. This includes ones specific to Unity, such as components and scriptable objects, and others universal to software development, such as the singleton pattern. Using the right combination of architecture and patterns in a software project is especially important when the project requires scalability - video games, for example, often receive frequent updates that add new items and game modes.
There is also a discussion on the use of the Data-Oriented Technology Stack (DOTS), which is a development stack provided by Unity. I state the usefulness of its features, despite choosing not to implement any into the game due to issues with refactoring.

3.2 Fundamentals

A GameObject in Unity is the colloquial term for an object within a scene, that represents the basis on which characters, props, and scenery are built in the development process[28]. A scene is a single environment within a game; each scene will contain a hierarchy which all GameObjects are placed into - GameObjects may even be placed underneath other GameObjects (see Figure 3), to create a nested parent-child GameObject.

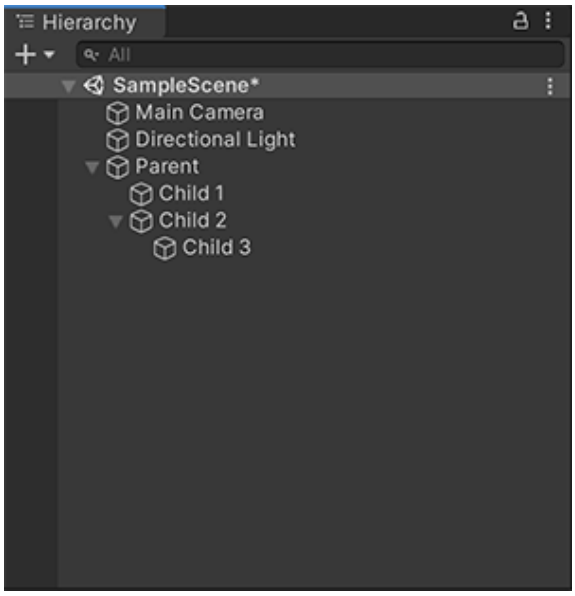


Figure 3: Hierarchy structure in a Unity scene. Source of image: `https://docs.unity3d.com/Manual/Hierarchy.html`

A GameObject can contain multiple components (see Figure 5). Developers may use components provided by Unity, such as Rigidbody and Mesh Renderer, or write their own components (referred to in Unity as scripts). Formally, this is known as Component-Based Architecture (CBA). The development of the PC(5) took advantage of this

system, so that the different mechanics of the PC could be added or removed to either enhance or restrict the player's ability to control the PC. Ideally, developers should keep components loosely coupled during implementation.

3.3 Scriptable Objects

A Scriptable Object (SO) is a data container that you can use to save large amounts of data, independent of class instances[29]. I found it especially useful for storing the value of settings in the settings menu (6) between scenes, as the public scope of an SO makes it easy to access, whilst the requirement to explicitly reference the SO prevents accidentally getting/setting the data stored.
A problem I ran into when using SOs are that the data is volatile, meaning the value that an SO holds will be set back to its default value the next time the player runs the game's executable. To solve this, I used Unity's built-in 'PlayerPrefs' class[31], which allows float, integer, and string variables to be stored and loaded from a JSON file on the player's computer.

3.4 Singletons

A singleton is a static class in Object-Oriented Programming (OOP) that has global scope, and has only one instance of it in a program.
There is much debate about the efficacy of singletons. The book *Game Programming Patterns* by Robert Nystrom details the overall negative impact that singletons have on the ability to produce clean code as the scale and subsequent complexity of a software project increases[1].
Nonetheless, the singleton pattern was a requirement as part of the functionality of Unity's NGO package. Furthermore, a singleton's simple implementation as 'managers' of systems in software, combined with their ease of access (from having a global scope) make them a desirable choice for prototype development. My plan was to limit the use of singletons where possible and opt for alternate approaches - such as the use of SOs for the storing of player settings.

3.5 Event System

To reduce coupling, OOP languages such as C# allow developers to invoke 'events' when a certain action/instruction is executed. The class that invokes an event is known as the 'publisher'. Other classes are then able to attach functionality to an event's invocation - these classes are called 'subscribers'. The publisher is unaware of which classes are subscribed to its events.
I have found the event system to be a useful tool for displaying information to the player as a result of back-end systems reaching a certain state. For example, when the timer in an online game expires, a 'GameOver' event is invoked - the GameOver user interface script, which is subscribed to this event, can then show the user interface to the player.

3.6 A Discussion on the Data-Oriented Technology Stack (DOTS)

I want to briefly discuss Unity's DOTS, as I learnt about this stack midway into the project's development, and wondered if it would have suitable application. As the name

suggests, this is a combination of technologies and packages that centre around a Data-Oriented Design (DOD) approach[30]. Unity suggest using DOTS for competitive online multiplayer games (among other types of games), as it allows for server-authoritative gameplay with client-side movement prediction (a feature I have discussed in 7). DOTS includes the Entity Component System (ECS), which is a modification of the default component system in Unity (the one I am using as described in 3.2).
I decided that I would take no further action with DOTS. Whilst the variety of technology available is impressive, it would require extensive refactoring of the systems I had already created, as well as an indefinite amount of time required to learn about DOD. However, I would certainly recommend for the reader to research DOTS and consider its usefulness when choosing what technology to use to facilitate the development of their own software projects in Unity.

4 Background - Accessibility

4.1 Overview

This section discusses the importance of accessibility in games, how we should approach difficulty, as well as how to design and implement games with accessibility in mind.

4.2 The Importance of Accessibility

Games that don't require physical activity have been a part of human culture for many generations. Chess dates back roughly 1500 years, having been invented in India in the 6th century CE[2]. These types of games (including video games) are especially appealing to individuals who aren't able to compete in intense sports such as football and gymnastics, as well as those who simply don't enjoy the fatigue and injury commonly associated with playing sports. Specifically, video games offer many benefits, from relieving stress to improving brain functionality[3].
Economically speaking, game studios can achieve greater financial success by focusing on accessibility, as they are designing games that aim to be played by a broader spectrum of players.

4.3 Difficulty in Games

In the 1950s, a computer dubbed 'Bertie the Brain' allowed players to modify the computer's level of intelligence in a game of noughts and crosses [4]. This feature enabled the game to be *"easier, more entertaining, and definitely more accessible"* as more players could have a go at beating the computer.
Difficulty is an interesting topic in accessibility, as most games are designed with an element of challenge that the player must overcome. Some games may be strict on a single experience - *Dark Souls*(FromSoftware) is infamous for its difficult boss fights, lack of navigation and guidance on what stats (health, strength, etc.) the player should spend their 'souls'[3] on to upgrade. On the surface, Dark Souls seems like a game that was not built with accessibility in mind. Whilst that may be true, it is nonetheless considered one of the most popular video games of all time, with the franchise selling over 35 million units[5] and spawning an entire genre in gaming known as 'souls-like'.
Perhaps the key point when measuring a game's difficulty is judging whether it has been designed and implemented correctly. That means introducing the player to new mechanics and systems gradually, allowing them to build up more experience over time as they face harder challenges.

4.4 The Design and Implementation of Accessibility

When we consider how to design and implement accessibility correctly, we must recognise that every component of a game needs to be built with accessibility in mind. The purpose of each part is to give the player the best experience possible, whilst remaining accurate to the design's intent. We can start this process by considering how the player

[3] In Dark Souls, the game uses a currency called 'souls' that players gain from defeating Non-Player Character (NPC)s.

'experiences' a game.

4.4.1 Considering Player Experience

The following information was provided by Microsoft's *Gaming Accessibility Fundamentals*[6] course (see reference). I highly recommend it as a starting point for anyone interested in game accessibility.

Players receive output from a game through visual, audio and haptic feedback. With this information, the player is able to understand the state of the game, so they can respond appropriately by executing certain inputs with their input device(s); the game will update based on these inputs.
What I have just described is the primary feedback loop that can be found in any video game. If the loop is disrupted - for example, a player with hearing loss cannot hear the sound of incoming enemy NPCs - then their experience is negatively affected. To limit the impact of this disruption, developers can implement a range of visual, audio, cognitive and physical accessibility features. I will list a few of these features, but note that this is not exhaustive.
Visual: Text size, text colour, HUD scale, HUD colour (including the background and accent). For players unable to view the screen, a game should make use of haptic and audio feedback - for example, text-to-speech narration and spatial audio to represent environmental cues (such as the location of a nearby enemy).
Audio: Use visual cues to represent audio-based information. Additionally, provide multiple volume sliders to change the volume of various audio channels. This will help players who are less able to distinguish differences in sounds.
Cognitive: Three aspects to consider: attention, memory, and decision-making. With attention, games need to highlight objects that the player is required to focus on. This is most important during the tutorial section, which aims to teach the player the core mechanics of the game. In regards to memory, developers must recognise that players may forget how to interact with the game, or what their objective is. This can be relieved by including replayable tutorials, providing maps of the control scheme, and displaying a list of objectives. Finally, restricting choice in parts of a game may improve the decision-making experience. Providing too much choice can be overwhelming, so ensure that the design avoids this whilst still letting the player feel empowered by their ability to be in control.
Physical: Provide input support, such as allowing the player to use a range of input devices (keyboard and mouse, gamepad, adaptive controllers, etc.) and to customise the method of inputs required (i.e. whether a button needs to be pressed multiple times to perform an action, or if said button just needs to be pressed once).

4.4.2 Considering Player Experience in Multiplayer Gameplay

What separates the multiplayer experience from that provided by single player gameplay is the ability to communicate with other players. This can be through text and voice chat channels, but may also include other forms of interaction such as emoting and pinging. If we do not consider the potential barriers to communication (e.g. players speaking different languages, players with poor vision or hearing loss) then we aren't developing online functionality with accessibility in mind. Fortunately, innovation in this area has

Figure 4: Audio options in *Apex Legends*(2019)[10]

seen development in recent years as online games grow in popularity, and we can look at examples such as *Apex Legends*(Respawn Entertainment) for inspiration. One of the key features players can expect to find is Text-to-Speech (TTS)/Speech-to-Text (STT) technology.
TTS is the ability for a player to enter a message, that is then broadcast to players on their team through a synthesised voice. This feature is applicable for players who may be unable to speak due to certain conditions, such as not owning a compatible microphone or playing in a noisy environment.
STT is the ability for a player to take voice communication from their teammates and convert the speech into text displayed on the player's screen. For players who have a hearing impairment, this feature enables them to participate in communication.

When developing a game with online functionality, the need for accessible communication is crucial to ensuring consistency amongst players' experiences. All players on a team must be able to discuss components of the game, including strategy, tactics, and opportunities, if they wish to improve their chances of beating the opposing team.

4.4.3 Heuristics for Accessibility

For this project, I shall use Microsoft's XAG[8] as a list of requirements for implementing accessibility features. Whilst many general usability heuristics guidelines exist - such as Jakob Nielson's *10 Usability Heuristics for User Interface Design*[9] - XAG is a 'set of best practices that have been developed in partnership with industry experts and members of the gaming & disability community': in other words, it is tailor-made for accessibility in video games.

5 Game - Player Character (PC) Mechanics

5.1 Overview

In the following section, I will discuss the multiple components of the PC, a GameObject controlled by the player through input. I will detail the design and implementation of its mechanics, followed by an implementation of the player input system. This section will then finish with a discussion on the game's tutorial, which focuses on teaching the player the PC's mechanics.

5.2 Introduction

The PC will have mechanics similar to the moves available to players in traditional court-played ball games, such as basketball and netball. In these sports, players are able to move freely along the court except if they are holding the ball: if this is the case, players will either need to pivot in a single spot or move a limited number of steps until they have passed the ball. To be accessible, the mechanics/controls must be intuitive, and the input system must allow players to choose between different input devices - see 8.1 for a test and evaluation of these requirements.

5.3 Components of the Player Character (PC)

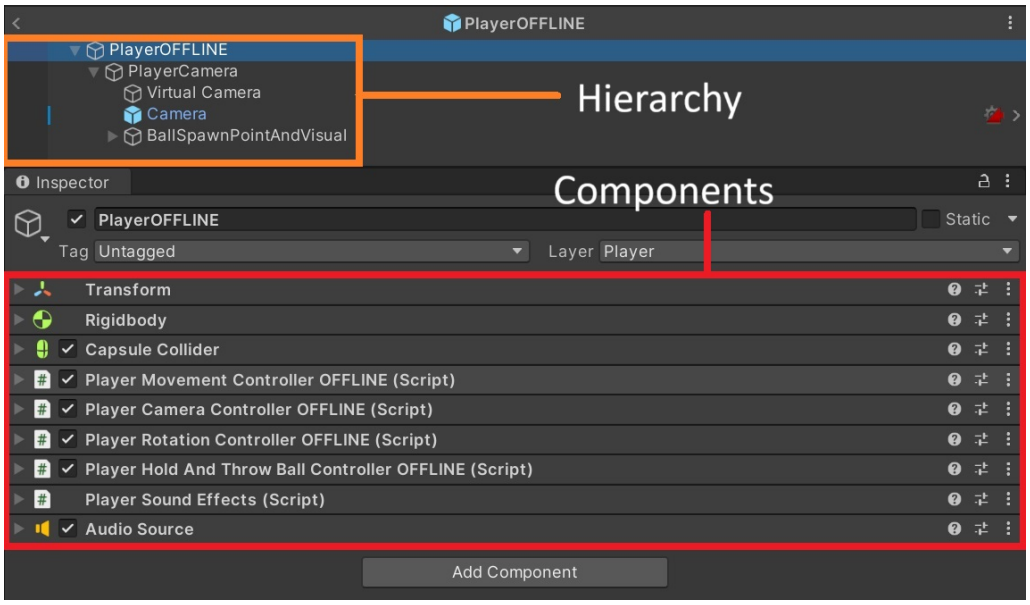


Figure 5: Hierarchy of Player(Offline) GameObject and components attached to the parent object.

5.4 Movement

5.4.1 Design

Being in a 3D environment, the PC has relative x, y and z axes for which it can move along.
Movement along the positive z axis will be considered 'forward' movement, with negative

z axis movement considered 'backward' movement.
Movement along the positive x axis is considered 'left' movement, with negative x axis movement considered 'right' movement.
Movement along the positive y axis is considered 'up' movement, with negative y axis movement considered 'down' movement.
Movement along these axes depends on input from the player as well as the state of the PC itself, which is handled by a simple state machine. The PC can be in one of three states per frame: Idle, Run, or Jump.

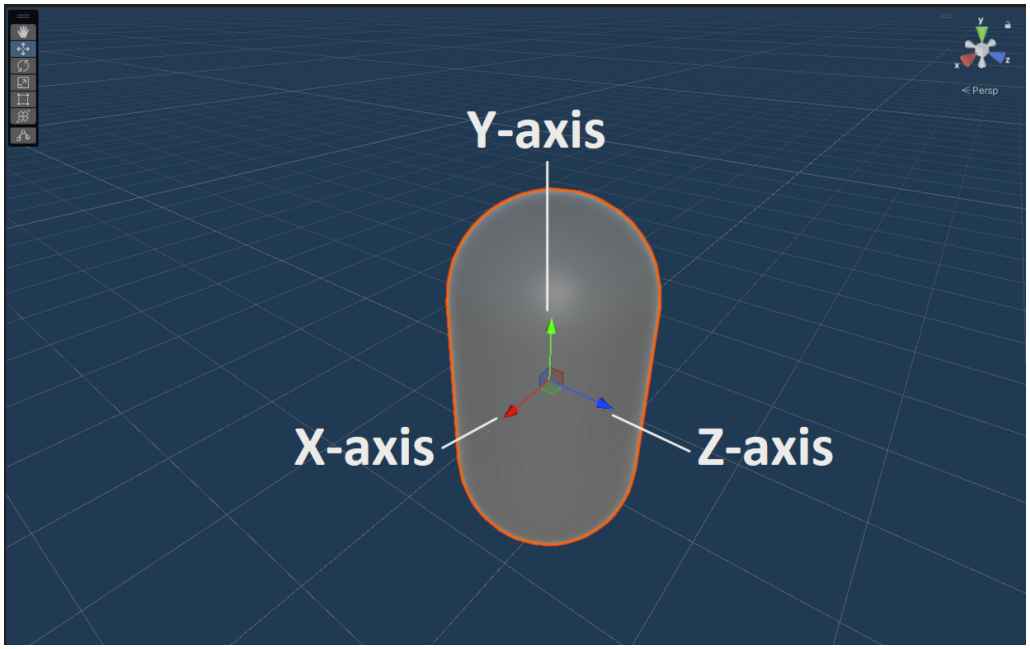


Figure 6: PC showing the direction of its axes in the Unity editor.

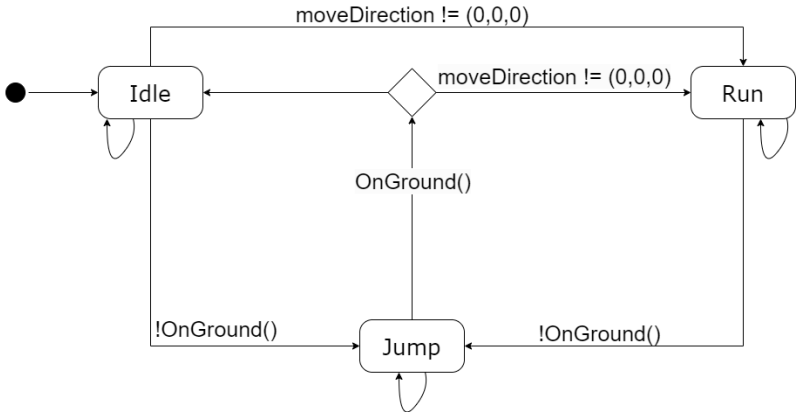


Figure 7: State Diagram for PC's Movements. Let *moveDirection* be a 3D vector variable storing the resultant vector of inputs pressed by the player per frame, and let *OnGround()* be a function returning a boolean value denoting whether the PC's ground collider is in contact with a 'ground' surface.

5.4.2 Implementation

Inspiration for programming the logic of the PC's movement came from Dave / GameDevelopment's *FIRST PERSON MOVEMENT in 10 MINUTES - Unity Tutorial* video on YouTube[32]. The solution includes a 'GroundDrag' function that's called in every 'fixed' frame. GroundDrag alters the drag variable of the PC's rigidbody component if the PC is on a grounded surface.
While a very useful source, modifications had to be made to the solution provided to accommodate the local scope of the PC. This is because of the Gravity Control mechanic (explained in 5.7) where the PC can rotate if it is near enough, and directly facing, a surface.

5.5 First-Person Camera

5.5.1 Design

To simulate the first-person perspective, the player's camera (the object which the player uses to view the game's environment) can move 360 degrees around the PC's y-axis, and 90-degree positive and negative movement around the PC's x-axis.

5.5.2 Implementation

Every frame, the 2D vector for the 'mouse look' input is checked. This input is based on the difference in the x and y axis movement of the player's mouse or right analog stick, depending on the chosen input device. This is recorded as a 2D vector *cameraInput* and multiplied by a variable *cameraSensitivity* set by the player. With this, we get the degree of rotation.

$$cameraX = cameraInput.y \times cameraSensitivity \tag{1}$$

$$cameraY = cameraInput.x \times cameraSensitivity \tag{2}$$

If camera movement is left or right (i.e. $cameraY > 0$), then the PC itself must rotate. If camera is up or down (i.e. $cameraX > 0$), then the camera object attached to the PC must rotate.

5.6 Pick Up, Hold and Throw Ball

5.6.1 Design

The PC will have the ability to simulate picking up the Ball during a match. They do so by getting near enough such that a spherical collider encompassing the PC collides with the Ball's collider. This places the Ball under the PC's possession, and remains there until the player controlling this PC decides to throw the Ball.

5.6.2 Implementation

Implementation of this mechanic first required design and implementation of the Ball. I will discuss this briefly.
One main feature of the Ball is that it has no direction of gravity. I chose this design decision because I believe it is easier for the player to throw the Ball to one of their teammates when their teammate's direction of gravity is different. Therefore, is just a

single force is applied to the Ball, in the direction of the z/forward axis of the player's camera.
Another feature of the Ball is how it bounces off surfaces in the game. When implementing this feature, I believed that the Ball moved too fast, making it hard to predict which direction the Ball was going to move in once it bounced off a surface. This may frustrate players who find it hard to focus on fast-moving objects. However, a fast speed is still required initially, as the main reason a player throws the Ball is to pass it to a teammate or to shoot it at the opponent's goal. Therefore, I altered the implementation so that the speed of the Ball halves each time it bounces off a wall. I found this modification to have excellent results at giving players the opportunity to pick up the Ball after it has been thrown.
For picking up the Ball, implementation followed mostly similar to its design. Inspired by traditional ball games, as well as a means of improving usability, I chose to have the game freeze the PC's position when they are in possession of the Ball.

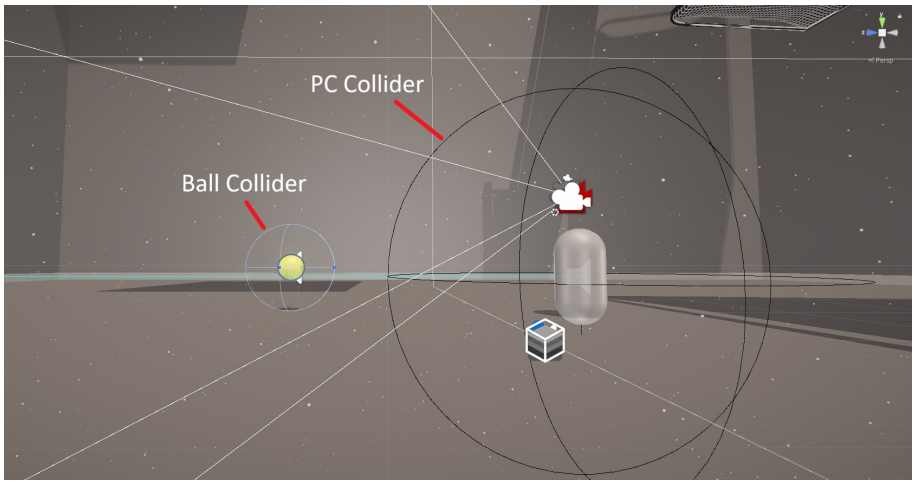


Figure 8: Ball and PC colliders near each other, but not colliding.

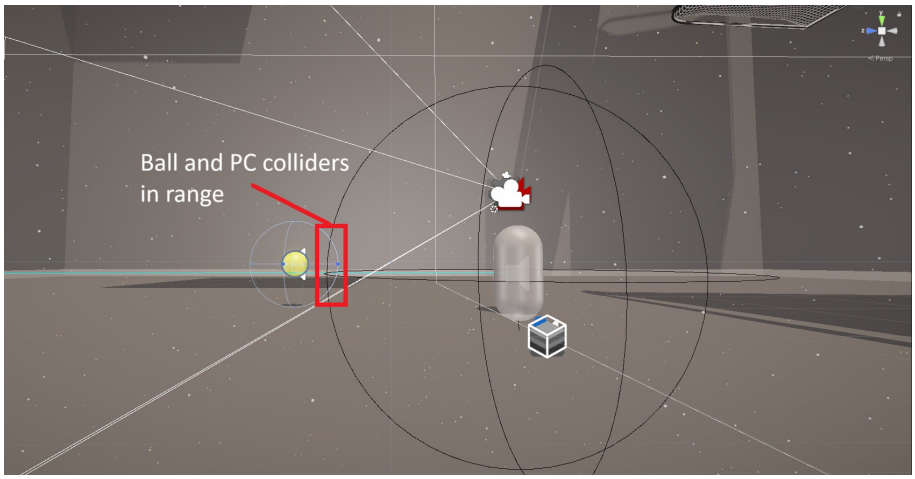


Figure 9: Ball and PC colliders in range of each other. Because of this, the Ball will now be attached to the PC and thus in the PC 'possession'.

5.7 Gravity Control

5.7.1 Design

The player will change gravity from the perspective of their PC, so that their PC can move on any surface. Games exist with this mechanic, notably *Manifold Garden*(William Chyr Studio LLC)[10], meaning I have a good foundation on which to take inspiration for my own solution of the mechanic.

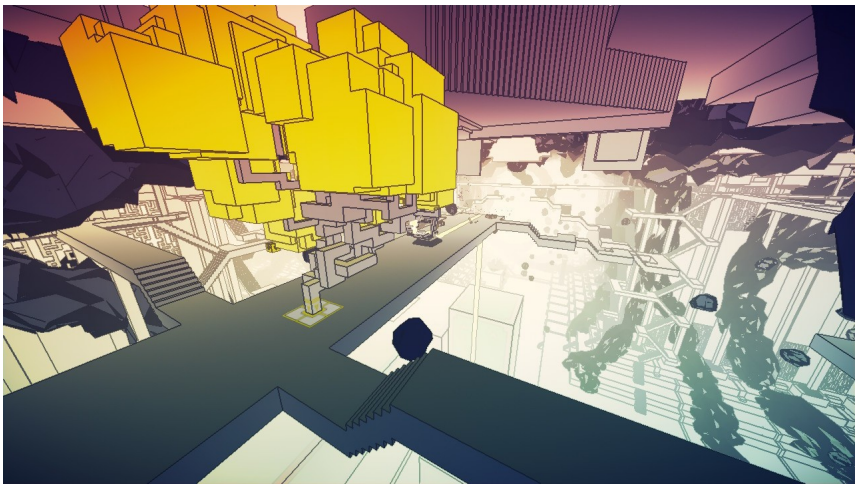


Figure 10: Manifold Garden allows the player to manipulate gravity in Escher-inspired levels, to solve puzzles. Source of image: `https://www.bonusstage.co.uk/archives/174081` [Accessed 24th April 2024]

5.7.2 Implementation

Implementation for the gravity control mechanic took a while to find a solution. In total, there were 4 distinct solutions created during development, which subsequent solutions embedding lessons learnt from the previous.

Solution 1: When the PC is facing a surface, the player can press a button that creates and fires a raycast from the PC in the positive direction of the PC's z axis. If the raycast collides with a surface object, then the rotation of that surface object is copied and applied to the PC.

A simple solution at first, this quickly created issues when I realised that copying the exact rotation of a surface means that you are also copying the direction of the surfaces' axes. This can create instances whereby the PC is facing backwards - a disorienting experience for the player and not a good solution for a game focused on accessibility. In addition, I had a component attached to every surface object that stated what the local downwards direction was for that surface (i.e. the direction of gravity for that surface). This made it hard to further develop an arena as adding new surfaces would require me having to manually input this information for every surface. Overall, a very poor solution.

Solution 2: The player could rotate their PC 90 degrees at any moment, against any axis.

This solution came from the idea that it gives the player more control over their PC. However, several problems arose when I tested this solution. The first problem was that sometimes, if I pressed the rotation button in quick succession, it appeared that the rotation for the previous input had not finished - but the rotation for the next input had started. This caused the PC to rotate such that it did not stand upright when on

a surface. Furthermore, I felt that this solution may be too challenging for a player to control perfectly, especially on a gamepad where the d-pad buttons would be used to control this mechanic.
Solution 3: When the PC is facing a surface, the player can press a button that rotates the PC so that it is directly facing a surface. The PC is then rotated 90 degrees backwards.
This solution provided promise at first, as initial testing showed the PC enacting the ability of moving on different surfaces. However, repeated playtesting created an issue where the PC would not stand upright. I deduced that this was because the player still has access to moving the PC whilst the PC is rotating. Locking player input was my initial thought, but I soon realised that it's impossible to determine how long the rotation actually takes to complete. I did not know how to solve this issue, and as a consequence I gave up on this solution.
Solution 4: Every frame, create and fire a raycast from the PC in the positive direction of the PC's z axis, with a distance such that the PC must be close to a surface. If the raycast collides with a surface object, find the normal of the point of collision and rotate the PC so that the player's up-axis is equal to the normal.
Out of all the solutions provided, **Solution 4** was the best. I believe that it is both the most sophisticated as well as the easiest to communicate with players. This is because by calling the functionality every frame, autonomy is provided; the player does not need to press a button when their PC is facing a surface to use the mechanic. Furthermore, the solution has excellent scalability, as the only external information (i.e. information not available from the PC) required is from the point of collision.

5.8 Player Input System (PIS)

One of the essential requirements I had devised for the Player Input System (PIS) was its ability to support multiple input devices, and for the player to easily switch between such devices in real-time. For this, I used Unity's Input System package. Unity's Input System allows you to create action maps, which are a collection of input actions. A single input action is able to be performed by multiple bindings: the Move action (related to movement mechanic at 5.4) in Figure 11 has WASD composite key binding and the left stick of a gamepad as default bindings.

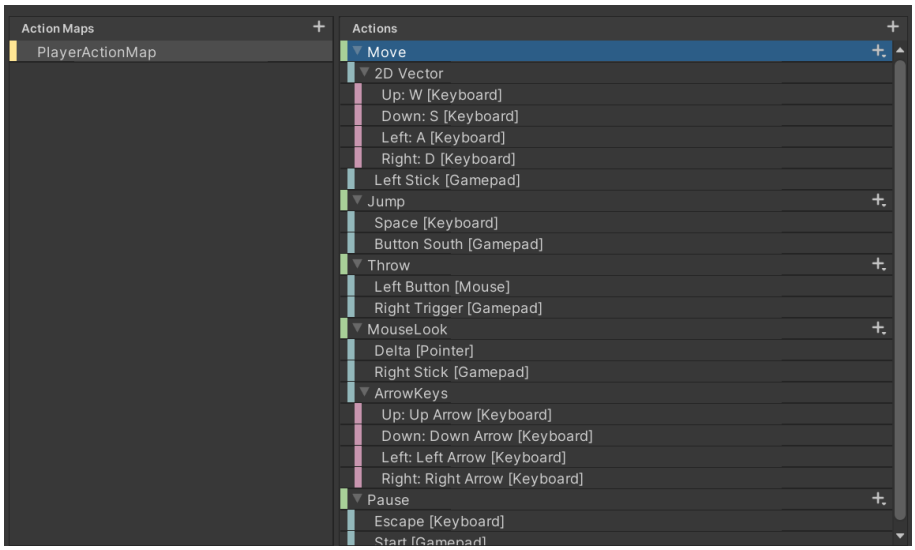


Figure 11: Creating an action map with input actions, using Unity's Input System

With this, I could now focus on how PIS handles these inputs, by creating a PlayerInputHandler class. While I had planned for PlayerInputHandler to be a component of PC, the class was eventually converted into a singleton, as to be a component of PC for online functionality the class would need to inherit from NetworkBehvaviour, while for offline gameplay (tutorial, free play) it would inherit from MonoBehaviour. In other words, we would need to create two versions of PlayerInputHandler despite there being no difference between versions. The singleton pattern is the most optimal solution because it allows other classes to check what input actions have been performed/cancelled. We can manage the problems associated with singletons here because only one instance of listening to inputs is required.

5.9 Tutorial

The tutorial was an important, lengthy part of development that focused on teaching the player the PC's mechanics in a way that was easy to understand all the while offering challenging interaction, such that it required the player to gain a firm grasp of the mechanics to complete the objective of each level. In development, I defined 2 types of objectives:

1. Pick up Ball - The player must move the PC close enough to the Ball so that the two GameObjects collide.

2. Score goal - The player must 'pick up' the Ball and 'throw' it into the goal. In other words, the Ball must collide with the internal collider of the Goalzone GameObject in the scene.

Upon completing the objective of a level, the player is able to progress to the next. The order (and thus the number of levels) in which to teach the player the mechanics had to be decided, and can be seen in Table 1.

Level	Objective	Purpose
One	Pick up Ball	Teaches movement and picking up Ball
Two	Pick up Ball	Teaches jumping
Three	Score goal	Teaches throwing the Ball (into Goalzone)
Four	Pick up Ball	Teaches changing gravity
Five	Score goal	Teaches throwing Ball into Goalzone that's on a different surface to the PC's initial surface
Six	Score goal	Teaches throwing Ball onto surface for it to rebound
Seven	Pick up Ball	Combination of the PC's mechanics to assure player has an understanding of them

Table 1: Brief outline of objective and purpose of each level in the tutorial.

In the final version, there are seven levels. My initial intention was to implement eight levels, with the eighth being a more challenging version of level seven. After devising the design of level eight I soon felt that this may impact UX in case the player struggles too much with level seven (meaning the challenge is overwhelming, instead of engaging). I decided to leave level eight so that I could spend time on other tasks. In the event that, during User Test 1, the majority of testers find level seven less challenging than I had designed, then I could consider an eighth level (or re-designing level seven) as future work.
To inform the player of the input system's different bindings, the game displays the appropriate information on the bottom-left side of the HUD. The input bindings change based on what input device the player is using (see Figures 12 and 13).

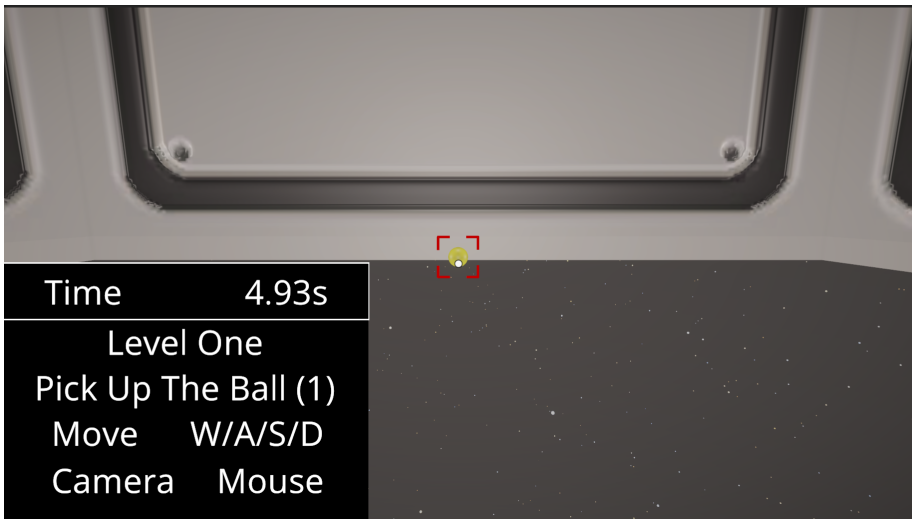


Figure 12: Tutorial HUD showing movement and camera inputs when a keyboard and mouse is in-use.

Figure 13: Tutorial HUD showing movement and camera inputs when a gamepad is in-use.

6 Game - Settings Menu

6.1 Overview

This section discusses the development of the game's settings menu, including the choice of settings, how the settings were implemented, and how the player interacts with the menu. In addition, this section includes information on the issues with the first iteration of the menu, what was learnt from these issues, and how the second iteration aimed to solve these issues.

6.2 Using Xbox Accessibility Guidelines (XAG)

In an ideal scenario, the game would have as many explicit accessibility features implemented as possible. However, time limitations mean that only a handful of these features can make it into the game. I have used XAG as a reference point - see Table 2 - whilst developing the settings menu.

6.3 Design and Implementation

I started with the front-end design of the settings menu, as this would allow me to get a better understanding of the UX - after all, the menu itself needs to be accessible to the player if they want to change settings. I used PowerPoint to create digital equivalents of a paper prototype - see Figures 14-20 - before implementing the design into the game. I took inspiration from Figure 4, given that Apex Legends was designed with accessibility in mind and it's a game I've already discussed.

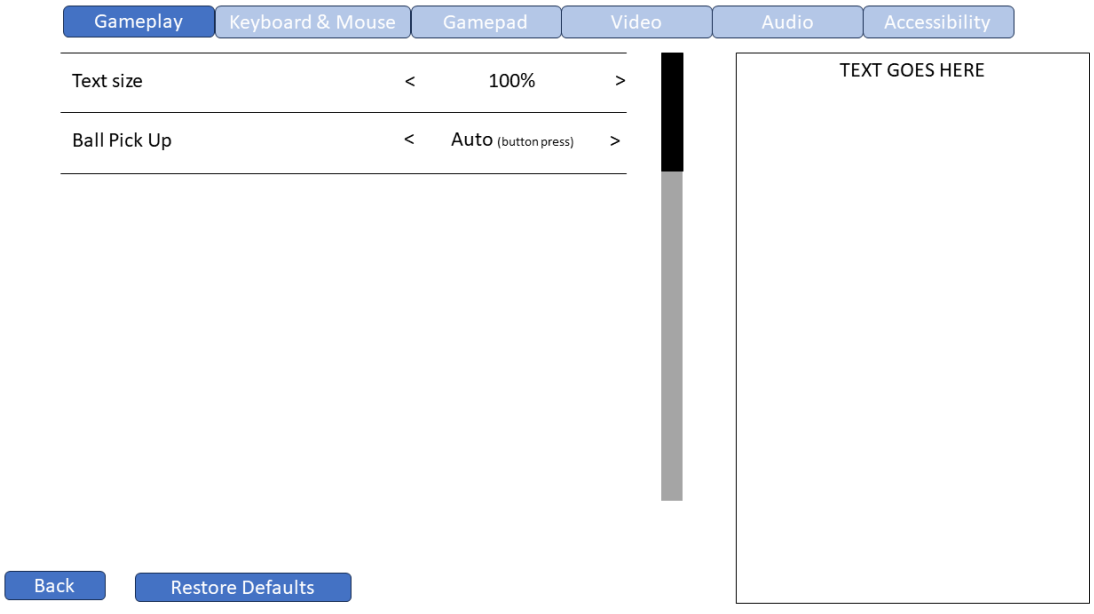


Figure 14: Gameplay Settings Version 1 Prototype

Guideline	Goal
101 - Text display	Ensure that text readability is optimized for all players, including players with low vision. This can be achieved by displaying text at minimum default sizes and spacings and providing configurable style and colour options.
102 - Contrast	Provide enough contrast between visual elements and their backgrounds so that these elements can be interpreted by gamers with low vision.
106 - Screen narration	Ensure that all on screen visual information can also be represented aurally through screen narration software. This benefits players who can't read on screen content because of things like blindness, low vision, learning disabilities, or temporary/situational circumstances.
112 - User Interface (UI) navigation	Provide players with clear and consistent UI navigation experiences throughout the entirety of a game. This can help players who are neurodiverse, have learning disabilities, are new to gaming, or are navigating the experience with various types of assistive technologies.
113 - UI focus handling	Ensure that players, including players with low vision, are always aware of which element in a game UI has focus.
114 - UI context	Ensure that players have enough context to operate a game's interface and understand its UI components and their functions. This can be especially helpful for players who need additional time to read screen content, for players with limited short-term memory, and players who are new to the game.
115 - Error messages and destructive actions	Ensure that players can identify and correct any player-input errors before permanent or destructive action takes place.
116 - Time limits	Ensure that players have adequate time to read, interpret, and interact with all the UI in the game.
117 - Visual distractions and motion settings	Ensure that players can pause or completely stop any content that scrolls, blinks, auto-updates, or otherwise moves. Additionally, players should be able to customize game settings related to Field of View (FOV), camera movement and sensitivity, and other visual settings related to camera or screen movement. This benefits players who are susceptible to motion sickness, players who are easily distracted by blinking, scrolling, or otherwise moving content, and players who are unable to quickly read text before an auto-update or scrolling occurs.

Table 2: List of the most relevant XAG guidelines (including the goal of each guideline) for the development of the settings menu.

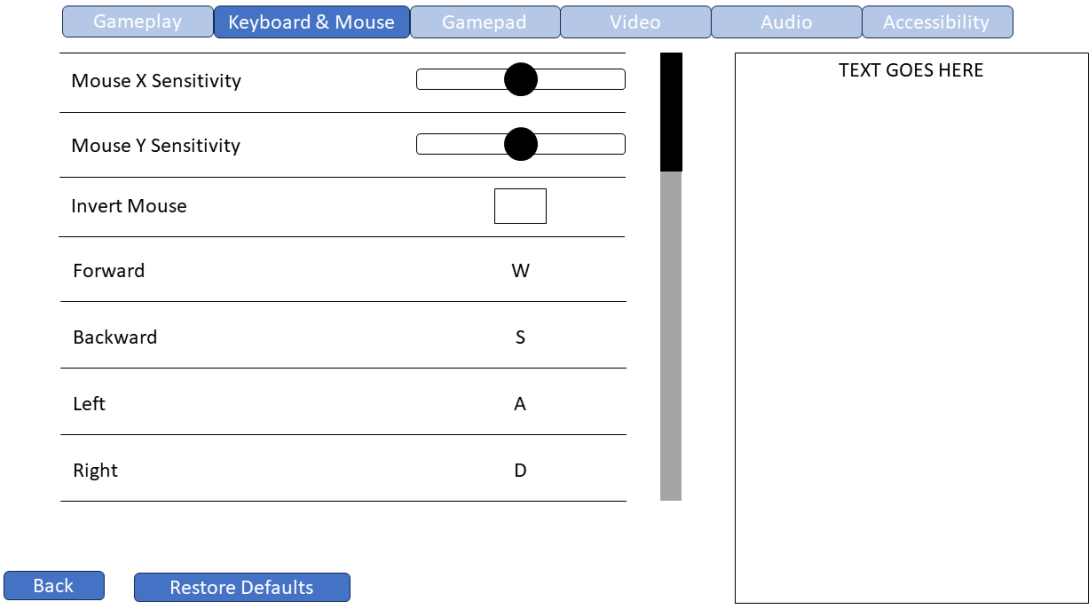


Figure 15: Keyboard and Mouse Settings Version 1 Prototype [part 1]

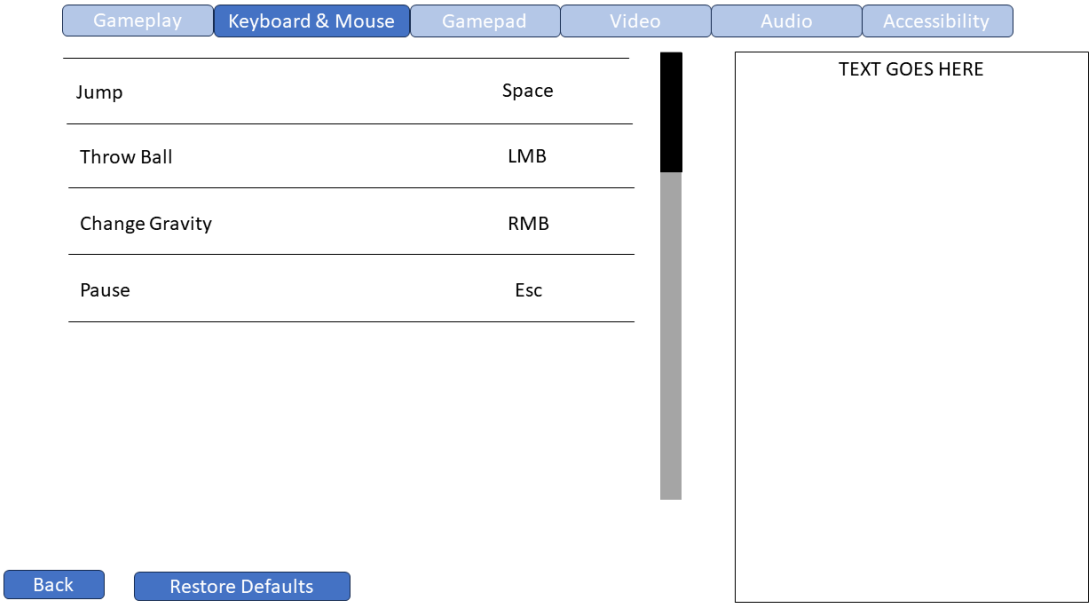


Figure 16: Keyboard and Mouse Settings Version 1 Prototype [part 2]

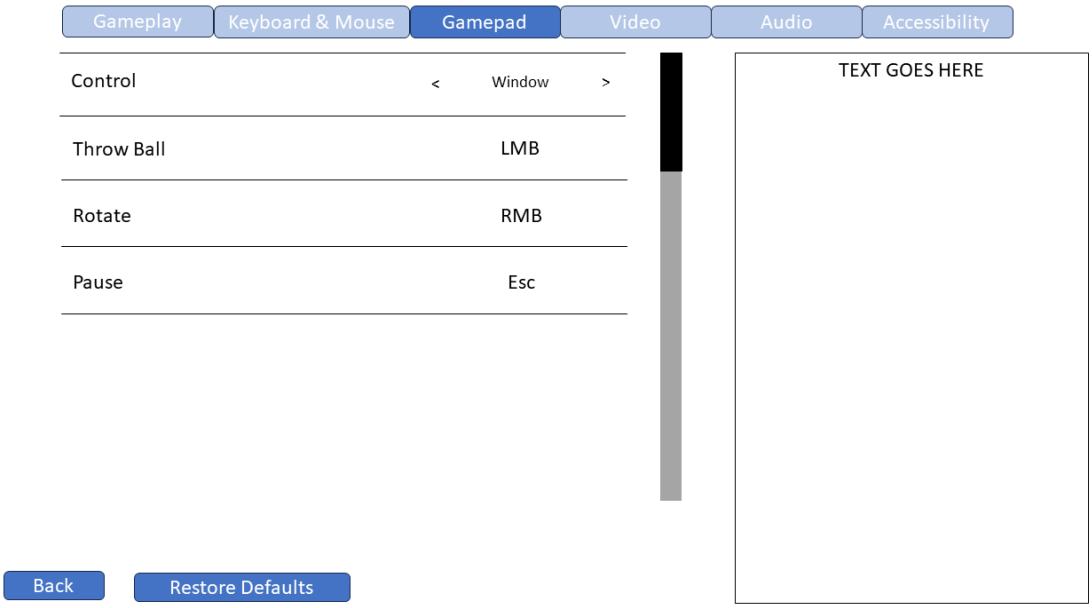


Figure 17: Gamepad Settings Version 1 Prototype

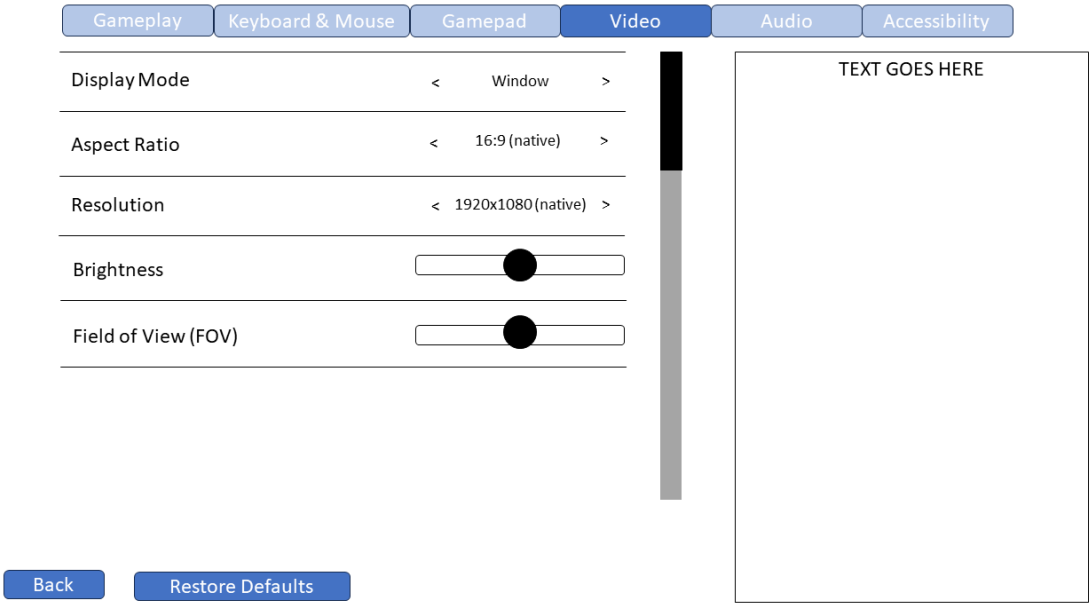


Figure 18: Video Settings Version 1 Prototype

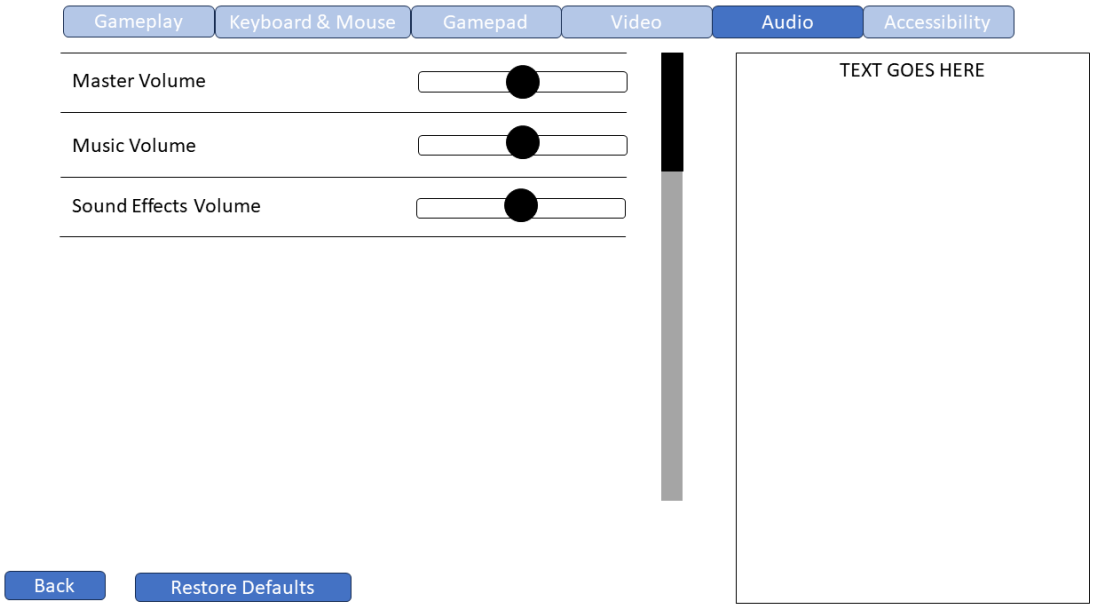


Figure 19: Audio Settings Version 1 Prototype

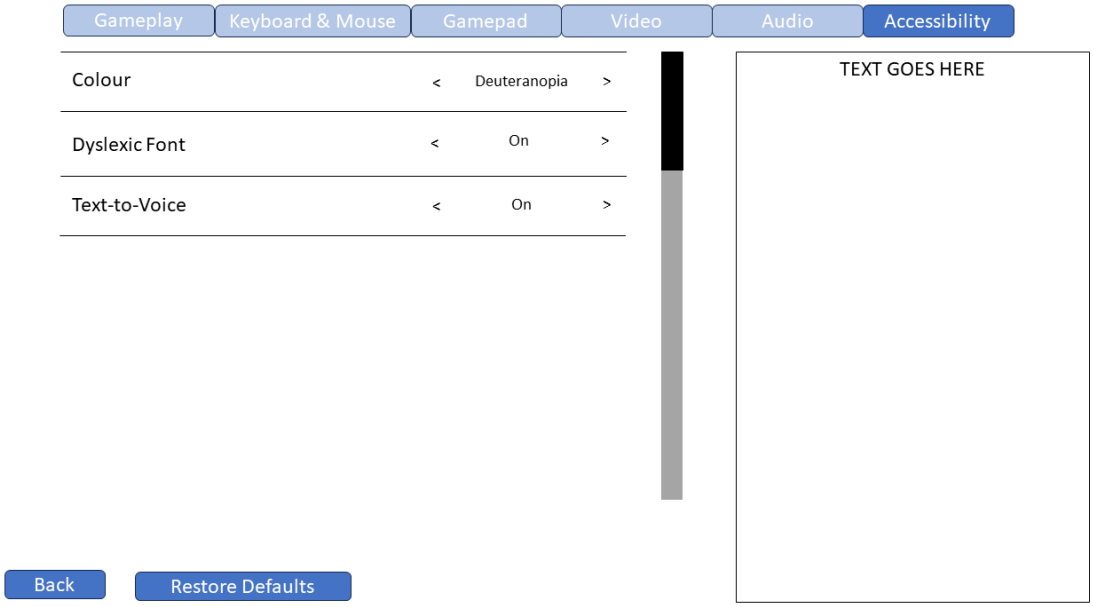


Figure 20: Accessibility Settings Version 1 Prototype

A main component of this design is the ability to add multiple settings to one category of settings, which can be scrolled through using a scroll bar. When the player hovers their mouse over a setting, the 'TEXT GOES HERE' box on the right-hand side shows a brief description of the setting. I chose to include this feature in the design because it satisfies XAG 114, helping players who may be new to gaming - for these players, terms such as 'field of view' may be unfamiliar to them.
Implementation of this design created no logical or UX issues until I added gamepad support. Upon adding gamepad support, I discovered that Unity's UI navigation system, which allows players to select a 'selectable' game object (e.g. button, scroll bar, slider)

and move between different selectable game objects in the interface, did not have appropriate support for scrolling through the different settings. This is because the game objects which are selected based on whether the player presses up, down, left or right, thus choosing the selectable game object for Select On Up, Select On Down, Select On Left, Select On Right (see figure 22), respectively, are set in the editor - in other words, they cannot be changed at run-time. To scroll through a list of settings with a scroll bar would mean changing which setting is selected based on how much the player has scrolled to the bottom/top of the list.
There was another significant issue, that I had not taken into account during design: XAG 101, which recommends a maximum font size of 72 for 4k screens (as 4k screens have more pixels than, say, a 1080p screen, a 4k screen requires the use of more pixels to display text of the same 'size' as a 1080p screen, in the eyes of the player). Changing the font size to 72 made it impossible to easily navigate the interface.
Overall, I had two significant problems to solve. The first problem was to accommodate a font size as large as 72. The second problem was to re-design the settings menu so that it removed the need for a scroll bar.

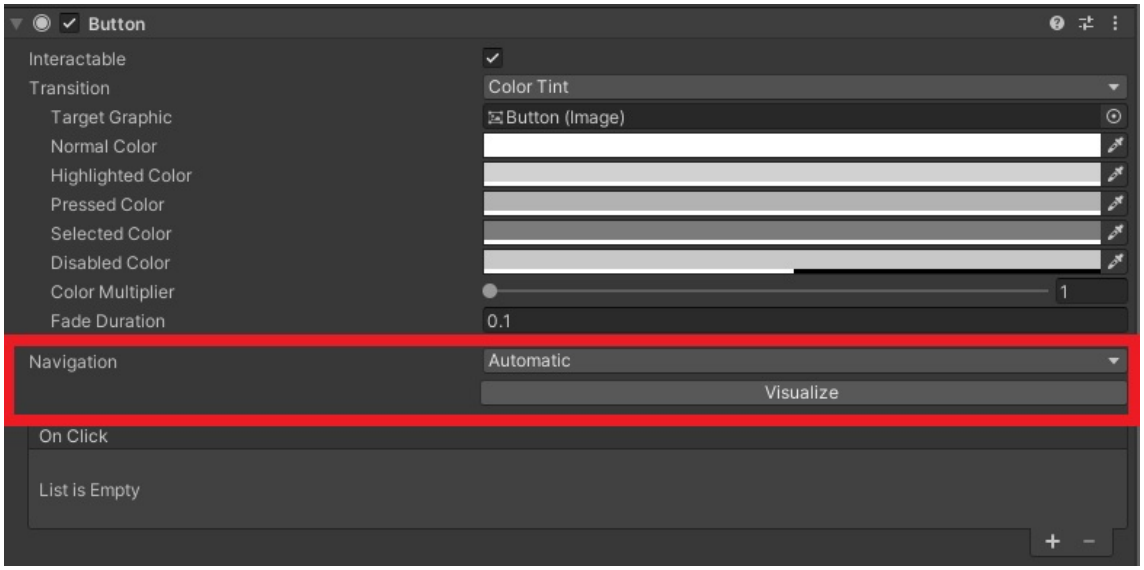


Figure 21: Choosing automatic navigation for a button game object

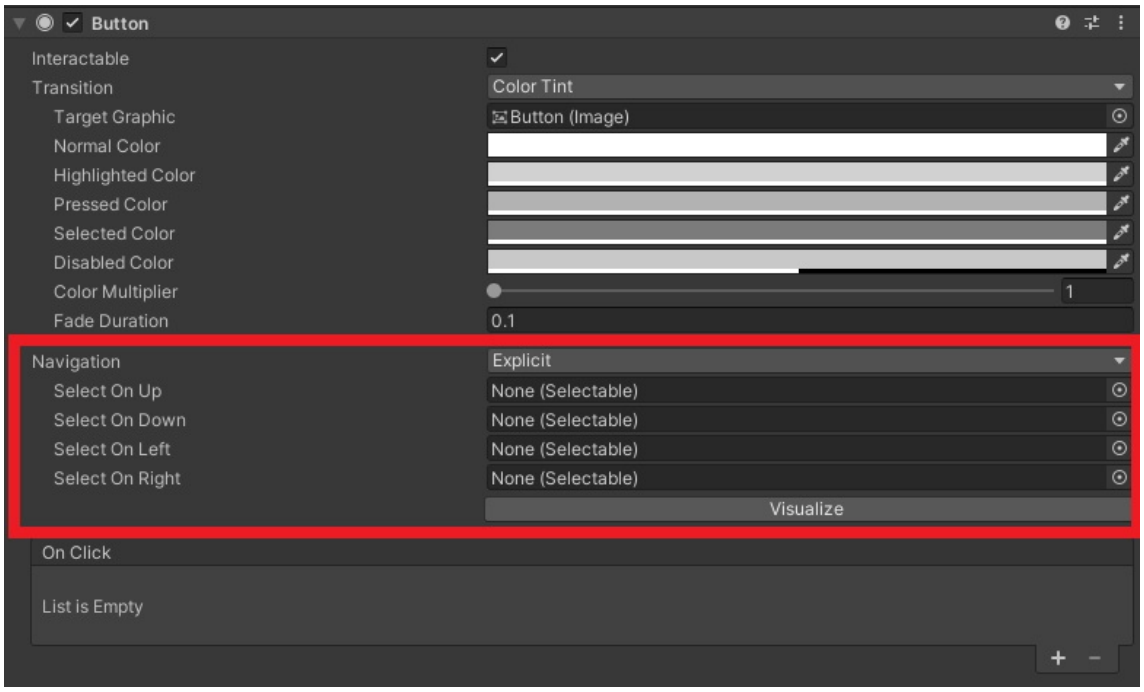


Figure 22: Choosing explicit navigation for a button game object

With this in mind, I went about designing the second version of the settings menu. The menu would have a limited amount of settings that could be available per page; this would mean creating a deeper tree of pages where a page shows either a list of buttons for other pages (Figure 23), a list of settings (Figure 24), or both (Figure 25).

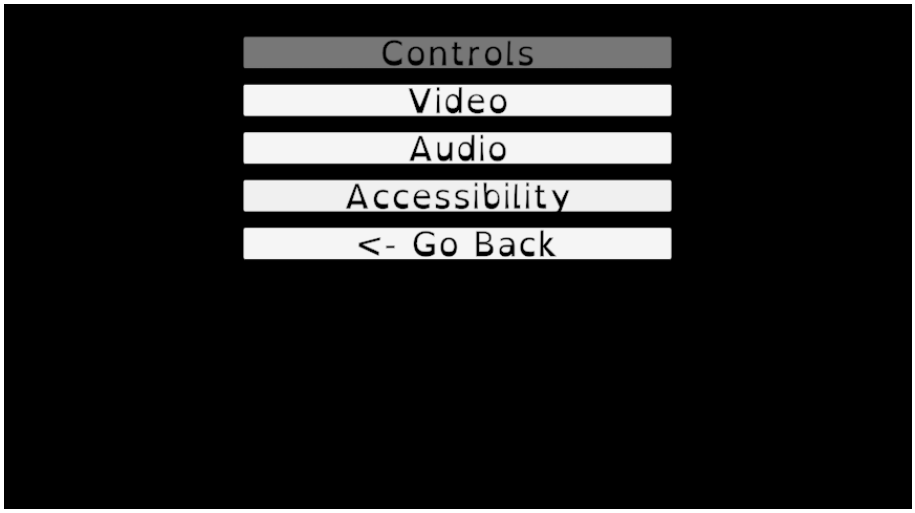


Figure 23: Home page for settings menu Version 2, with 'controls' button selected

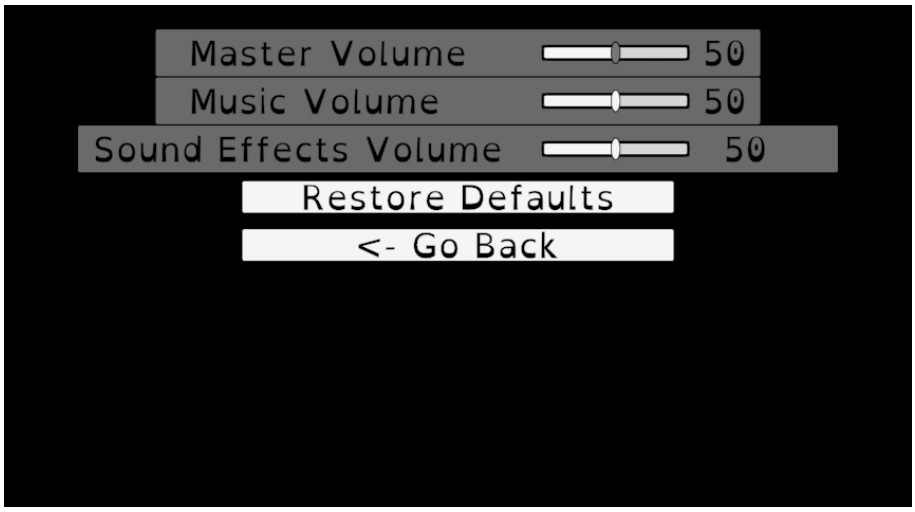


Figure 24: Audio page for settings menu Version 2, with 'master volume' slider selected

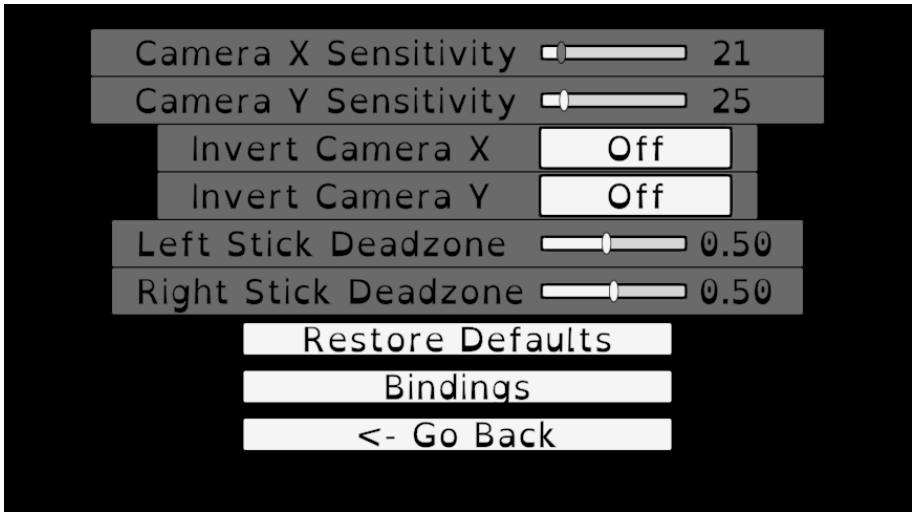


Figure 25: Controls page for settings menu Version 2, with 'Camera X Sensitivity' slider selected. Includes button to show Bindings page

The currently selected game object is highlighted with a dark grey, as a contrast to the white fill of a non-selected game object. This satisfies XAG 113 because the player is always aware of which game object is in focus.
With the front-end design complete, I then focused on implementing settings.
A method of easily scaling the number of settings was an essential component to saving development time as well as to improve the efficiency of system and asset resources. This was something I had realised only after implementing version 1, where I had created scripts that handled every setting per category. This created highly inefficient code, as I was often duplicating functions and changing just the reference for the selectable game objects that controlled the settings. I decided that for version 2 I would need to improve my code efficiency. Upon completion of the front-end, there were four interface variations the player would interact with to customise their settings:

- Variation 1: a button, for enabling/disabling a feature (Figure 26)
- Variation 2: a button, for changing the keybind of an input binding (Figure 26)

- Variation 3: two buttons, for choosing an option from a list (Figure 28)
- Variation 4: a slider, for increasing/decreasing a numerical value (Figure 30)

To improve the efficiency for version 2, I created prefabs for these four variations. For variation 1 and 2, only a single button prefab was necessary. Therefore, three prefabs were created.

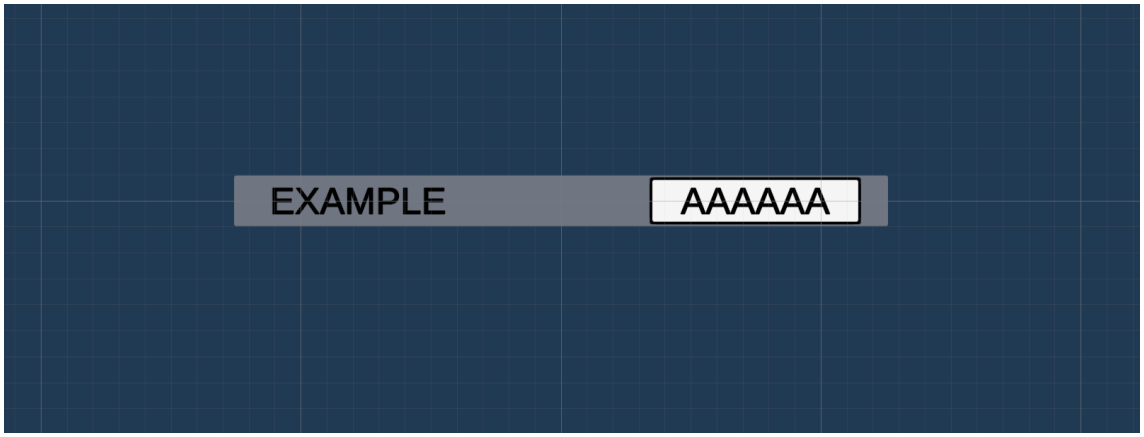


Figure 26: Settings Button Template Visual

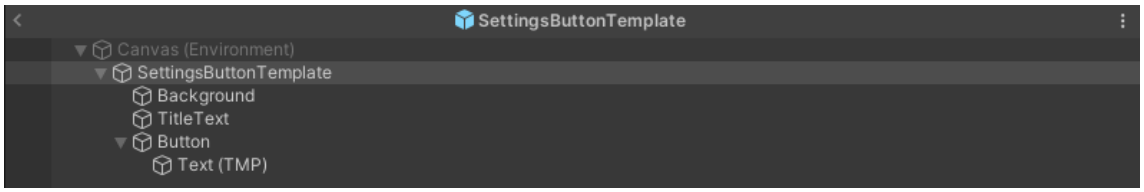


Figure 27: Settings Button Template Hierarchy

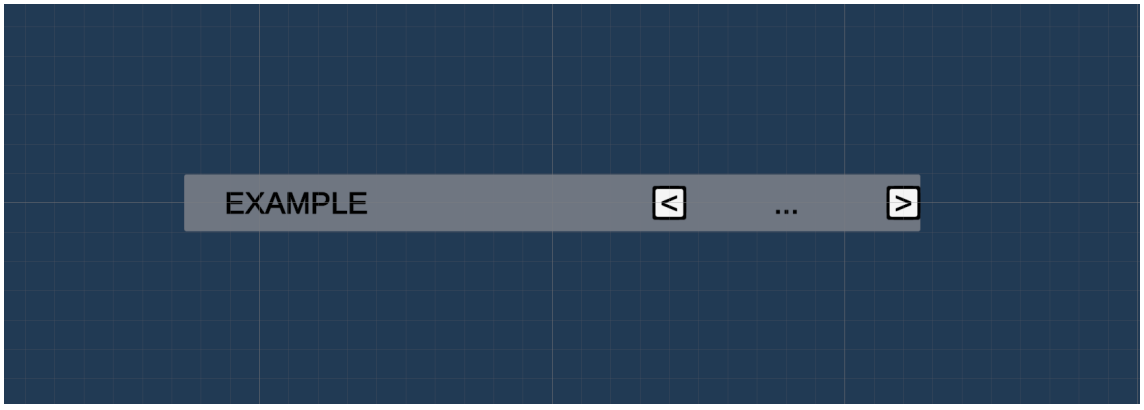


Figure 28: Settings Choice Template Visual

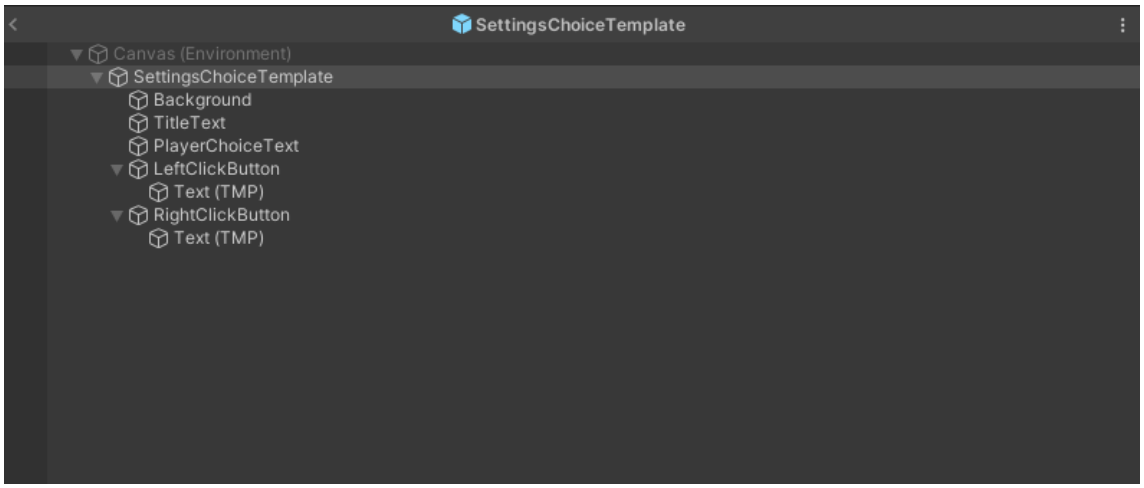


Figure 29: Settings Choice Template Hierarchy

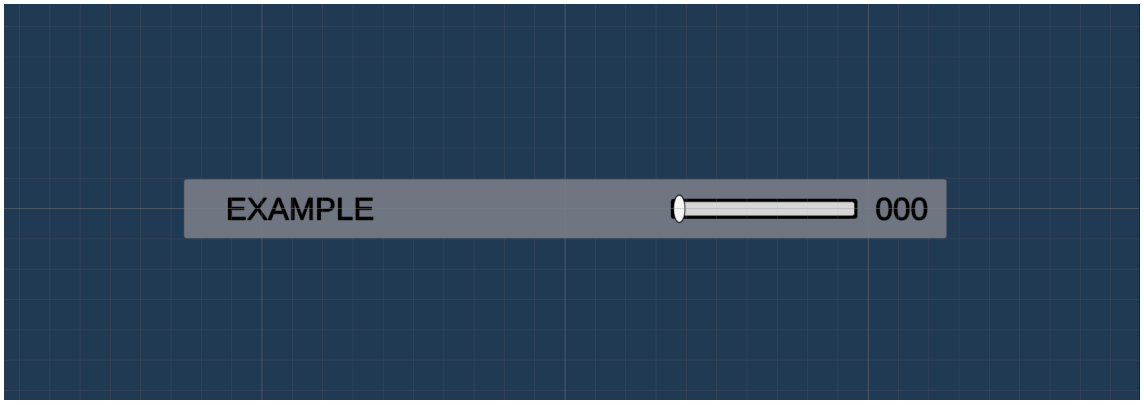


Figure 30: Settings Slider Template Visual

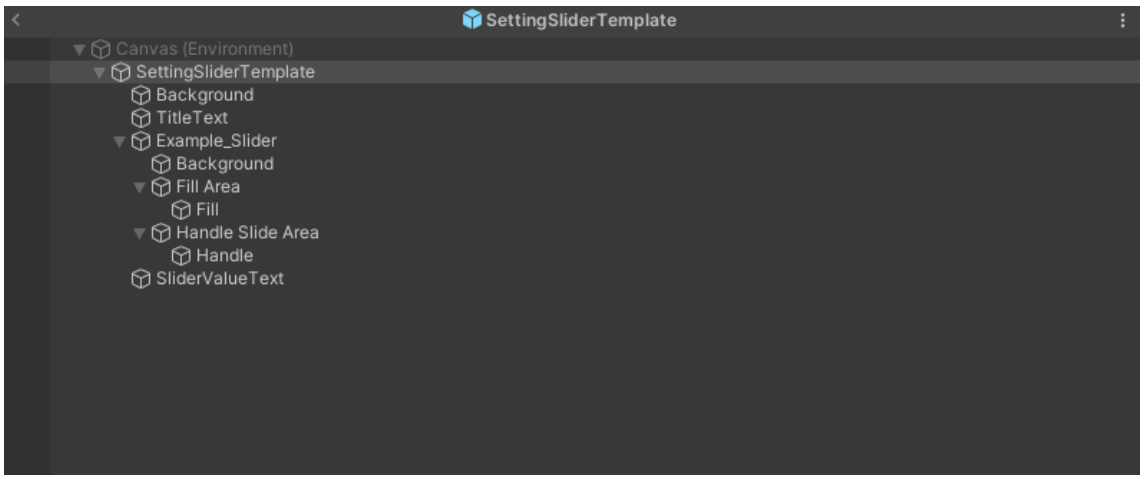


Figure 31: Settings Slider Template Hierarchy

In addition, a back-end solution was necessary for storing and updating the settings - for this, I used SOs and PlayerPrefs (the reason for this choice can be found at 3.3). When creating the SO classes, I tried to generalise the classes so that they represented the variable type. This meant a single SO class could be instantiated for multiple types

of settings. This worked in some cases, however there were times where the class represented a more specific type - such as 'FullScreenModes'. In the final prototype, the player has access to the settings found in Table 3.

Category	**Settings**
Controls	• Camera Sensitivity • Invert Camera • Gamepad Stick Deadzone • Keybinds (for Keyboard, Mouse and Gamepad peripherals)
Video	• Display Mode • Aspect Ratio • Resolution • FOV
Audio	• Master Volume • Music Volume • Sound Effects Volume
Accessibility	• Ball Colour • Ball Marker Colour • Ball Marker Size • Reticle Size • Reticle Colour • Text Size • Text Font

Table 3: List of Implemented Settings

As the player will need to view different pages, this means enabling/disabling the visibility of pages. Therefore, there are two similar functions: `Show()` and `Hide()`. I decided to write these functions inside a super-class which each menu-related class

inherits from. Because I have to create other menu interfaces that won't be directly linked to settings, I also decided to create an intermediary *settings* super-class. See Figure 32 for an understanding of this inheritance structure.

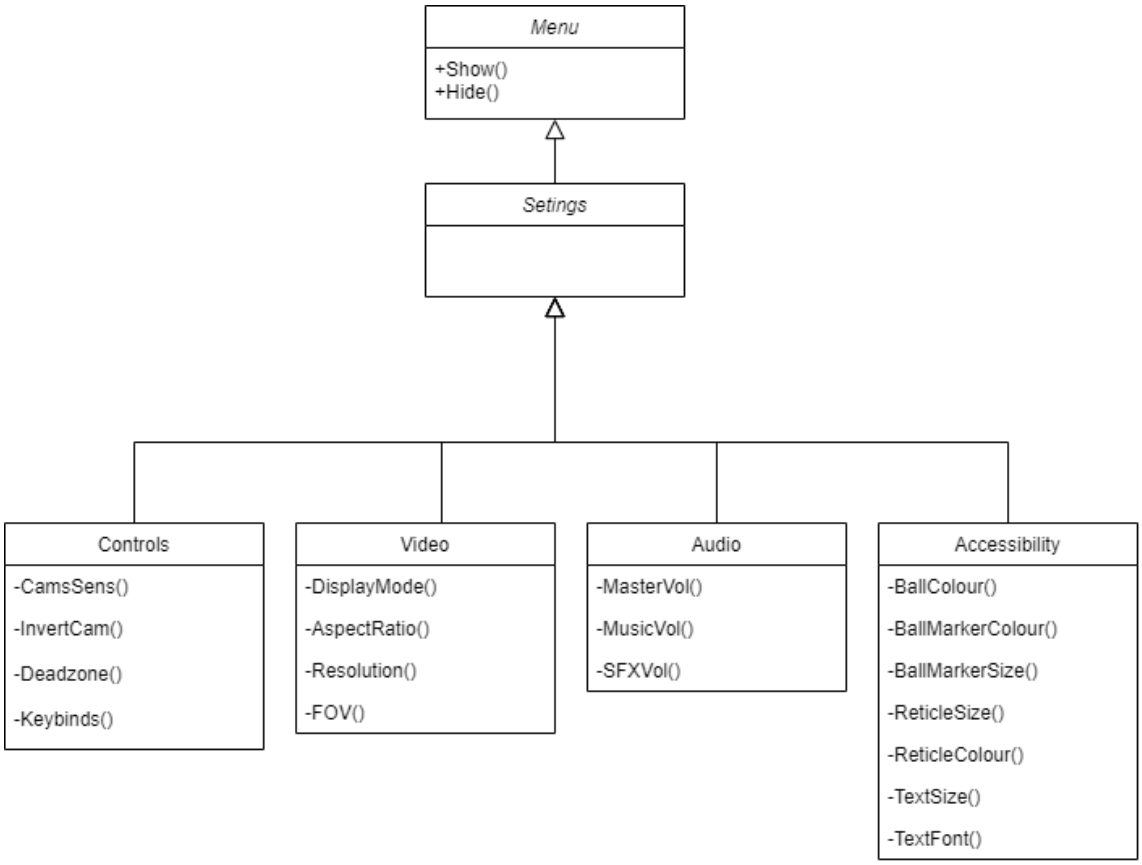


Figure 32: Class diagram showing inheritance design for settings menu.

6.3.1 Controls

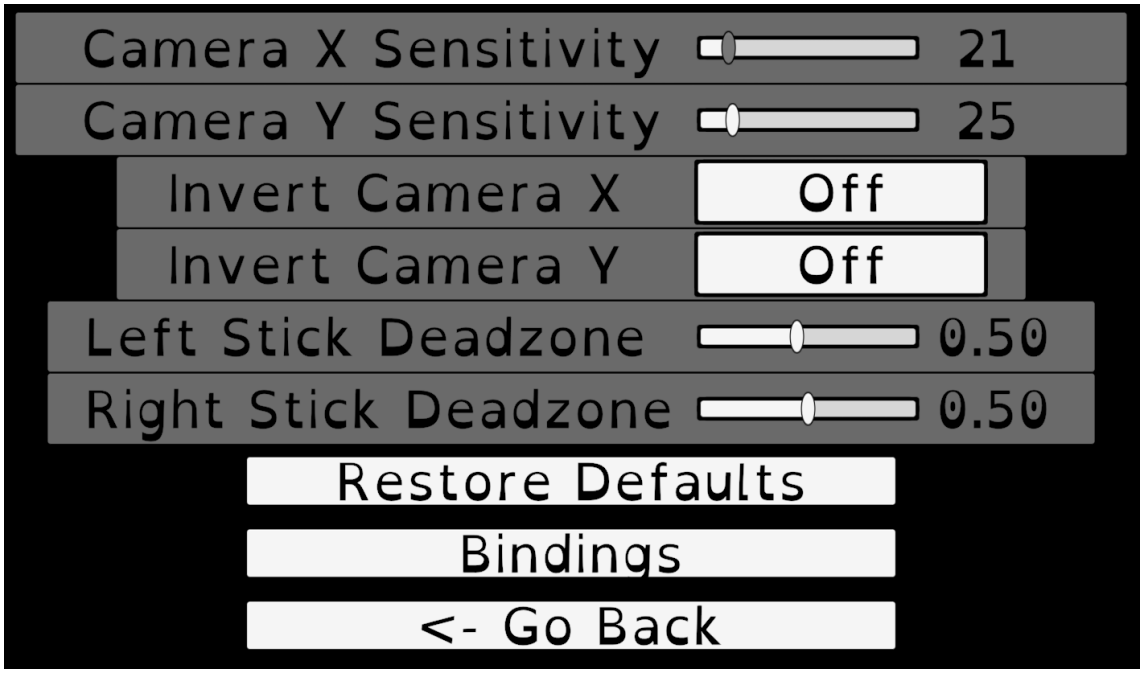


Figure 33: Control settings page.

Control settings were associated with input from the player, whether that be with a keyboard, mouse, or gamepad.

Camera Sensitivity

Adjusting the sensitivity at which the camera moves is modified by the player through a slider template (Figure 30). Players may prefer a different level of sensitivity along the x and y-axis, so individual sliders were implemented to allow for this use case.

Invert Camera

The 'invert camera' setting is two-sided: either the player wants the camera to be inverted, or they don't. Therefore I used a button template (Figure 26). Similar to camera sensitivity, the player can choose whether to invert only the camera's x or y movement, or both.

Stick Deadzone

Deadzone is specific to a gamepad, as it defines the point at which the game should register input from an analog stick. The setting is popular in games as players may experience 'stick drift', where a gamepad's analog stick(s) no longer responds accurately to the player's input - or when the objects controlled by the analog stick move/drift on-screen without touching the stick.

Bindings

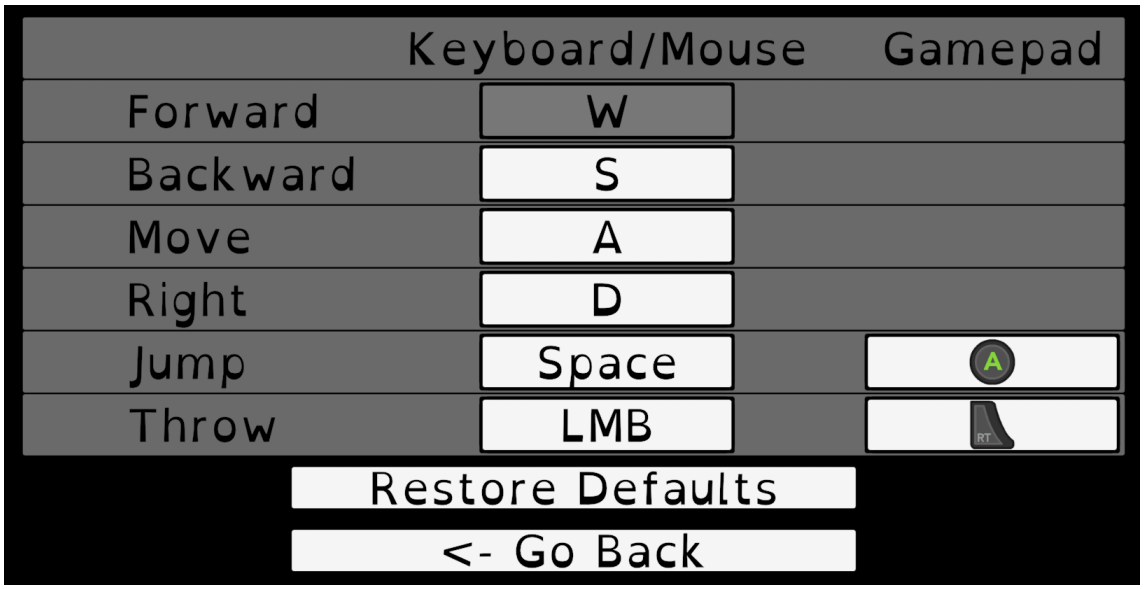


Figure 34: Keybind page.

With the minimal set of inputs for controlling the PC, all the bindings can be displayed on a single page. I have chosen to exclude a few inputs, specifically camera control (for all input devices) and movement (for gamepad) - as these inputs are standard in many first-person games.
If the player chooses to re-bind a binding, a display (Figure 35) will show up informing the player they have to input their new keybind. If they wish to cancel this action, they can do so by pressing the 'Esc' key.

Figure 35: Display informing the player to choose a new keybind.

6.3.2 Video

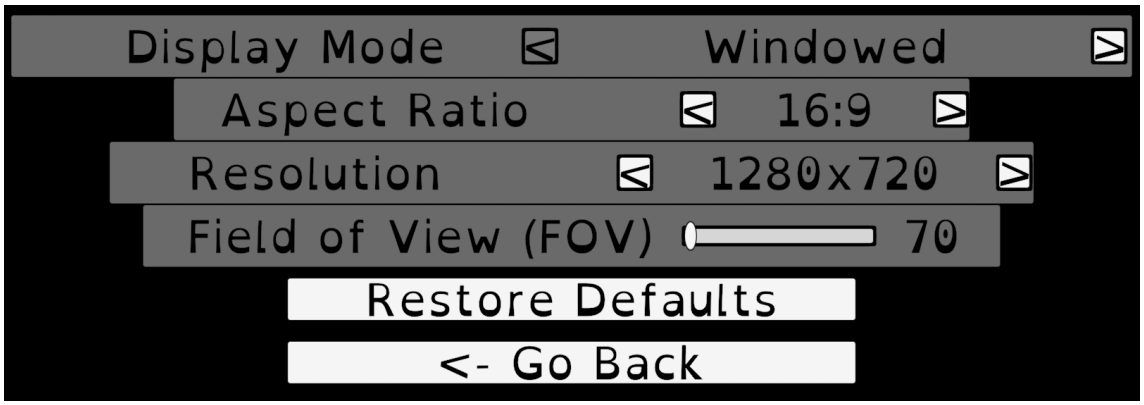


Figure 36: Video settings page.

The video settings are associated with how the game is displayed.
Display Mode
For numerous reasons, players may prefer to have the game in a window tab or in fullscreen mode. With different options, I opted for the choice template (Figure 28).
Aspect Ratio + Resolution
Resolution is the number of pixels on a display, often labelled based on length and height. Displays will usually have either a 4:3, 16:9 or 16:10 aspect ratio, which is the ratio of pixels length:height. In other words, the aspect ratio determines which resolutions can be used - in Figure 36, 1280x720 is associated with the 16:9 aspect ratio.
Field of View (FOV)
Due to hardware limitations, some players may prefer a FOV of a smaller value as this ensures a smaller space is rendered on-screen.

6.3.3 Audio

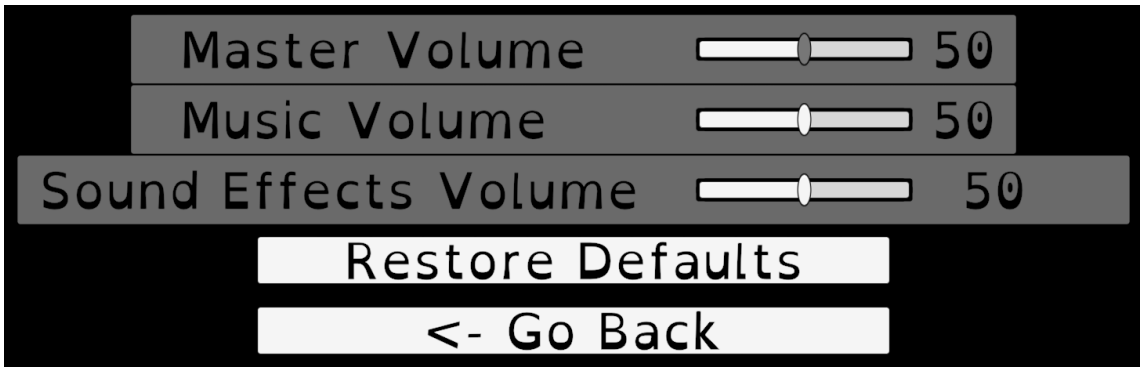


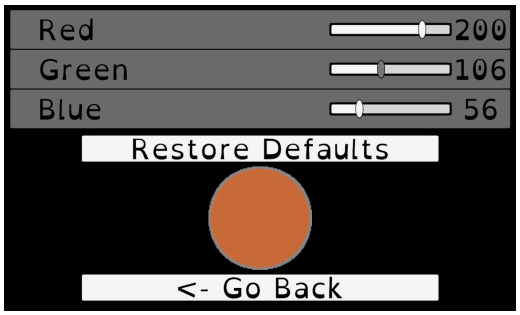
Figure 37: Audio settings page.

The audio settings are associated with the output of sound from the game.
Master Volume + Music Volume + Sound Effects Volume
Found in many entertainment applications, the ability to control the volume of sound is important for players who may want to prioritise certain sounds over others. The addition of a 'master volume' slider allows players to easily adjust the overall volume of all sounds.

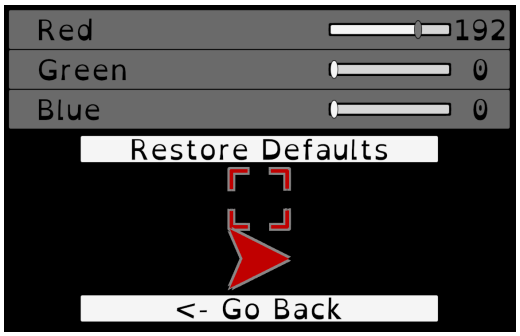
6.3.4 Accessibility

The accessibility settings are associated with settings that aren't specific to Controls, Video, or Audio, but are still important to enhancing the player's experience.

Ball Colour + Ball Marker Colour + Reticle Colour



(a) Ball colour page (Reticle Colour page looks identical).



(b) Ball marker colour page.

Figure 38: Colour pages.

Players with colourblindness, or who may find trouble in the contrast of the Ball and ball marker to other objects within the game, are able to change the colour to their preferred setting. The inclusion of primary colour (Red, Green, and Blue) sliders, where each slider is representing a value from 0 to 255 (inclusive), provides the player with just over 16.7 million different colour combinations.

Ball Marker Size + Reticle Size

(a) Ball marker size page.

(b) Reticle size page.

Figure 39: Size pages.

Adjusting the size of the ball marker and reticle is useful for players with higher resolution displays, where such HUD objects may be compressed to a smaller size. In addition, players with low vision may be able to see the objects more easily on their display by increasing the size.

Text Size

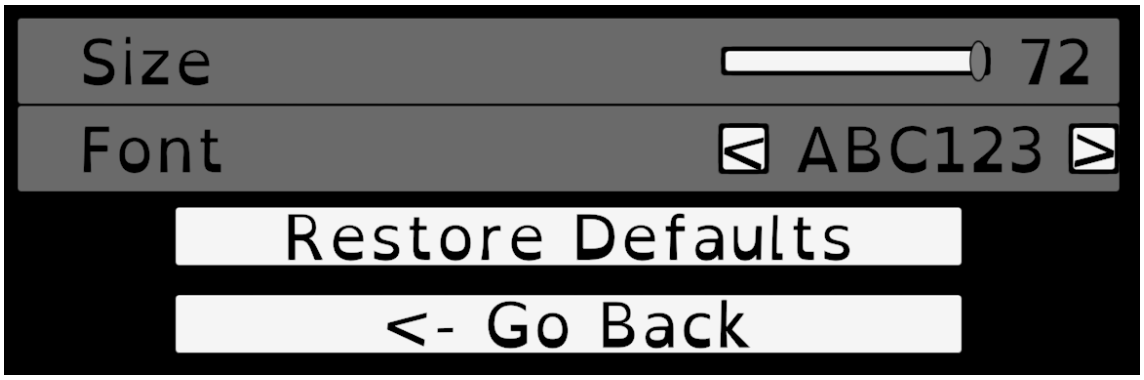


Figure 40: Text page.

Similar to the ball marker and reticle size pages, by adjusting the size of the text in the UI the player should be able to read information displayed more easily.

Text Font

In the final prototype the player can choose between sans serif or OpenDyslexic font. Sans serif[4] is a typeface commonly associated with online applications, as computers have been found to struggle rendering serifs properly. As for OpenDyslexic font, the characters are 'weighted' at the bottom so that readers can more easily interpret their orientation.

[4] 'sans' translates to 'without' in English.

7 Online Functionality

7.1 Overview

This section describes the tools used to facilitate the development of the game's online functionality, followed by an overview of its design and implementation process. The section then concludes with a discussion on some of the challenges faced during development.

7.2 Choice of Technology

With a game engine like Unity that has been around since 2005, a lot of support has been created to facilitate the development of online functionality. Different libraries/SDKs provide different ways to implement online functionality; in recent years, Unity have released a cloud-based gaming services program known as UGS. 3rd party services also exist, such as Photon Unity Networking[36], or independent packages which can be found on the Unity Asset Store.
I will be using UGS to implement the game's online functionality. I have two main reasons for making this decision: as a 1st party platform, UGS will provide tighter (easier) integration into Unity - in theory, this should speed up development time. In addition, Unity provides extensive documentation for the services I will be using within UGS. These services are:

- NGO
- Lobby
- Relay
- Vivox Voice and Text Chat

I will also discuss Multiplay, which offers dedicated server hosting for lobbies. There is also the additional benefit of being able to use the services for free, at least for prototyping/testing. Furthermore, tutorials on YouTube can provide support for implementation where necessary. I encourage the reader to learn more about UGS at the UGS homepage, which can be found at `https://unity.com/solutions/gaming-services` [Accessed 24th April 2024].

7.3 Netcode for GameObjects (NGO)

NGO is essentially a high-level networking library. All networks set up with NGO use a star topology (Figure 41), so all communications happen between the client and the server/host and never between clients directly.

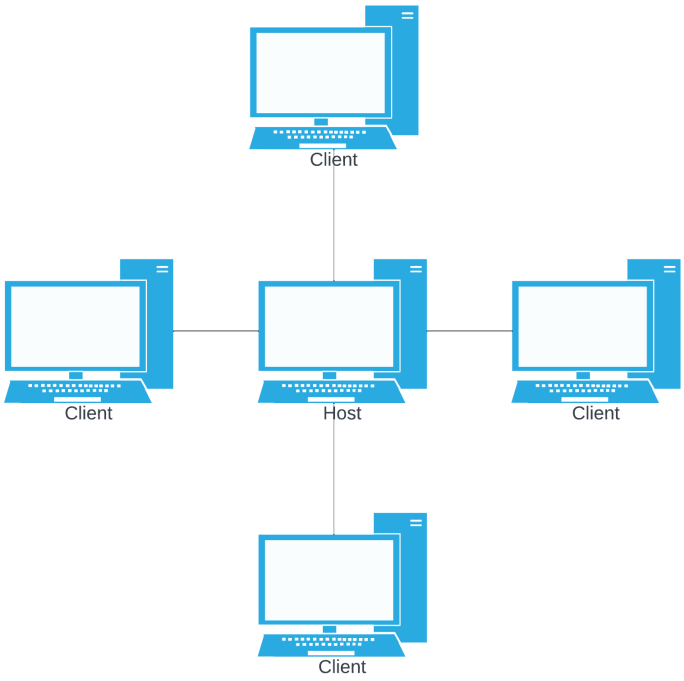


Figure 41: Representation of star topology involving five players, where one player is the host and four players are clients.

7.3.1 Remote Procedure Calls

A Remote Procedure Call (RPC) is a function in a script that can be called on objects that aren't in the same executable[33]. RPCs are always sent directly to the server, however their distribution over the lobby network is dependent on whether the RPC is a ServerRpc (Figure 42) or ClientRpc (Figure 43).

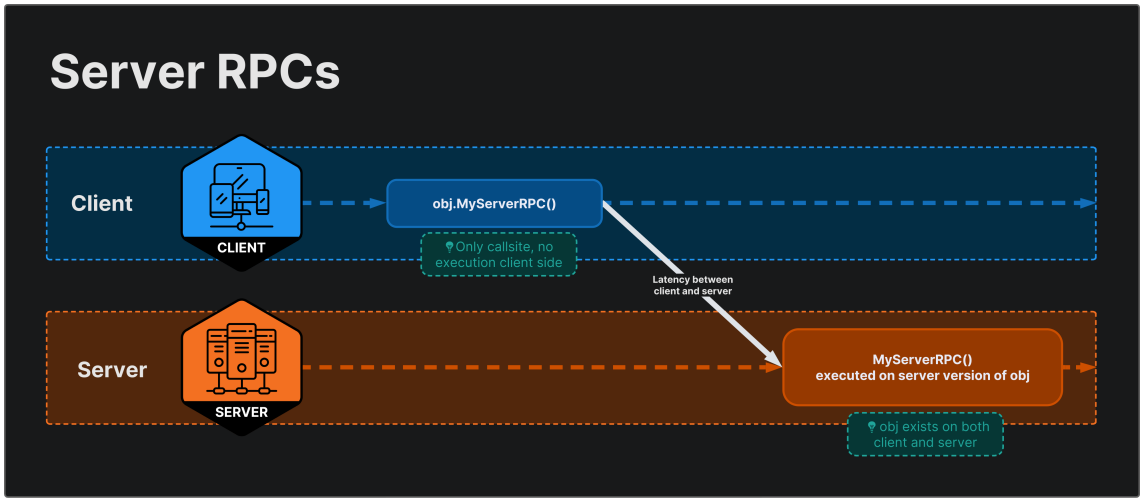


Figure 42: Concept of ServerRpc. Source of image: `https://docs-multiplayer.unity3d.com/netcode/current/advanced-topics/messaging-system/` [Accessed 30th March 2024]

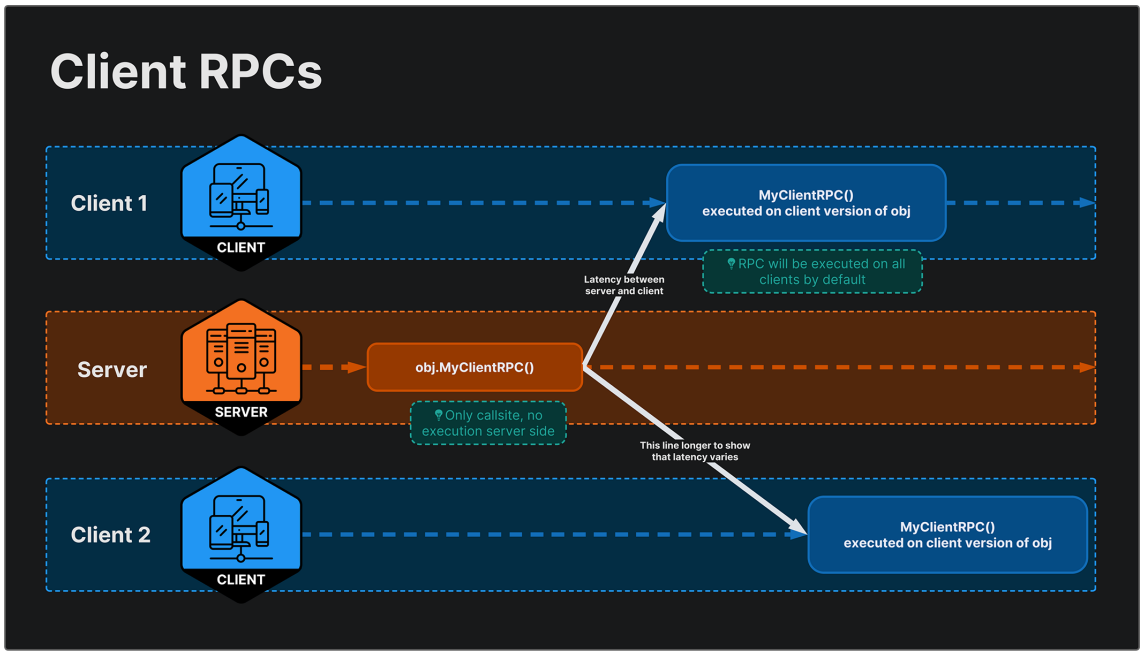


Figure 43: Concept of ClientRpc. Source of image: `https://docs-multiplayer.unity3d.com/netcode/current/advanced-topics/messaging-system/` [Accessed 30th March 2024]

For a ServerRpc, the method is only executed on the server, for the server's NetworkObject that is equivalent to the NetworkObject in the client.
For a ClientRpc, the method is sent to the server first. The server then distributes the call to all connected clients (alternatively, the developer can customise who the ClientRpc is distributed to).

7.3.2 Server vs Client Authoritative

An important decision to make in computer networking is whether to use a server authoritative or client authoritative framework. I will discuss both cases, then explain my solution.

Server

In a server authoritative framework, input actions on a client's machine are sent to the server: the server is responsible for updating the state of the game for all players based on these inputs. For example, if a client requests to perform the 'jump' action, an RPC is sent to the server. The server then sends an RPC to each player's machine with the functionality of the jump action embedded into the rpc.

Advantages: A client is less capable of modifying the functionality of input actions. This is because the client is essentially just mimicking the state of the server. Especially important in competitive games where if a client were to change the code in the game, they would be able to modify it to give themselves an unfair advantage.

Disadvantages: Very complex to implement without effecting a client's experience. For every input, the client must wait to receive the functionality from the server. This waiting time is formally known as Round-Trip Time (RTT), and can vary significantly between players connected to a network for a number of reasons (bandwidth speeds, geographical distance from server, etc.). This can lead to a poor experience with the game as on a client's side, their inputs will feel delayed. To combat this problem, developers

can implement a ghosting feature where a predicted outcome of an input is performed on the client, while the client waits for the correct outcome from the server.

Client

In a client authoritative framework, the functionality of input actions are first performed on the client, before a packet containing this functionality is sent to the server. The server then executes this functionality on the host/other connected clients.

Advantages: A simple method of syncing inputs from players.

Disadvantages: In contrast to a server authoritative framework, clients have the ability to modify the game on their side. For instances where players in a network can trust each other, or the game is co-operative rather than competitive, this wouldn't be regarded as a significant issue. However, when this is an issue, the player's experience can be negatively impacted as games become unfair.

Decision

In the final implementation, I have used a hybrid framework of client-side scripting for the PC's movement, and server-side for all other data that needs to be synced between clients - including the movement of the Ball. The ghosting feature was a challenging system to implement and I failed to find a solution that would suffice, although I recommend the reader to try themselves (if the reader wishes to develop a game similar to one in this project) as it is the most ideal solution for a competitive multiplayer game. I have left the evaluation of this decision to user testing (see 8.2).

7.4 Lobby & Relay

Lobby allows players to connect before or during a game 'session'[34], by letting players create lobbies that other players join. Lobby helps in that it establishes the network of players that can then share information on the NGO server (created by the host), to have the players connect using that information. Relay enhances this functionality by handling port forwarding, so that players do not need to modify their firewall - a technical challenge which is very inaccessible to players unfamiliar with this.

7.5 Vivox Voice and Text Chat

Vivox provides cross-play voice and text chat communication channels for online games[35]. This is a crucial service to competitive games where players need to communicate strategy in order to win matches. Vivox also has TTS input and output functionality, making the service accessible to players with visual and/or audio disabilities.

7.6 Design and Implementation

The following design and implementation would not be possible without the use of external guidance, particularly the *Learn Unity Multiplayer (FREE Complete Course, Netcode for Game Objects Unity Tutorial 2023)* YouTube video tutorial[37] (see reference). I highly recommend this resource to anyone who wishes to learn more about online functionality in Unity.

7.6.1 Design

Players will have the ability to create or find lobbies. When creating a lobby, the player can choose to make the lobby public or private, as well as the maximum amount of players (up to 10) and whether they want the game to be casual or competitive. If private, the player will have to provide a join code to other players so that they can join the lobby. If public, the lobby will be displayed in a list amongst other public lobbies.
When a player successfully joins a lobby, they enter an 'InLobby' state, where they are able to communicate with other players and customise settings whilst waiting for other players to join. Players should then have a way of stating they are 'ready' to play. When all players are 'ready', a game session will begin.
If the host disconnects during a game session, the game session will be 'abandoned' and the appropriate information should be displayed to inform all players.
When the game session is complete, the game will inform all players of the winning team.

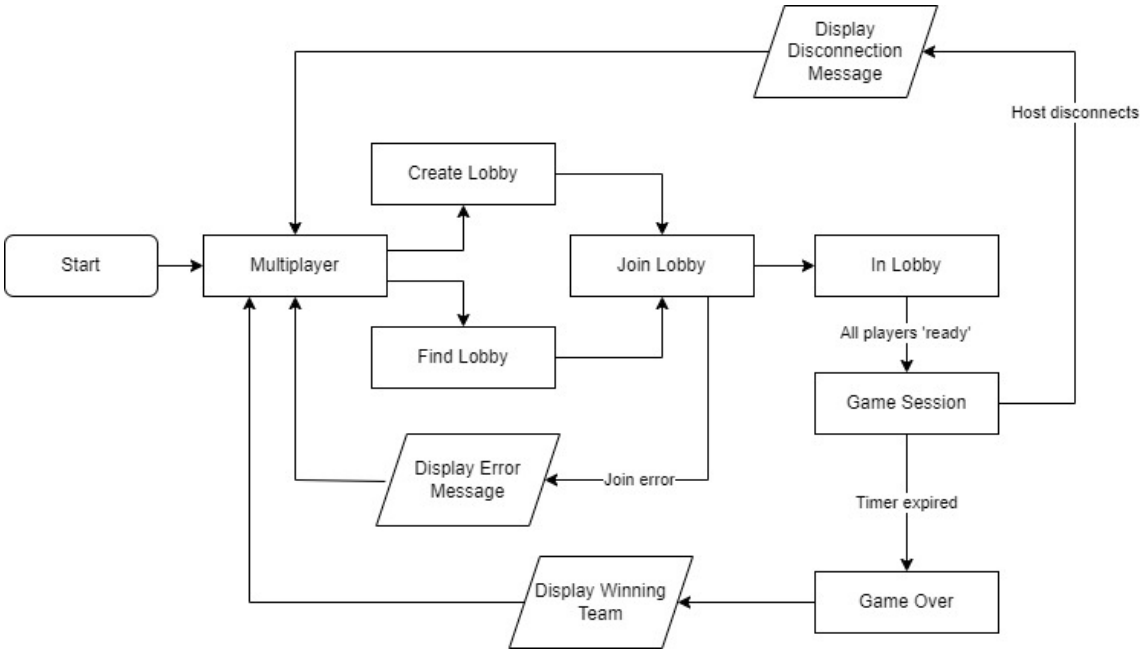


Figure 44: Flowchart showing player experience with online functionality.

7.6.2 Implementation

I will use this part of the report to describe the UX flow that can be found in the final prototype, describing the back-end functionality where necessary.

Lobby Menu

When the player chooses 'multiplayer' from the main menu, they are shown the lobby menu UI (see Figure 45). If public lobbies exist that are currently waiting for other players to join, then a selection of those lobbies will be displayed (see Figure 46).

Figure 45: Lobby UI joining lobbies, with an option to create a lobby. There are no public lobbies currently available in the image

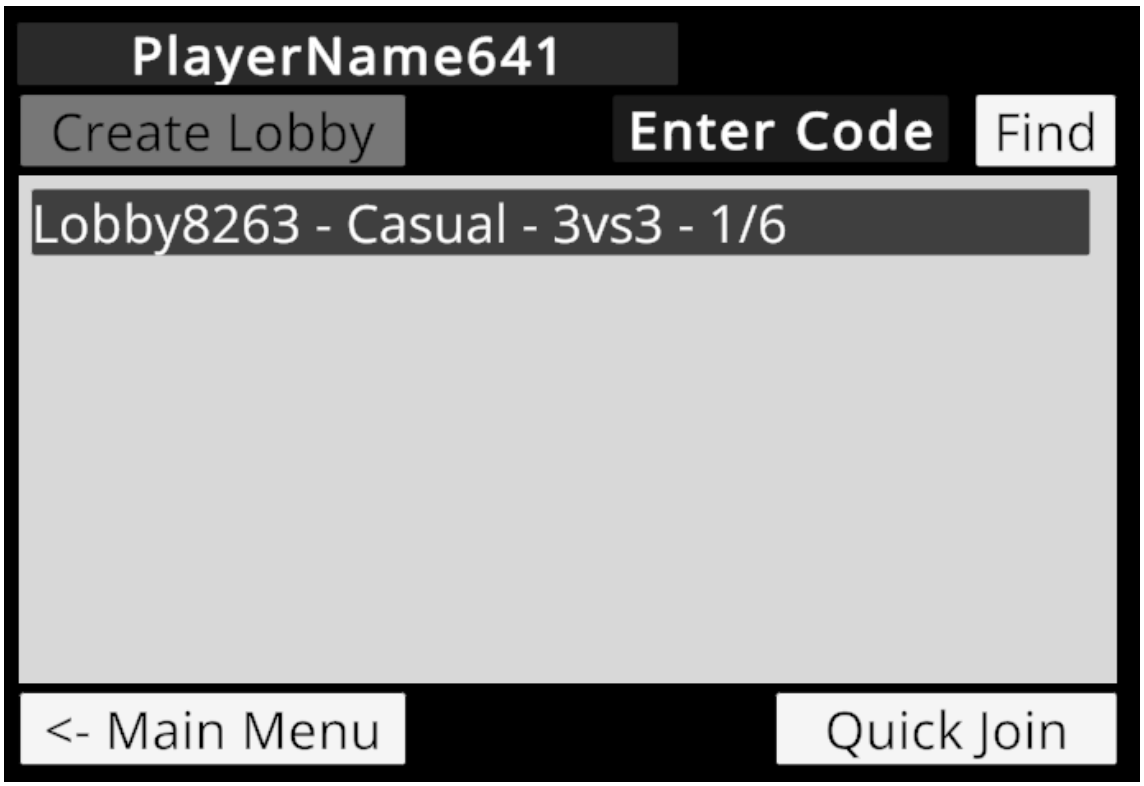


Figure 46: Lobby UI for joining lobbies, with an option to create a lobby. In the image is a public lobby available for players to join

Create Lobby

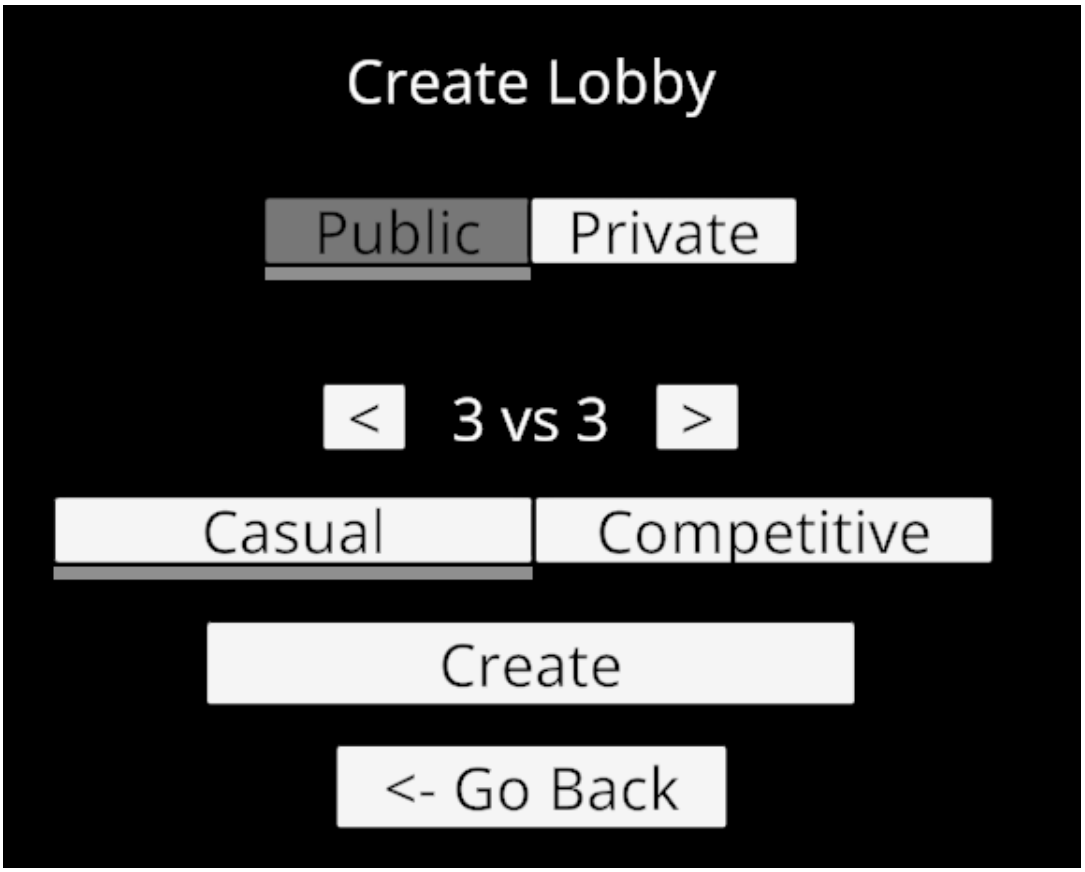


Figure 47: Create Lobby page.

When the player wishes to create a lobby, they are shown a page with a range of options (see Figure 47).

Find Lobby

As I have mentioned already, the player may find a public lobby in the lobby menu. Additionally, all lobbies have a 'join' code (which I will explain later); if a player is provided the code for a lobby, then they can enter it in the 'Enter Code' input field (see Figure 45) before pressing the 'Find' button.

There is also a 'Quick Join' button, which connects the player to a random public lobby.

Join Lobby

If the player creates a lobby, then the options selected in the creation process are passed as arguments (see Table 4) into a 'CreateLobby' function.

Variable	Type	Description
`lobbyName`	String	Name of the lobby. Set as "Lobby+(Random number between 0 and 9999, inclusive)".
`isPrivate`	Boolean	True: lobby is private. False: lobby is public.
`maxPlayers`	Integer	Maximum number of players allowed in the lobby. Absolute maximum value is 10 (for 5vs5 lobbies).
`difficulty`	String	Holds either "Casual" or "Competitive" value.

Table 4: Descriptions of Create Lobby function parameters.

In this instance, the player is set as the host of the lobby. Other players joining the lobby are considered clients.
Depending on whether the player has created or found a lobby, when they join it they are notified of this connection process by having the pages in Figure 48 displayed on their screen.

(a) Creating lobby. (b) Joining lobby.

Figure 48: 'Connection in progress' pages.

Figure 49: 'Could not join lobby via Quick Join' page.

If the connection process is disrupted, an event is invoked based on the context of the disruption. For instance, failing to connect via 'Quick Join' will display the message seen in Figure 49.

In Lobby

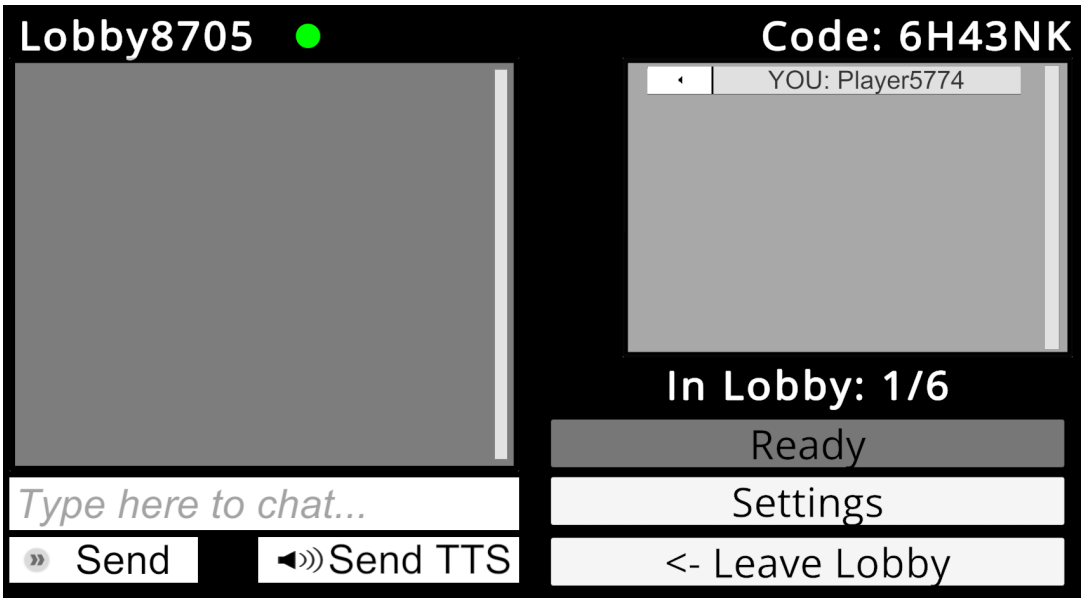


Figure 50: InLobby menu.

If the host has successfully connected, then the Vivox channel is created. This channel is attached to the lobby so that when clients connect, the game is able to direct the player's communication to the correct channel.
Finally, the player is able to view the InLobby menu (see Figure 50). Here they can enter and read text messages, as well as see the names of players and mute their voice if they wish. In the settings menu, an additional set of 'communication' settings can be found [by going to Audio $->$ Communication in the menu], to adjust features such as TTS, input and output devices and their corresponding volume (see Figure 51).

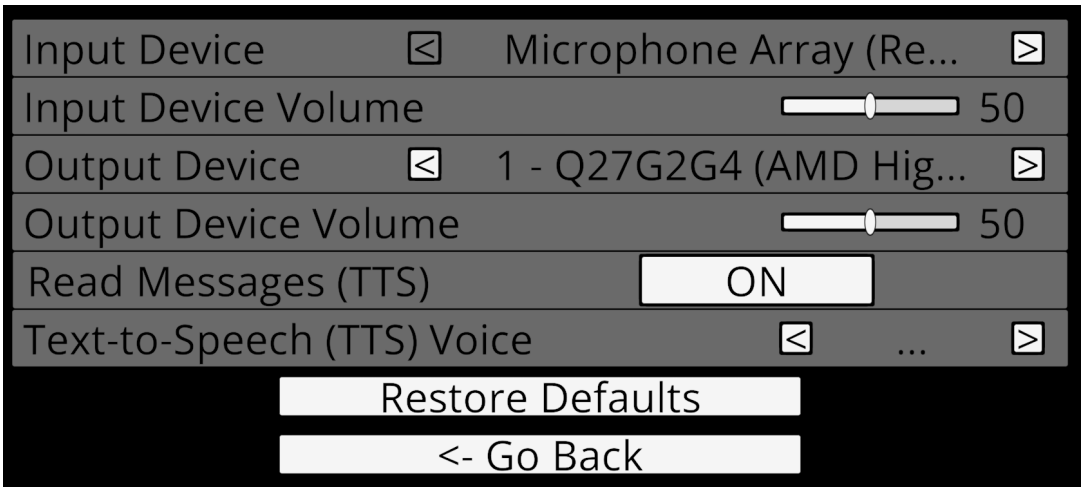


Figure 51: Communication settings page.

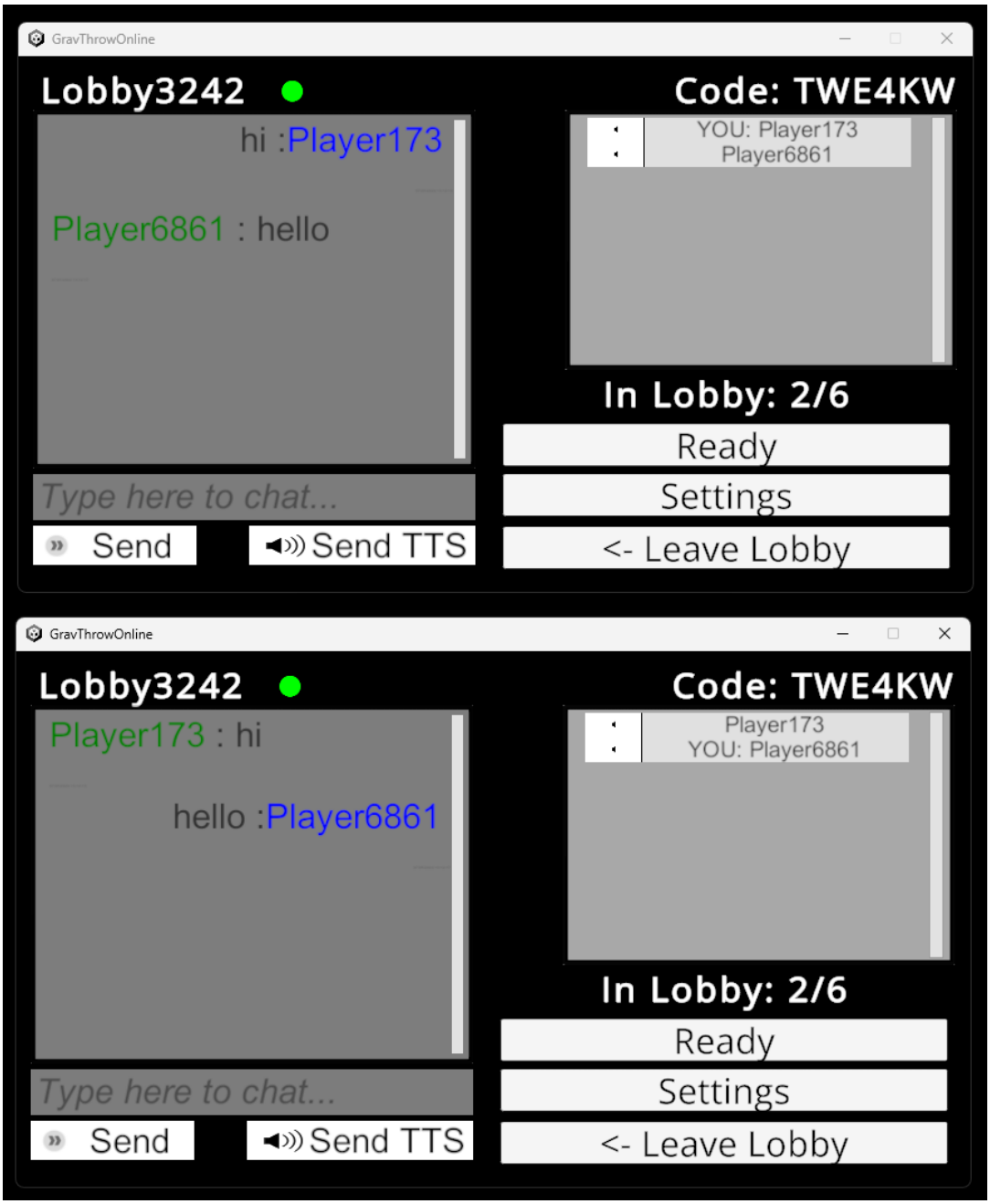


Figure 52: Two players connected to the same lobby.

If the player is ready to play, then they must press the 'ready' button.

Game Session

Once all players are ready, a game session can begin. This is where the players compete. Each player's PC is allocated a spawn point - this indicates whether the player is on team A or team B.

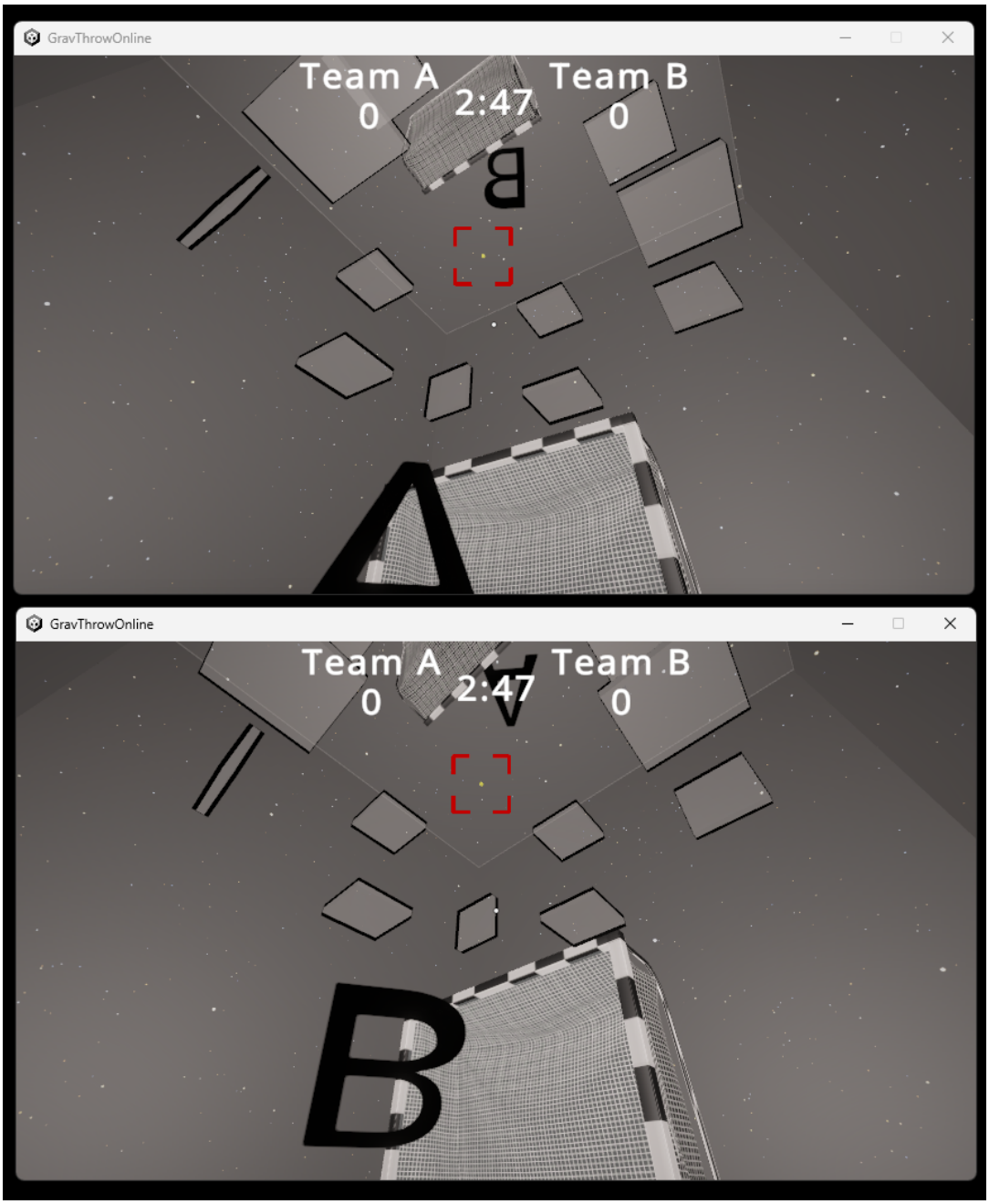


Figure 53: Two players in a game session.

The game session continues until either: (1) the host disconnects, in which case the session is abandoned (and the player is notified about this - see Figure 54), or (2) the timer expires, in which case the 'Game Over' event is invoked.
Players (including the host) have the option to leave the lobby. To ensure the player does not accidentally leave, a 'confirmation' window is shown (see Figure 55).

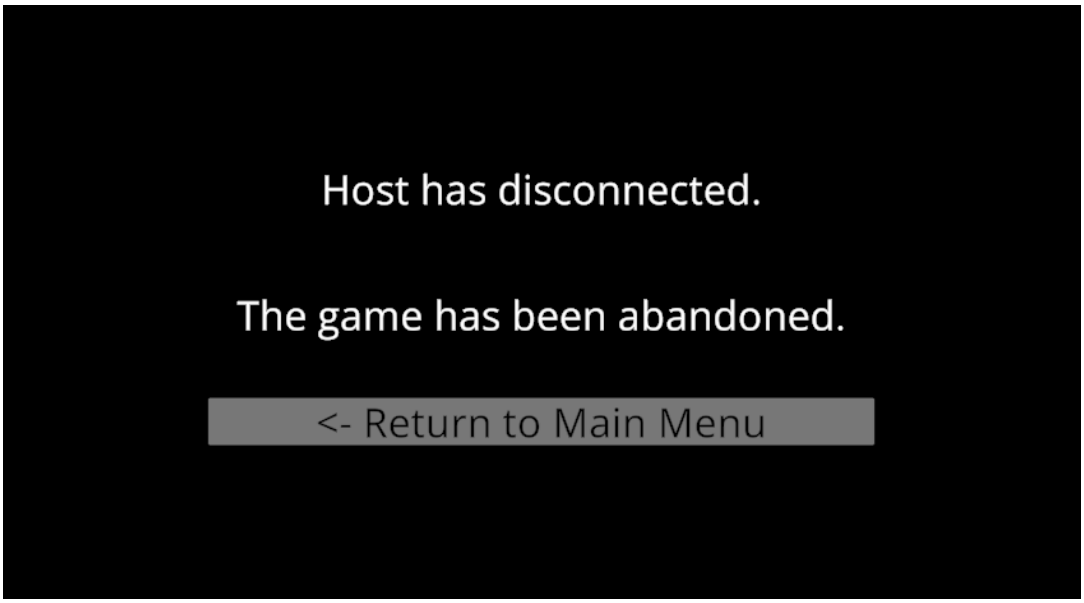


Figure 54: Host disconnected from game session.

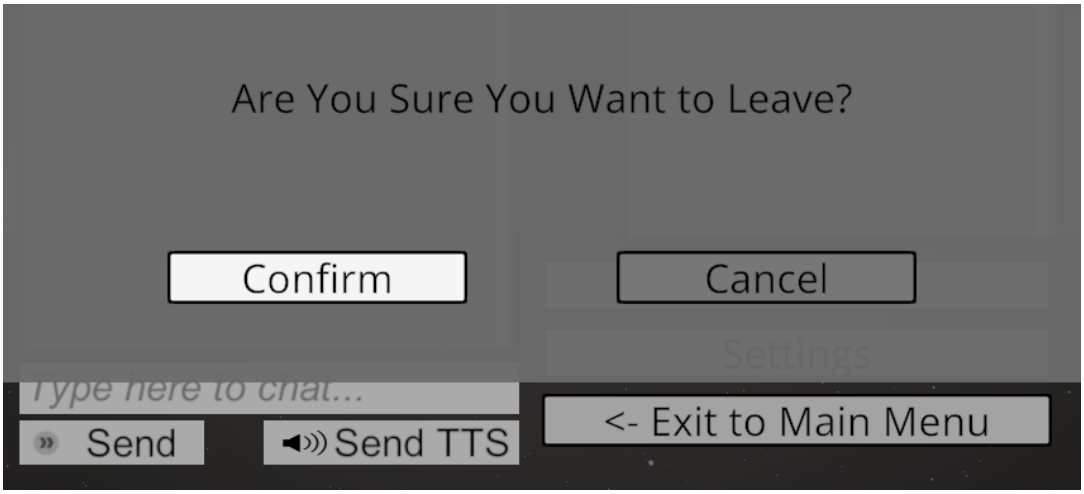


Figure 55: Confirmation window that appears for every destructive action in the game.

Game Over

In the game over event, all PCs are no longer able to move and the Ball is frozen. The game over page is displayed (see Figure 56), informing all players on which team won.

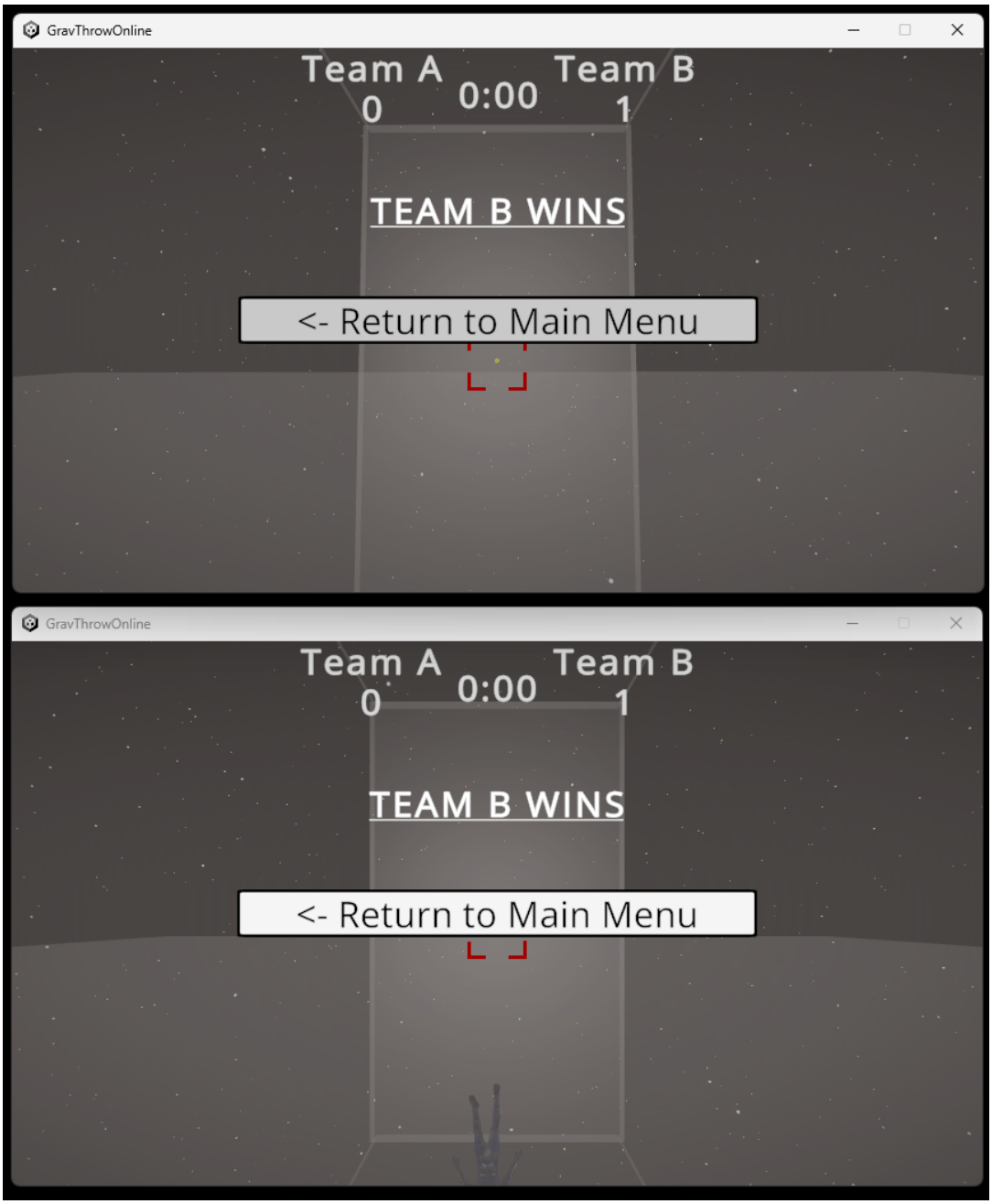


Figure 56: Game over page - appears on every player's screen. Here team B has one goal, whilst team A has zero goals: team B has won.

7.7 Challenges

System testing the online functionality was a time-consuming process, as it required building the game's executable every time in order to run separate instances. I had moments during development where fixing bugs, and checking if the fix worked, would create a cycle of 'fix, build, test', where each build process would take roughly 30 seconds to complete. This time mounted the more I tested, so my way of minimising the impact was to dedicate times in my schedule to listing bugs found in the game. I would then work on this list before the next time I tested the system. Ultimately, this is a fault of the version of Unity I chose, as Unity 2023 offers dual game windows inside the editor,

thus bypasses the need to build the game. Whilst I was aware of this when choosing which version of Unity to use, the recency of the 2023 version at the time of choosing meant that there was limited support for the suite of UGS tools mentioned earlier in this section.
I also found issues regarding the use of the Vivox service. Whilst the Vivox package came with a sample scene (with the voice and text chat technology already set up) that I could import into my game, the sample used Unity's legacy UI objects, whilst the interfaces I had already created used Unity's new UI objects. As a result, the adjustable 'text size' and 'text font' accessibility settings do not work on the final prototype. I had tried to modify the sample scene, however the functionality was too tightly integrated with the legacy objects that making changes to the setup would cause errors.
Vivox offers developers the ability to customise communication settings, and as such I was able to implement these settings in Figure 51. Whilst this improved the voice and text chat system from an accessibility standpoint, the settings could only be adjusted if the player was connected to a Vivox channel. In other words, players could not customise these settings directly from the offline version of the settings menu. Not a major problem, however it may confuse players who adjust their settings before playing and are unable to find these settings.

8 Testing and Results

8.1 User Test 1 - Player Character Mechanics and Settings UI

8.1.1 Introduction

The aim of this test was to measure how effective the game's tutorial is at teaching the game's controls - as well as measuring the intuitiveness of the controls and the usability of the settings UI.

8.1.2 Choosing Test Method

In outlining the requirements for the test, I had to consider how I would actually collect feedback from players; many user research methods exist which I could choose from. To prevent feeling overwhelmed with choice, I decided to shortlist the methods to choose from to those taught during my human-computer interaction module at university. These were

- Interviewing
- Focus groups
- Questionnaires
- Observations

These methods provide pros and cons which need to be weighed against each other to assure I am using the method that best suits this test.
Interviewing is the process of talking directly to a user tester. Interviews are either: structured, where questions (and the order which they are listed in) are pre-defined and there is no deviation; unstructured, where the lack of a list of questions allows for a more exploratory conversation, or; semi-structured, which is a middle ground between structured and unstructured (the interviewer will guide the direction of the interview, but follow-up questions/unplanned discussions may be allowed).
Focus groups involve organising a meeting with a group of testers, and asking questions to this group with the hope of gaining feedback based on multiple viewpoints at once.
Questionnaires are forms that user testers fill out. Questionnaires may be filled out before, during, and/or after the user performs the test, and usually consist of a variety of different question formats: 'yes' or 'no' checkboxes, open-ended questions, rating scales, and multiple-choice.
Observations involve watching testers use the software, recording information on what the tester does and how they behave. The observer may be active or passive in the test, either providing support for the tester and controlling the tester's experience, or leaving the tester independent to complete the test. Indirect observation can be taken too, where the user is asked to record their thoughts, ideas and experiences using a paper/digital, audio/video or photo diary.
Of these, I decided that creating a questionnaire would be the best approach. They are easy to create and distribute, and can be distributed online such that the data I collect is anonymous. That being said, I also allowed the test to be modified such that I could perform observational study too. I believe the main benefit of observation is the ability to record how the tester interacts with the system in real-time, which cannot be achieved by other methods.

8.1.3 Evaluation

Raw data for User Test 1 can be found at Appendix A. The response to the feedback was written in the sense that I would aim to fix some of the issues players had, as a means of iterative testing. However this evaluation was conducted too late into development for me to achieve this: instead the most important improvements will be left as future work.

Question 1
Unexpected results came from question 1 of the test. I had hypothesised that the class of tester defined in question 2, meant to indicate their experience with video games, would show relevance in the time taken to complete each level. This was not the case, however, as testers reported varying results. T8 and T9, who spend more hours per week playing games than other testers, appear to record times longer than other testers in most levels.

Question 2
No direct evaluation can be gathered from this question. Used to enhance evaluation of other feedback.

Question 3
No direct evaluation can be gathered from this question. Used to check if testers had issues directly based on their input device. Some testers experienced sensitivity problems when using the mouse.

Question 4
Used to check if testers had issues directly based on their operating system. From the feedback received it appears there were no errors of this manner, however only one tester used macOS with the rest using Windows. Will need to check myself if the game can run on Ubuntu.

Question 5
Appearing more than once in the feedback provided, the confusion for the moving platform's trail as a spike will need to be changed.
The idea of testing the sensitivity is interesting, however the player can test the sensitivity themselves by playing the game and going back into settings to adjust. Other testers have mentioned in other parts of the questionnaire that they had to make a significant adjustment from the default value.
I will need to check the reticle colour is saved when changing it in settings. As this is a significant accessibility feature, if the choice of colour doesn't save, I will need to fix it.
The idea of using a button on the gamepad in replacement of selecting the 'go back' interface button is interesting, and I can see how it would improve quality of life. Will look into this.
The mention of glitches in levels 7 and 3 will need to be investigated. I will try to replicate these issues, and fix them if possible.

Statement 1
It appears I have satisfied the goal for XAG 101. Testers were extremely positive about the ability to change text size and font, making the UI easy to use. One tester appears to have wanted more fonts, which is understandable as the game currently has just sans-

serif and dyslexic font options. Will look into this if I have the time after working on more serious improvements.

Statement 2

A near equal amount of positive and constructive points made here. Most testers appear to understand the purpose of different objects, however T10 chose 'Slightly disagree'.
One suggestion was to use colour to provide more pronounced details, which is an interesting point. I had wanted to go for a grey-scale palette so that players with colour-blindness have the same experience as other players, however for those other players they may have some trouble viewing the environment. Perhaps the solution is to look into the use of more contrasting textures that can be applied to the different objects in an environment.
One tester was confused about when they could change gravity. In the tutorial it mentions that transparent surfaces are the ones that allow the player to change gravity. However maybe I need to look into the use of a texture that clearly identifies which platforms allow you to change gravity. This improvement is given further justification when another tester mentions seeing the outline of transparent surfaces on the other side of solid walls. Outlining was used to make transparent surfaces easier to see, but using it has created an unintended side-effect.
Further mention of moving platform trail looking like spikes, so will definitely need to fix that.

Statement 3

Very positive feedback here. T6 said that sometimes realising what to do in a level wasn't obvious, however this appears to be an outlier given that other testers found they could understand what to do. T6 may be alluding to their performance (Figure 70) in level seven, where they had the longest time of over 120 seconds.

Statement 4

Testers expressed some UX issues when navigating the settings menu. I will need to improve the speed at which players can navigate the menu, and make the menu less cluttered. Perhaps I could improve speed by allowing players to perform the 'go back' action with a button on their input device.
T6 suggests changing 'reticle' to 'crosshair', which I can understand. But this is by far a serious issue and given time constraints it's better for me to focus on more important improvements.
Interesting point made by T7 where they thought the button that was 'selected' was, in their eyes, unclickable. Showing what button in the interface is selected is important for gamepad navigation (as well as for keyboard-only navigation, using the arrow keys). Perhaps I could solve this issue by having the game check if the player is using the mouse. If they are, then don't select a button (or any other selectable interface object).

Statement 5

Mixed experience here. On the constructive side, T2 provides the same point as T7 in A. Will need to fix this issue.
May look into the use of dropdown menu style however won't make this a priority. Not sure what slider T4 is referring to when talking about the screen resolution, as the settings choice template (Figure 28) was used for that setting.

Statement 6
Will need to improve camera sensitivity settings based on this feedback. Adjusting camera settings appears to be 'easy' and 'intuitive' for testers, however T8 had a problem with the camera settings being part of the Controls setting type. Could create a new setting type specifically for Camera if I have the time.

Statement 7
Not much to comment on here, but mostly positive feedback is an indication that I have satisfied XAG 114. However some testers made reference to previous feedback in earlier parts of the questionnaire that also has relevance here, so can't say for certain that XAG 114 is fully achieved.

Statement 8
From the feedback here, I feel confident that the tutorial was, for the most part, successful.
Interesting point from T2 about the tutorial not 'feeling' like a tutorial. Whilst it is clear they have written this as constructive feedback, I would say that this was my intention for the tutorial. I wanted the player to learn about the mechanics of the game, whilst also giving the player freedom to achieve the objective of each level on their own.
I'm not sure what T9 is referring to about the the tutorial text 'staying' on-screen for longer, as the tutorial text is always on-screen during the tutorial (except when the player is in settings menu), so can't comment on this feedback.

Statement 9
Positive feedback from all testers here. T7's suggestion of an extra mechanic to add to the PC is interesting, but I most likely won't explore this further. Overall I feel confident saying that players were comfortable with the game's controls.

Question 6
Good to see testers enjoyed the concept of the game, as well the mechanics of the PC. Not sure what T7 is referring to about playing 'after the tutorial' as there was no additional content.

Question 7
Will look into making quality of life improvements.
Will look into improving the jump mechanic. Hard to gauge the importance when other testers have said they enjoyed the feel of the PC's mechanics, but fortunately T4 provides enough information on where specifically the jumping mechanic needs to be improved.
Mention of improving graphics and using colour in the environment.
Will definitely change the transparent surfaces as T8 mentions problems with how they look and is something I agree with.

8.1.4 Reflection

Now I have to mention, the only way the timer itself can be stopped (aside from completing the objective of the level) is by pausing the game. One case that I hadn't considered was the possibility of testers reporting long times because they were occupied with other

activities whilst performing the test. Naively, I had not considered this possibility, and upon reflection of the test I realise that this could very well be the reason for outlying results.
Also on reflection: question 2 provides a limited insight into the experience of testers *with games similar to the one being developed in this project*. The question itself is too general in the sense that a tester may have a lot of experience with, say, the mechanics of a 2D tower defence games, but little/no experience with those of a 3D first-person game. This could mean that the testers who chose 5-10 hours or $>$10 hours may not have, in reality, spent much time playing games with similar mechanics found in 3D first-person games.

8.2 User Test 2 - Online Functionality

8.2.1 Introduction

The aim of this test was to measure the stability of lobbies as well as the usability of the lobby setup process (i.e. creating, finding and joining lobbies). The test involved organising a date and time for getting testers to create and join lobbies. I had managed to recruit 10 testers, however 2 were unable to participate, meaning I could not test the functionality of a lobby at maximum capacity. Nonetheless, the test provided enough qualitative data to perform an evaluation of the game's online functionality.

8.2.2 Choosing Test Method

8.2.3 Evaluation

Raw data for User Test 2 can be found at Appendix B.

Question 1
Here we see that we have a range of testers with experience in similar games. The question is identical in purpose to Question 2 of User Test 1 (8.1)

Question 2
Only keyboard and mouse was used by testers, which is an issue as we cannot evaluate the usefulness of the gamepad as an input device for online functionality. I had emphasised to the players prior to the test that they are free to choose their preferred input device (as a means of usability).

Question 3
All testers had to use Windows after discovering that Vivox could not run on macOS computers with an Apple Silicon processor. There were two testers with this issue, who fortunately owned a Windows computer and thus could continue to participate in the test.

Statement 1
Very positive feedback in relation to navigating the UI. The mention of everything having 'obvious labels' by one tester is an indication that XAG 114 has been followed correctly.

Statement 2
Positive feedback here, although testers had mixed feelings about the highlighted button remaining selected despite using keyboard and mouse.

Statement 3
A major UX problem occurred during this test: testers were unable to join lobbies if they had created one and then deleted it, or if they had completed a game session. We discovered that the issue can be resolved if you restart the game's executable, which is how we proceeded for the remaining sessions in the test. This was disappointing discovery, although one could make the argument that without this issue, the process of finding and joining lobbies works as intended, as can be seen by the positive feedback. Nonetheless, the overall response to this statement was negative, and will need to be fixed.

Question 4

Asking testers to provide only positive feedback, a range of responses were given. From enjoying the simplicity of the setup process, to the usefulness of joining public lobbies in the event that a player is playing on their own. There were comments about the text chat, however when writing this question I wanted to know what players thought of creating, finding and joining lobbies - not what they like once they are in a lobby. May have been better to specify this in hindsight as more relevant data could've been collected.

Question 5

Interesting to see what testers have responded with here. One tester has mentioned turning off TTS, which is actually available in the settings menu whilst playing online. However, this was not told to the testers as I had wanted to see if they would instinctively use settings. I think this brings up a useful point about informing players on what they have control over - perhaps players believe a feature can't be changed because of experience with other games where this wasn't an option.
Another tester mentions providing more visual feedback whilst connecting, with a progressive loading bar of spinning icon animation - this would certainly be a useful addition as it prevents confusion from players who may believe that their computer has frozen.

Statement 4

Overall, the feedback here was positive. Some testers state the HUD was easy to see, whilst others think it blended into the background as they share the grey-scale colour palette. Perhaps a high contrast colour option would be a suitable solution to this issue.

Statement 5

Mixed feedback for the usability of the voice and text chat system. When it worked, it was effective and easy to use. However other players reported issues of not knowing to use it in-game. One theory I have for this is that the text chat is in the pause menu, which may not have been a great decision in reality as it is unconventional. Usually, games will have a separate screen for voice and text chat.

Statement 6

The response in this statement is the most critical out of all in this test. I had thought that the spawn location would indicate which team the player was on, however I had considered the possibility of players forgetting as they move around the arena.
So the player needs to know two pieces of important information: what team they are on, and who their teammates are. The HUD could display the letter of the team that a player is on, and show an icon as a world canvas image above PCs to indicate their team.

Statement 7

Not much to discuss for this statement, as I had only recruited 3 testers who weren't part of User Test 1 to see if the controls of the game can be learnt without requiring the tutorial. Testers of this category responded positively, stating that the simple mechanics and controls made it quick and easy to learn. Similarities with other first-person games also meant they could transfer their experience.

Question 6

Asking players what they enjoy about the game is a good indicator of what systems and ideas in the game are working as intended. One tester mentions the fluidity of camera rotations (I assume this is when they change gravity), and another tester enjoying the game's sound effects.

Question 7

Lots of useful feedback came from this final question. One tester wishes for increasing the size of the PC's visual as a way of improving the visibility of teammates. I would also say that the textures of the PCs could be improved, as the ones I had chosen did not contrast strongly enough with the arena.
Another tester suggests making it easier to save shots. Whilst this was my intention - a challenge of the game is in knowing when to move to save shots - I think future versions of the game could have a more advanced PC with mechanics such as sprinting, flying, wall-running etc. to solve the issue that this tester has mentioned.
A point about using colour to highlight important information was mentioned, and may be a consideration as future work. Perhaps in hindsight using colour to identify importance is a useful design practice - but games should still ensure that if this is used, players with colourblindness are able to navigate the interface and play the game without frustration.

8.2.4 Reflection

Feedback was generally positive, however problems with the lobby setup process led to a less-than-ideal experience. After the test had finished, I tried to identify the root of the issue mentioned in **Statement 3**, where testers had to restart the game if they wanted to play more game sessions. It appears to be a bug with connecting the player to UGS, as the issue can be found in multiple discussion forums - see `https://forum.unity.com/threads/http-1-1-401-unauthorized.1313526/`[Accessed 8th May 2024] as an example. This is unfortunate as it means there isn't much I can do to solve the problem except to use another networking SDK (which is impossible at this stage as the project is too far into development).

9 Conclusion, Thoughts and Future Work

9.1 Overview

In this section, I highlight the results of the project, then provide my personal thoughts on how well the development went. Following this I discuss potential improvements to the game based on feedback from User Test 1 (8.1) and User Test 2 (8.2).

9.2 Results

To conclude, I have created a competitive online multiplayer game with explicit and implicit accessibility features. However, when observing testers I sometimes noticed them not using the settings menu - despite the fact that by doing so, it would improve their experience. As an example, one tester had complained that their camera sensitivity was too high, which could've been lowered by modifying the sensitivity in settings.
For the testers that did use the settings menu, the majority responded positively to its usability and quantity of settings available.
The majority of the requirements set out in 2.3 have been achieved. One requirement not achieved is the development of an AI player agent, which could be used in offline matches. This was simply too ambitious, as the project already focused on accessibility and online functionality. Nonetheless, I would consider this an important component of the game to be added, if future versions of the game were to be developed.

9.3 Personal Thoughts

Whilst this project has given me a greater understanding of the importance of accessibility (as well as how to implement it correctly), and an awareness of the challenges that come with online functionality, I do believe that focusing on two significant areas of game development have led to a prototype which could've been more polished had I chosen to dedicate this project to only one of the two.
Furthermore, the order in which I had focused on parts of the project could've been planned better. Development was split into three distinct segments, and completed in this order: (1) PC mechanics, (2) online functionality, (3) accessibility settings. I had originally chosen to research online functionality before accessibility settings as I was unable to gauge the amount of time I would need to understand, design and implement it. The issue with this is that I had planned to test the PC mechanics alongside the accessibility settings, but it turned out that the settings took the longest amount of time to complete due to the re-design. When I finally had a high fidelity prototype which could be tested, I was too far into development to spend time making improvements on feedback and re-testing. This was the most disappointing part of development as iterative testing is a core principle to building accessible games.

9.4 Future work

The suggestions for future work have already been written in some detail in the evaluations of User Test 1 and User Test 2, however here I will highlight what I believe to be a few of the most necessary improvements. This excludes fixing the problem with players

unable to create/join lobbies unless they restart the game, as I concluded that to be an external issue with the Lobby service.

9.4.1 Settings Menu Quality of Life

The main complaint about the settings menu available to players in testing was the number of screens/pages players had to navigate to get to the settings they wanted to change. To solve this issue, a button on their input device could be set to perform the 'go back' action whilst on a page.
In addition, the accommodation for large font size meant that XAG 114 - UI Context (which suggests that players should be able to understand the purpose of UI components) could not be fully achieved in the final prototype - the limited screen space prohibited the addition of text informing the player on the meaning of each setting. On reflection, perhaps the background of each setting could be a selectable object (Figure 57) which has an action tied to displaying more information (Figure 58).

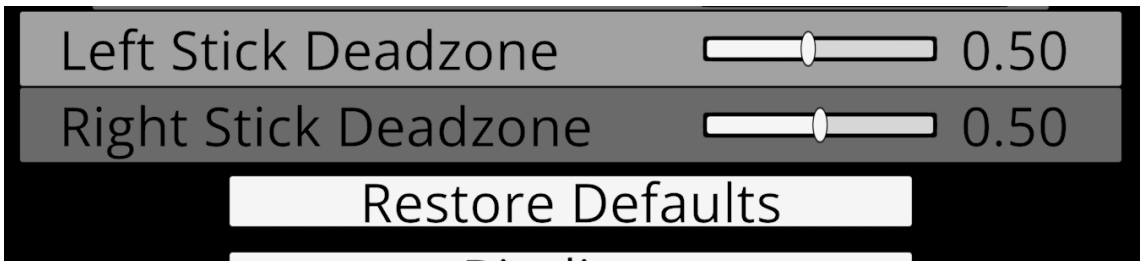


Figure 57: Right stick deadzone 'more information' selected - seen by the darker colour compared to non-selected.

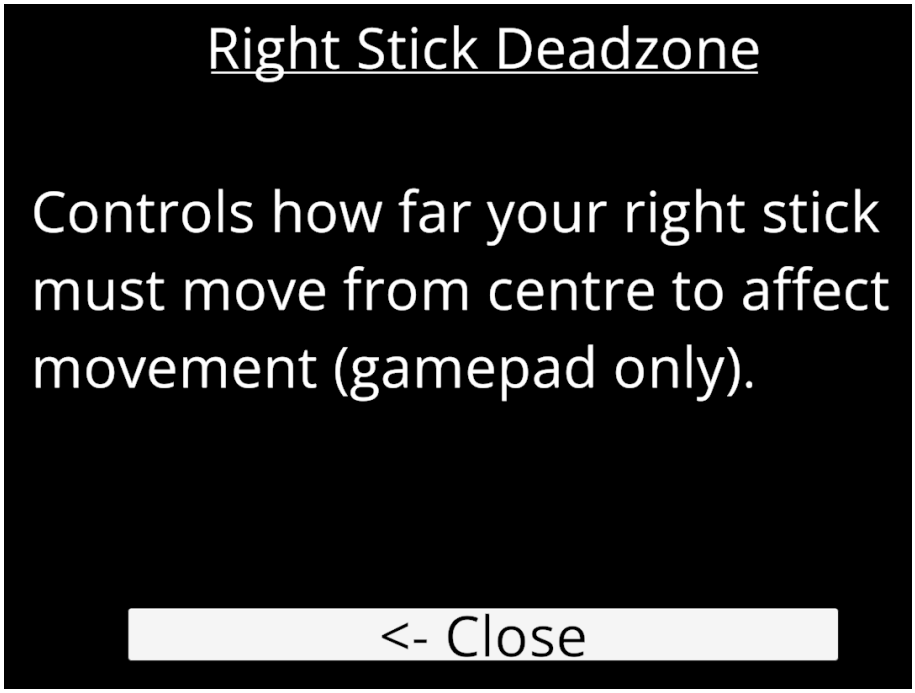


Figure 58: More information page for right stick deadzone setting.

Some testers experienced confusion with the 'currently selected' interface button/slider

being a darker shade than others: this could be solved by using non-colour options, such as Figure 59.

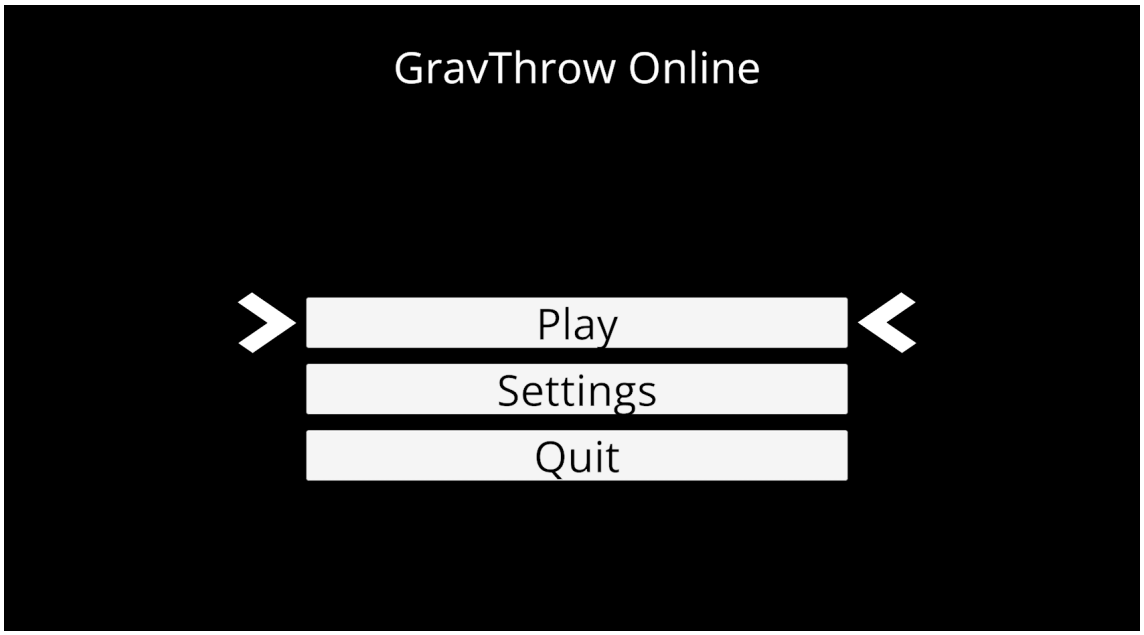


Figure 59: Re-design of interface showing the player what the 'currently selected' object is.

9.4.2 Textures

It is very important that players understand which surfaces in the game enable them to change their PC's direction of gravity. Choosing contrasting textures would help players identify different objects/surfaces in the game. However to choose the correct textures we would need to conduct iterative testing as this is subjective to the player. Perhaps the game could offer alternative textures to provide more flexibility.

9.4.3 Identifying Team

Players need to know which team they are on, as well as who their teammates and opponents are. One possible solution is to display the letter of the team that a player is on using the HUD. To identify other players, a world canvas image showing 'A' or 'B' could be placed above each PC, with a slider setting allowing players to adjust the size of this image to suit their vision.
One possible solution is to display a table showing which players are in team A and team B (Figure 60). By doing this, the player is aware not just of the team they are on, but also who their teammates and opponents are.

Figure 60: Possible design for displaying which players are in each team. The team that the player is on will also be underlined, as can be seen at the top of the HUD (in this instance, the player is on team A).

Additionally, the player could identify which team other players are on by there being a world canvas image showing 'A' or 'B' above each PC. For accessibility, there could be settings which allow players to adjust the size, font, and colour of these images.

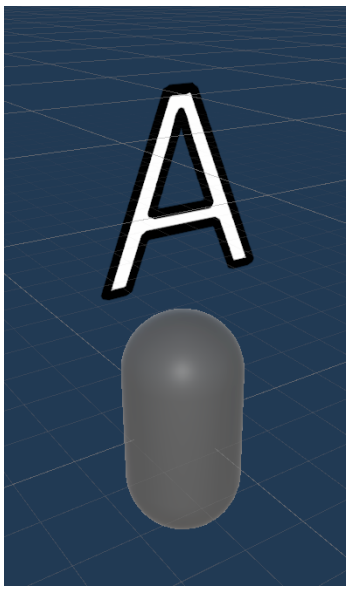


Figure 61: Letter above PC to identify which team they are on.

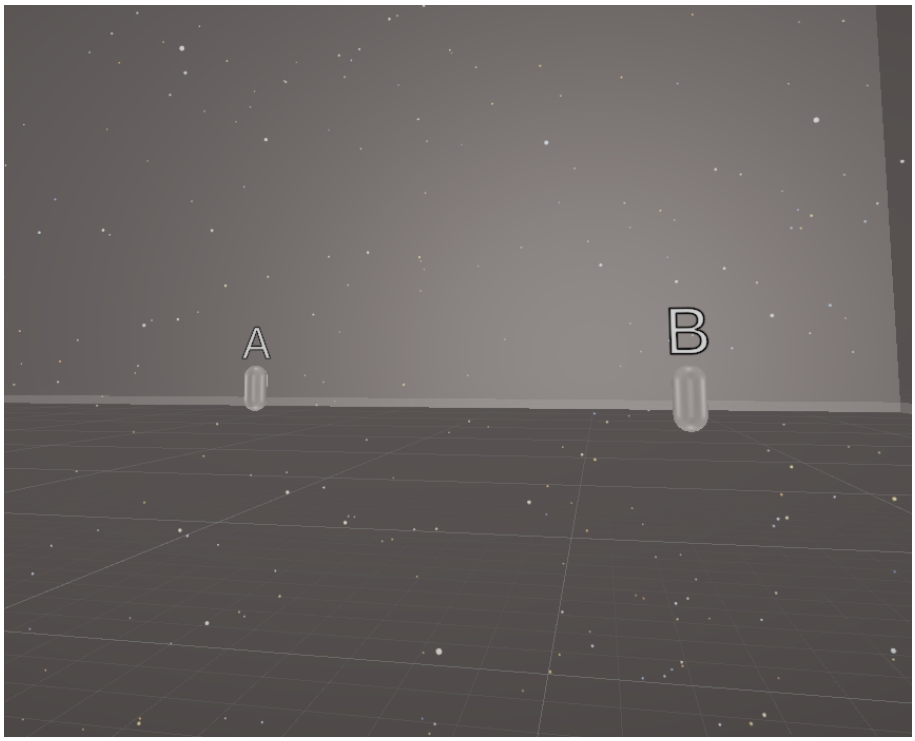


Figure 62: Letter above PC to identify which team they are on - in-game.

9.4.4 Matchmaking

In the game, players can choose whether to make a lobby 'casual' or 'competitive' when creating a lobby. For a 'competitive' lobby, the players are supposed to play with the sole intention of winning the match. As players improve their skill over time, they may end up being connected to other players with less experience. This is clearly an accessibility issue, as the weaker players have an unfair disadvantage. To solve this, we can take inspiration from existing eSports games that use a Matchmaking Rating (MMR) system. MMR is a numerical value indicating the experience of a player. If a player wins a game, their MMR increases (and vice versa). Those with similar MMR values are matched against each other - in other words, players with more experience and matched against players of the same calibre.
We could implement this system into the game by allowing players who create a 'competitive' lobby to set a skill range. This setup would then be displayed for other players to see, as in Figure 63. Players outside of this skill range would be unable to join.

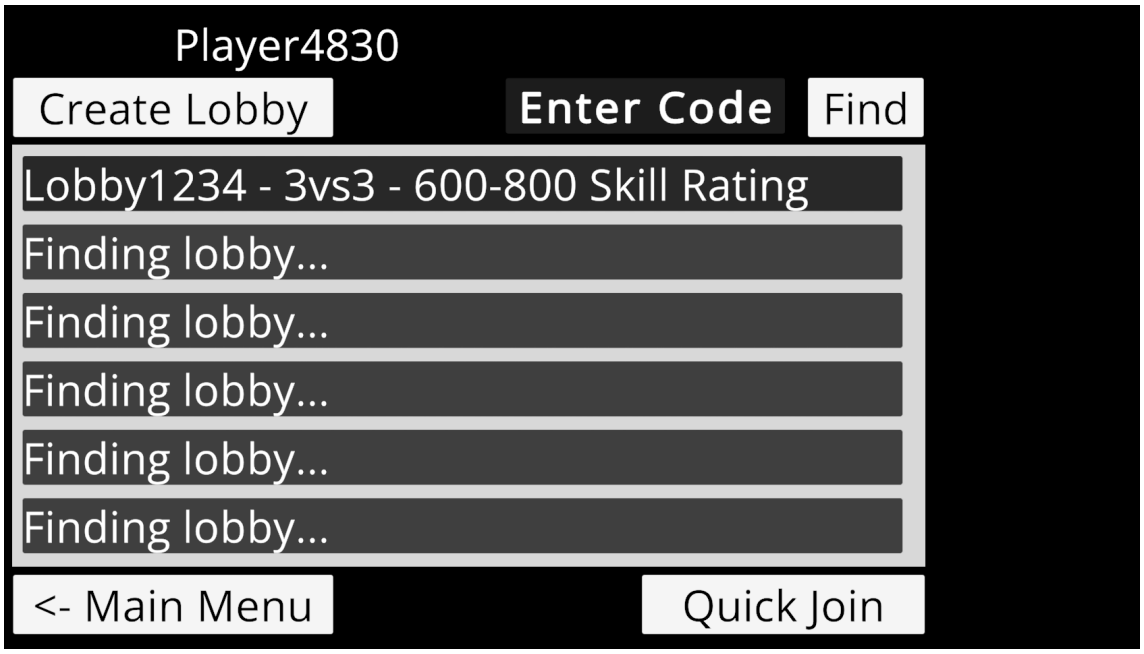


Figure 63: Showing MMR range for a public 'competitive' lobby.

To facilitate this feature, UGS offers a Matchmaker service. Documentation on this service can be found at `https://docs.unity.com/ugs/manual/matchmaker/manual/matchmaker-overview` [Accessed 7th May 2024].

References

[1] Nystrom, Robert (2011). Singleton · Design Patterns Revisited · Game Programming Patterns. [online] Available at: https://gameprogrammingpatterns.com/singleton.html. [Accessed 11th November 2023]

[2] The House of Staunton (n.d.). History of Chess [online] Available at: https://www.houseofstaunton.com/history-of-chess#:~:text=People%20have%20been%20playing%20chess. [Accessed 6th May 2024]

[3] HelpGuide (n.d.). The Benefits of Play for Adults - HelpGuide.org. [online] Available at: https://www.helpguide.org/articles/mental-health/benefits-of-play-for-adults.htm#:~:text=of%20electronic%20gadgets.-,Play%20helps,-%3A [Accessed 11th November 2023].

[4] CIVIC UK. (2022). History of accessibility in gaming. [online] Available at: https://www.civicuk.com/blog-item/history-accessibility-gaming#:~:text=So%2C%20what%20was%20the%20first%20game%20made%20inclusive%20on%20purpose%3F [Accessed 11th November 2023].

[5] Video Game Sales Wiki. (n.d.). Dark Souls. [online] Available at: https://vgsales.fandom.com/wiki/Dark_Souls#:~:text=Dark%20Souls%20is%20an%20action. [Accessed 6th May 2024].

[6] kaitlynjones (n.d.). Gaming accessibility fundamentals - Training. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/training/paths/gaming-accessibility-fundamentals/ [Accessed 11th November 2023].

[7] World Atlas (n.d.). The Most Popular Sports In The World. [online] . Available at: https://www.worldatlas.com/articles/what-are-the-most-popular-sports-in-the-world.html [Accessed 26th March 2024]

[8] kevinasg (n.d.). Xbox Accessibility Guidelines - Microsoft Game Dev. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/gaming/accessibility/guidelines [Accessed 11th November 2023].

[9] Nielsen, J. (1994). 10 Heuristics for User Interface Design. [online] Nielsen Norman Group. Available at: https://www.nngroup.com/articles/ten-usability-heuristics/. [Accessed 6th May 2024].

[10] Arts, E. (2019). Apex Legends Accessibility Resources For PC - An Official EA Site. [online] Electronic Arts Inc. Available at: https://www.ea.com/able/resources/apex-legends/pc [Accessed 11th November 2023].

[11] Unity Technologies (2019). Unity. [online] Unity. Available at: https://unity.com/ [Accessed 11th November 2023].

[12] Epic Games (n.d.). Unreal Engine — The most powerful real-time 3D creation tool. [online] Unreal Engine. Available at: https://www.unrealengine.com/en-US [Accessed 11th November 2023].

[13] Godot Engine (2019). Godot Engine - Free and open source 2D and 3D game engine. [online] Godotengine.org. Available at: https://godotengine.org/ [Accessed 11th November 2023].

[14] create.unity.com. (n.d.). Unity Gaming Report. [online] Available at: https://create.unity.com/gaming-report-2022 [Accessed 11th November 2023].

[15] Unity Learn. (n.d.). Practical Game Accessibility. [online] Available at: https://learn.unity.com/course/practical-game-accessibility?uv=2021.3 [Accessed 11th November 2023].

[16] Technologies, U. (n.d.). Powerful 2D, 3D, VR, and AR software for cross-platform development of games and mobile apps. [online] unity.com. Available at: https://unity.com/pricing#plans-student-and-hobbyist [Accessed 11th November 2023].

[17] Technologies, U. (n.d.). Unity - Manual: System requirements for Unity 2019.3. [online] docs.unity3d.com. Available at: https://docs.unity3d.com/Manual/system-requirements.html [Accessed 11th November 2023].

[18] docs.unrealengine.com. (n.d.). Accessibility. [online] Available at: https://docs.unrealengine.com/5.0/en-US/accessibility-in-unreal-engine/ [Accessed 11th November 2023].

[19] docs.unrealengine.com. (n.d.). Networking and Multiplayer. [online] Available at: https://docs.unrealengine.com/5.2/en-US/networking-and-multiplayer-in-unreal-engine/ [Accessed 11th November 2023].

[20] docs.unrealengine.com. (n.d.). Hardware and Software Specifications. [online] Available at: https://docs.unrealengine.com/5.0/en-US/hardware-and-software-specifications-for-unreal-engine/ [Accessed 11th November 2023].

[21] Unreal Engine. (n.d.). Unreal Engine (UE5) licensing options. [online] Available at: https://www.unrealengine.com/en-US/license [Accessed 11th November 2023].

[22] docs.unrealengine.com. (n.d.). Introduction to Blueprints. [online] Available at: https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/GettingStarted/ [Accessed 11th November 2023].

[23] Engine, G. (n.d.). Showcase. [online] Godot Engine. Available at: https://godotengine.org/showcase/ [Accessed 11th November 2023].

[24] Godot Engine documentation. (n.d.). High-level multiplayer. [online] Available at: https://docs.godotengine.org/en/stable/tutorials/networking/high_level_multiplayer.html [Accessed 11th November 2023].

[25] store.steampowered.com. (n.d.). Godot Engine on Steam. [online] Available at: https://store.steampowered.com/app/404790/Godot_Engine/#:~:text=not%2Dfor%2Dprofit.-,SYSTEM%20REQUIREMENTS,-Windows [Accessed 11th November 2023].

[26] Engine, G. (n.d.). License. [online] Godot Engine. Available at: https://godotengine.org/license/ [Accessed 11th November 2023].

[27] Godot Engine documentation. (n.d.). GDScript basics. [online] Available at: https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_basics.html [Accessed 11th November 2023].

[28] Studios, S. (2021). What is a Unity GameObject, and How Do You Fit It Into Your Game? [online] Starloop Studios. Available at: https://starloopstudios.com/what-is-a-unity-gameobject-and-how-do-you-fit-it-into-your-game/. [Accessed 24th April 2024]

[29] Unity Technologies (2019). Unity - Manual: ScriptableObject. [online] Unity3d.com. Available at: https://docs.unity3d.com/Manual/class-ScriptableObject.html. [Accessed 24th April 2024]

[30] Technologies, U. (n.d.). DOTS - Unity's new multithreaded Data-Oriented Technology Stack. [online] unity.com. Available at: https://unity.com/dots. [Accessed 24th April 2024]

[31] Unity Technologies (2019). Unity - Scripting API: PlayerPrefs. [online] Unity3d.com. Available at: https://docs.unity3d.com/ScriptReference/PlayerPrefs.html. [Accessed 25th April 2024]

[32] Dave / GameDevelopment (2022). FIRST PERSON MOVEMENT in 10 MINUTES - Unity Tutorial. [online]. Available at: https://www.youtube.com/watch?v=f473C43s8nE [Accessed 26th March 2024].

[33] Unity Technologies (n.d.) [online] Available at: https://docs-multiplayer.unity3d.com/netcode/current/advanced-topics/messaging-system/ [Accessed 30th March 2024].

[34] Technologies, U. (n.d.). Lobby: Private Video Game Room Creator Software — Unity. [online] unity.com. Available at: https://unity.com/products/lobby. [Accessed 24th April 2024]

[35] Unity. (n.d.). Vivox: In-Game Voice Chat App Software. [online] Available at: https://unity.com/products/vivox-voice-chat [Accessed 24 Apr. 2024].

[36] photon (n.d.) [online] Available at: https://www.photonengine.com/pun [Accessed 31st March 2024]

[37] Code Monkey (2023). Learn Unity Multiplayer (FREE Complete Course, Netcode for Game Objects Unity Tutorial 2023). [online] Available at: https://www.youtube.com/watch?v=7glCsF9fv3s [Accessed 6th May 2024].

A User Test 1 - Player Character Mechanics and Settings UI (Raw Data)

A note On Erroneous Data

With the reliance on testers to submit data, there is the chance of testers entering incorrect data. This may have happened with T5, who appears to have selected 'Strongly disagree' for the 'statements' whilst providing positive sentiment in the corresponding 'please explain your response' text box. I have decided to alter the 'statement' answers for T5 where I believe this to be the case. In doing so, I will place * asterisks next to the modified data.
Furthermore, there is the possibility that testers entered the incorrect time for the levels.

Question 1: Tutorial level Times

With seven levels in the tutorial, and with each level aiming to teach and reinforce the game's mechanics, the time taken to achieve the objective in each tutorial can be important to understanding whether testers are spending too much time on a certain level. As all the data was provided voluntarily, and thus all answers in the questionnaire were optional, there were a few instances where testers did not submit their time for a level. Please refer to (5.9) for a description of each level.

Level One Time

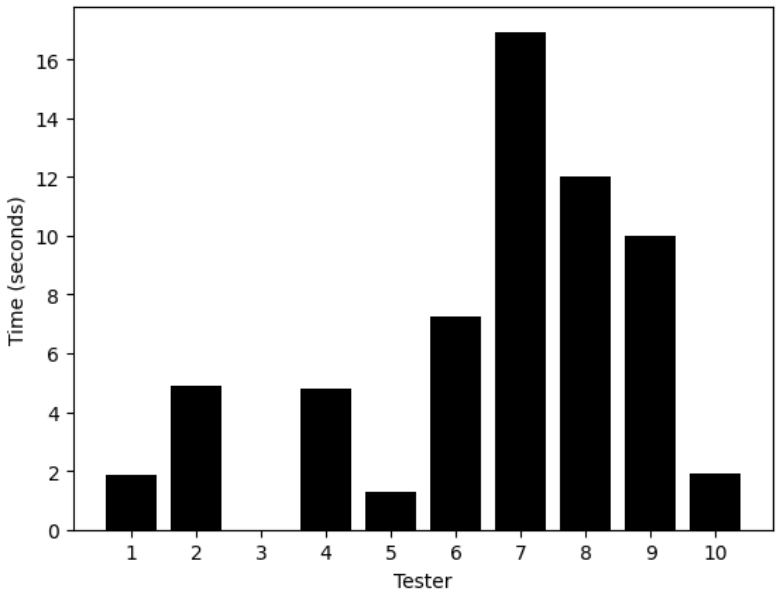


Figure 64: Bar chart showing time taken to complete the objective of level one. T3 did not submit a time. Mean: 6.77 (seconds). Standard deviation: 5.32

Level Two Time

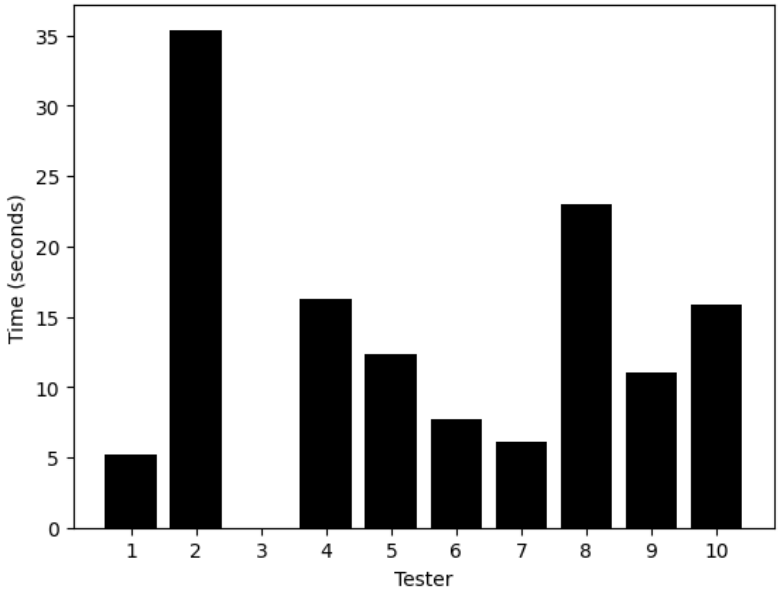


Figure 65: Bar char showing time taken to complete the objective of level two. T3 did not submit a time. Mean: 14.78 (seconds). Standard deviation: 9.57

Level Three Time

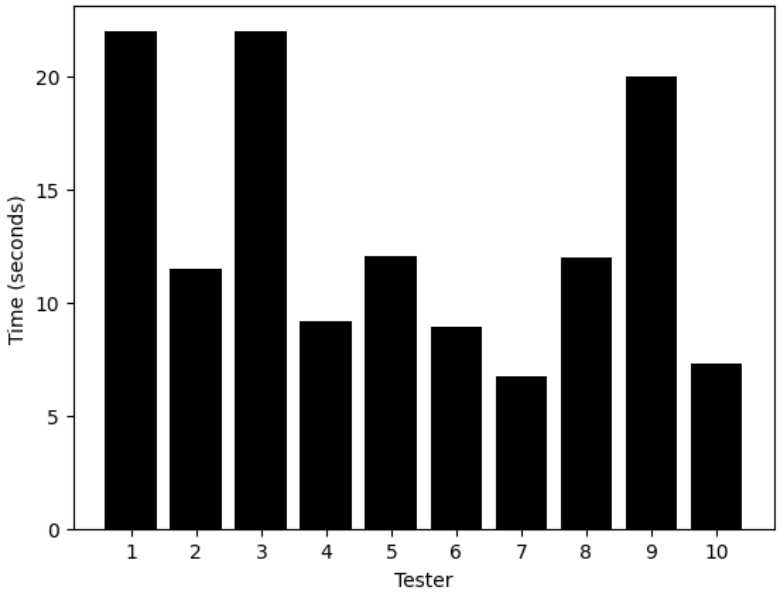


Figure 66: Bar char showing time taken to complete the objective of level three. Mean: 13.17 (seconds). Standard deviation: 5.94

Level Four Time

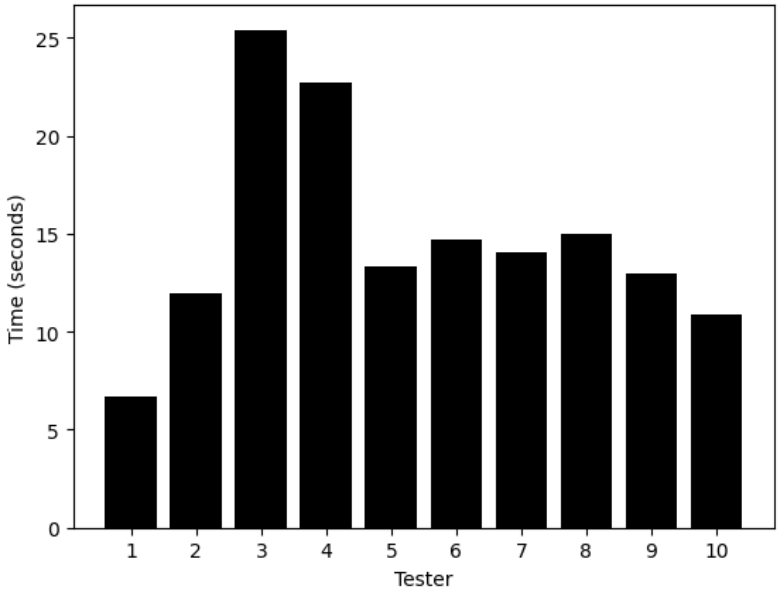


Figure 67: Bar char showing time taken to complete the objective of level four. Mean: 14.77 (seconds). Standard deviation: 5.47

Level Five Time

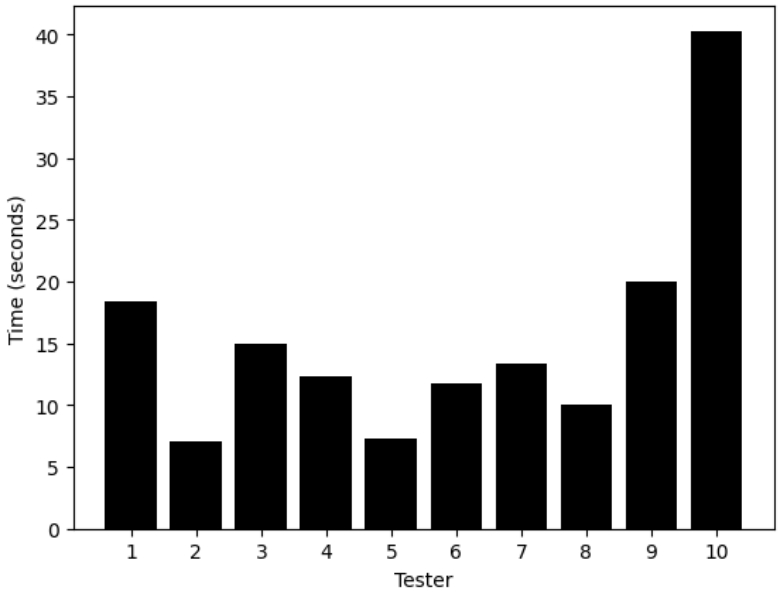


Figure 68: Bar char showing time taken to complete the objective of level five. Mean: 15.55 (seconds). Standard deviation: 9.67

Level Six Time

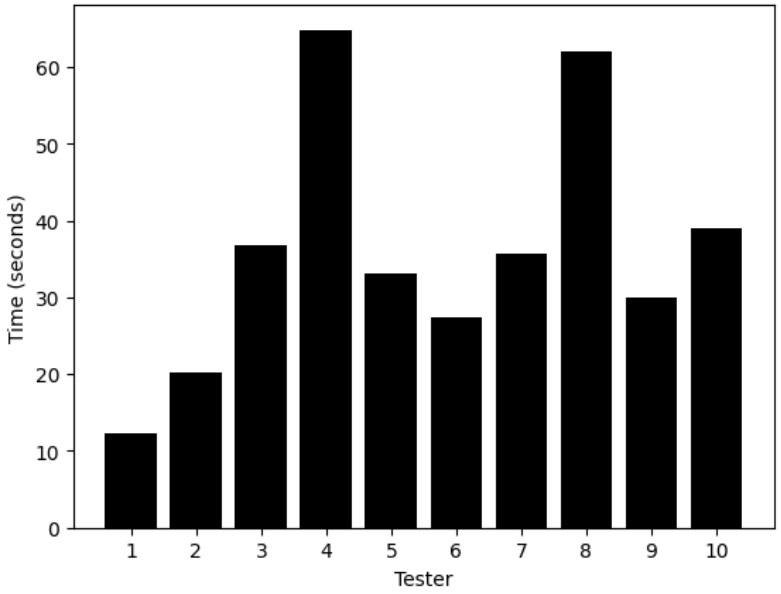


Figure 69: Bar char showing time taken to complete the objective of level six. Mean: 36.13 (seconds). Standard deviation: 16.48

Level Seven Time

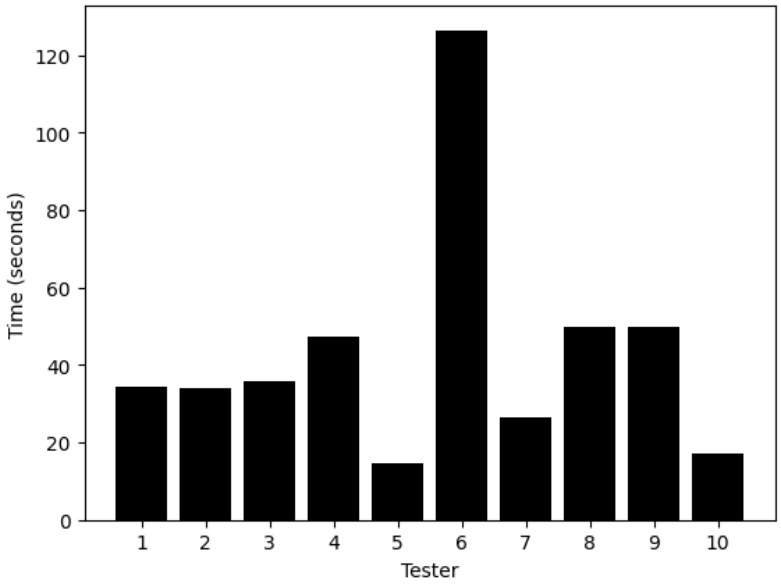


Figure 70: Bar char showing time taken to complete the objective of level seven. Mean: 43.66 (seconds). Standard deviation: 31.64

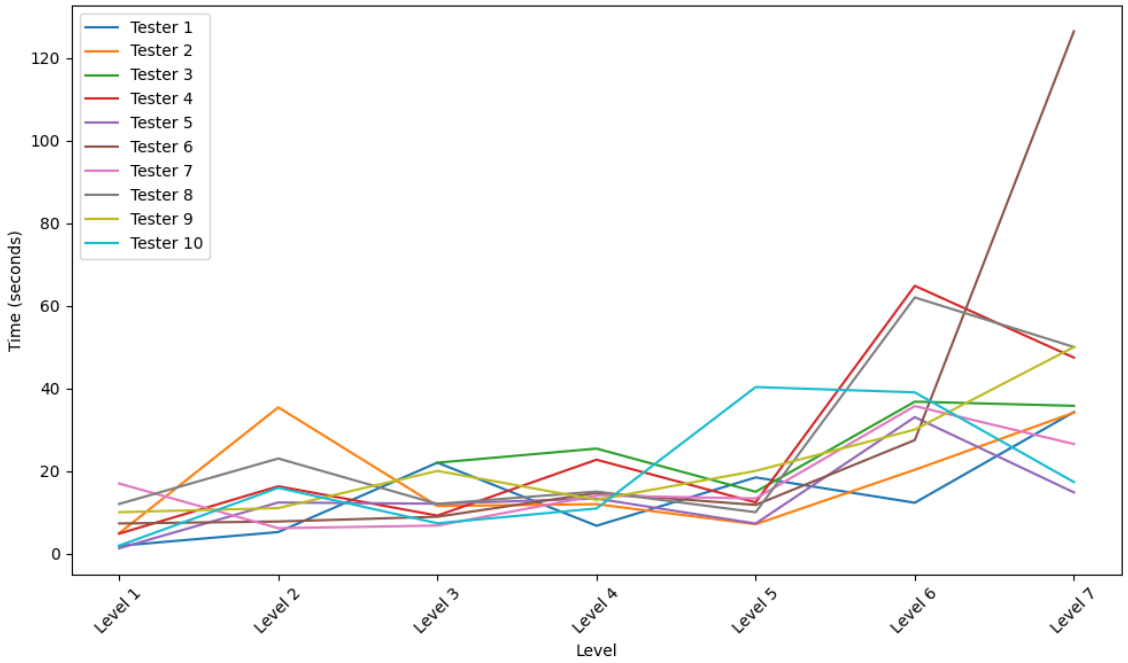


Figure 71: Tutorial times for each level, displayed in a single graph.

Question 2: On average, how many hours per week do you spend playing video games?

Tester	Response
T1	5-10 hours
T2	5-10 hours
T3	5 - 10 hours
T4	1-2 hours
T5	2-5 hours
T6	<1 hour
T7	2-5 hours
T8	>10 hours
T9	>10 hours
T10	2-5 hours

Table 5: Responses for Question 2 of User Test 1 questionnaire

Question 3: Which input device did you use to play the tutorial?

Tester	Response
T1	Gamepad (Xbox)
T2	Keyboard + Mouse
T3	Keyboard + Mouse
T4	Keyboard + Mouse
T5	Gamepad (PlayStation)
T6	Keyboard + Trackpad
T7	Keyboard + Mouse
T8	Keyboard only
T9	Keyboard + Mouse
T10	Keyboard + Mouse

Table 6: Responses for Question 3 of User Test 1 questionnaire

Question 4: Which operating system did you use to play the tutorial?

Tester	Response
T1	Windows
T2	Windows
T3	Windows
T4	Windows
T5	Windows
T6	Windows
T7	Windows
T8	macOS
T9	Windows
T10	Windows

Table 7: Responses for Question 4 of User Test 1 questionnaire

Question 5: Did you encounter any issues while playing that you felt limited your experience?

Tester	Feedback
T1	Yes - "Slightly confusing moments, did not know I couldn't move when I had the ball, thought the 'spikes' were spikes."
T2	Yes - "The default sensitivity was too high at first, and I needed to adjust it mid-game. It would be interesting to create a sensitivity tester with sliders that load as soon as you start the game."
T3	No
T4	Yes - "Exiting the game and not remembering reticle colour choice or when even entering the game not remembering reticle colour choice."
T5	Yes - "I felt there needed to be an option to use the 'circle' or 'B' button to go back through the menus instead of having to scroll down and select 'go back' as it makes the menu quicker to traverse."
T6	No
T7	Yes - "I had to restart level 7 due to a glitch- after falling from the top, I was unable to jump on the floor."
T8	Yes - "I think there is a glitch on level 3, if you jump into the back of the goal and then out of it, you kind of float about. You lose some control and the camera can clip with the walls."
T9	No
T10	No

Table 8: Feedback for Question 5 of User Test 1 questionnaire

Constructive Feedback:

- "Slightly confusing moments, did not know I couldn't move when I had the ball, thought the 'spikes' were spikes."
- "The default sensitivity was too high at first, and I needed to adjust it mid-game. It would be interesting to create a sensitivity tester with sliders that load as soon as you start the game."
- "Exiting the game and not remembering reticle colour choice or when even entering the game not remembering reticle colour choice."
- "I felt there needed to be an option to use the 'circle' or 'B' button to go back through the menus instead of having to scroll down and select 'go back' as it makes the menu quicker to traverse."
- "I had to restart level 7 due to a glitch- after falling from the top, I was unable to jump on the floor."
- "I think there is a glitch on level 3, if you jump into the back of the goal and then out of it, you kind of float about. You lose some control and the camera can clip with the walls."

Statement 1: The text on-screen was easy to read, understand and modify

For XAG 101

Tester	Feedback
T1	Strongly agree - "Very large font option, was using 1080P resolution."
T2	Strongly agree - "Fluid text settings, really liked the structure there."
T3	Strongly agree - "Two types of font and the ability to adjust the size of the text makes it very easy to customise it to suit what you personally need to read."
T4	Slightly agree - "I was able to modify the size of the text which worked well however there was a limited amount of text options available for font."
T5	Strongly agree* - "The options for font and sizes allowed for text to be customised to how I see fit which made to easy to read and understand."
T6	Strongly agree - "Settings are clear, font size easy to change."
T7	Strongly agree - "Having the feature to resize the text and font made it very easy to set an ideal size for my screen."
T8	Strongly agree - "Lots of customisation options so I could choose what font/size/etc suited me."
T9	Strongly agree - "Text was to the point and readable, consistent like plain toast."
T10	Strongly agree

Table 9: Feedback for Statement 1 of User Test 1 questionnaire

Positive Feedback

- "Very large font option, was using 1080P resolution."
- "Fluid text settings, really liked the structure there."
- "Two types of font and the ability to adjust the size of the text makes it very easy to customise it to suit what you personally need to read."
- "I was able to modify the size of the text which worked well however there was a limited amount of text options available for font."
- "I was able to modify the size of the text which worked well"
- "The options for font and sizes allowed for text to be customised to how I see fit which made to easy to read and understand."
- "Settings are clear, font size easy to change."
- "Having the feature to resize the text and font made it very easy to set an ideal size for my screen."
- "Lots of customisation options so I could choose what font/size/etc suited me."

- "Text was to the point and readable, consistent like plain toast."

Constructive Feedback

- "...there was a limited amount of text options available for font."

Statement 2: It was easy to see and understand the purpose of different objects in the game (for example, a platform, a ball, etc.)

For XAG 102

Tester	Feedback
T1	Strongly agree - "The gameplay is simple to understand."
T2	Slightly agree - "Loved the design, but I feel like the details could be pronounced a bit with either colour (unless you are going for a monochrome aesthetic) or more protruding borders on the structures to make sure you understand exactly what everything is and how it works. Think how Don't Starve does its bordering work."
T3	Slightly agree - "All really good but the outline that the see through block have would show up even on the other side of a solid wall."
T4	Slightly agree - "On the later levels when the platforms were moving up and down there were spikes on them which looked like if I hit them then the level would restart."
T5	Strongly agree* - "The tutorial instructions outlined the functionality very well."
T6	Strongly agree - "Ball, platform and goal concepts all easy to understand."
T7	Slightly agree - "It was very clear to understand most of the objects- although I had a bit of confusion on one of the levels knowing which side of the goal I was on. When first seeing the platforms it was very clear what they were- however the central parts looked like they may have been spikes. One other source of confusion was that there were brief times that I thought I could change my gravity on platforms and regular walls."
T8	Strongly agree - "Understood from playing games with similar elements."
T9	Strongly agree - "Goal is Goal, ball is ball, spike is deadly, good job!!."
T10	Slightly disagree

Table 10: Feedback for Statement 2 of User Test 1 questionnaire

Positive Feedback

- "The gameplay is simple to understand."
- "The tutorial instructions outlined the functionality very well."
- "Ball, platform and goal concepts all easy to understand."

- "Understood from playing games with similar elements."
- "Goal is Goal, ball is ball, spike is deadly, good job!!."

Constructive Feedback

- "...the details could be pronounced a bit with either colour (unless you are going for a monochrome aesthetic) or more protruding borders on the structures to make sure you understand exactly what everything is and how it works.
- "All really good but the outline that the see through block have would show up even on the other side of a solid wall."
- "On the later levels when the platforms were moving up and down there were spikes on them which looked like if I hit them then the level would restart."
- "...I had a bit of confusion on one of the levels knowing which side of the goal I was on."
- "...When first seeing the platforms it was very clear what they were- however the central parts looked like they may have been spikes."
- "...One other source of confusion was that there were brief times that I thought I could change my gravity on platforms and regular walls."

Statement 3: It was easy to understand what I was supposed to do in each level

Tester	Feedback
T1	Strongly agree - "They were explained well."
T2	Strongly agree - "The level design was clear and intuitive, some were challenging to a small extent, but enjoyable level of challenge, not hindering the experience at all."
T3	Strongly agree - "Simple but fun 'mission' to beat each level and it explained it in a neat and concise way in the bottom of your screen."
T4	Strongly agree - "Was told what to do."
T5	Strongly agree* - "The gradual build up of using different abilities to complete the levels allowed for easy understanding."
T6	Slightly agree - "Task was always clear, with text explaining on the overlay. Sometimes solution not obvious, but that is understandable."
T7	Strongly agree - "Yes, the levels progression and bottom left text really helped me understand what to do. The only thing I thought I could do that I wasn't able to do was move whilst touching the ball- however the first time I did this I learned this straight away.."
T8	Strongly agree - "The goal is made clear in the bottom left."
T9	Strongly agree - "Good simple gameplay loop, find ball and throw ball in goal (try not to die aswell)."
T10	Strongly agree

Table 11: Feedback for Statement 3 of User Test 1 questionnaire

Positive Feedback

- "They were explained well."
- "The level design was clear and intuitive, some were challenging to a small extent, but enjoyable level of challenge, not hindering the experience at all."
- "Simple but fun 'mission' to beat each level and it explained it in a neat and concise way in the bottom of your screen."
- "Was told what to do."
- "The gradual build up of using different abilities to complete the levels allowed for easy understanding."
- "Task was always clear, with text explaining on the overlay. Sometimes solution not obvious, but that is understandable."
- "Yes, the levels progression and bottom left text really helped me understand what to do. The only thing I thought I could do that I wasn't able to do was move whilst touching the ball- however the first time I did this I learned this straight away.."
- "The goal is made clear in the bottom left."
- "Good simple gameplay loop, find ball and throw ball in goal"

Statement 4: The settings menu was easy to navigate

For XAG 112

Tester	Feedback
T1	Strongly agree
T2	Slightly agree - "I think the menu was clear, but the options seemed a little redundant with needing to press multiple buttons at times, I feel like it should be button into sliders, rather than buttons into more buttons, although it is more organised this way, what I suggest feels more natural to me personally."
T3	Slightly agree - "Easy to find where things were only gripe could be to include the option to press escape or some other key or button to go back a layer in the menu as well as the button on screen."
T4	Slightly agree - "The menus were slightly cluttered with the number of options available but overall could find most things."
T5	Strongly agree* - "Other than what was mentioned previously, the menus were clearly categorised and well made."
T6	Strongly agree - "Settings are clear. All options are where you would expect them to be. Only suggestion would be to change 'Reticle' to crosshair, as I didn't think it was immediately obvious when loading the game, although thats just nit-picking."
T7	Slightly agree - "The options were perfect, however one slight confusion when using keyboard and mouse was that the top setting is selected by default- which made me initially think it was marked as unclickable."
T8	Strongly agree - "Easy to change between settings and adjust things."
T9	Strongly agree - "To the point, no jargon."
T10	Strongly agree

Table 12: Feedback for Statement 4 of User Test 1 questionnaire

Positive Feedback

- "...the menu was clear"
- "Easy to find where things were"
- "...the menus were clearly categorised and well made."
- "Settings are clear. All options are where you would expect them to be."
- "Easy to change between settings and adjust things."
- "To the point, no jargon."

Constructive Feedback

- "...the options seemed a little redundant with needing to press multiple buttons at times, I feel like it should be button into sliders, rather than buttons into more buttons, although it is more organised this way, what I suggest feels more natural to me personally."
- "...include the option to press escape or some other key or button to go back a layer in the menu as well as the button on screen."
- "The menus were slightly cluttered with the number of options available"

- "...change 'Reticle' to crosshair, as I didn't think it was immediately obvious when loading the game"
- "...one slight confusion when using keyboard and mouse was that the top setting is selected by default- which made me initially think it was marked as unclickable."

Statement 5: I always understood what button/slider in the settings menu was currently selected

For XAG 113

Tester	Feedback
T1	Strongly agree
T2	Strongly agree - "Nothing to comment on here, solid ."
T3	Slightly agree - "The first menu when you pressed escape seemed a little buggy and would highlight 'controls' despite me hovering over a different menu option."
T4	Slightly agree - "For certain options like selecting screen resolution the slider was very fidgetty and having a drop down would have made it easier to select."
T5	Strongly agree* - "Everything was well labelled."
T6	Strongly agree - "Well made with numerical indication of progress."
T7	Strongly agree - "The buttons and sliders were very intuitive in the settings, the sound the button made made the menu feel a lot more interactive."
T8	Strongly agree - "The button turns to a darker shade when selected."
T9	Strongly agree - "Once again, good ui."
T10	Strongly agree

Table 13: Feedback for Statement 5 of User Test 1 questionnaire

Positive Feedback

- "Everything was well labelled."
- "Well made with numerical indication of progress."
- "The buttons and sliders were very intuitive in the settings, the sound the button made made the menu feel a lot more interactive."

Constructive Feedback

- "The first menu when you pressed escape seemed a little buggy and would highlight 'controls' despite me hovering over a different menu option."
- "For certain options like selecting screen resolution the slider was very fidgetty and having a drop down would have made it easier to select."

Statement 6: I felt it was easy to customise the camera's movement settings (do not answer if you did not change the camera's movement settings)

For XAG 117

Tester	Feedback
T1	
T2	Strongly agree - "Very intuitive settings for this, but the starting point could be improved as I suggested before."
T3	Slightly agree - "Easy to do and allowed for a lot of customisation of the speeds of rotation."
T4	Slightly disagree - "No matter how much I changed the camera's movement setting it still felt too fast which made it clunky."
T5	Strongly agree* - "The sensitivity between mouse and controller differed a lot and was very easy to change when switching between."
T6	
T7	Slightly agree - "It was very intuitive to change, however when using the mouse on my computer- I had to set the setting down to 0.10 to be ideal."
T8	Slightly agree - "Easy to use. Camera sensitivity being under controls meant it took me a while to find. Perhaps the settings could have a section titled 'Camera'."
T9	
T10	Strongly agree

Table 14: Feedback for Statement 6 of User Test 1 questionnaire

Positive Feedback

- "Very intuitive settings"
- "Easy to do and allowed for a lot of customisation of the speeds of rotation."
- "The sensitivity between mouse and controller differed a lot and was very easy to change when switching between."
- "It was very intuitive to change"
- "Easy to use"

Constructive Feedback

- "...the starting point [of the camera's x and y sensitivity sliders] could be improved"
- "No matter how much I changed the camera's movement setting it still felt too fast which made it clunky."
- "...Camera sensitivity being under controls meant it took me a while to find. Perhaps the settings could have a section titled 'Camera'."

Statement 7: I understood the purpose of each setting in the settings menu

For XAG 114

Tester	Feedback
T1	Strongly agree
T2	Strongly agree - "All good here!"
T3	Slightly agree - "All were labelled well and made sense as to were each sub section was kept."
T4	Strongly agree
T5	Strongly agree* - "Everything was done well."
T6	Slightly agree - "Same reticle thing as mentioned above - as it wasn't immediately obvious what it does when first loading the game."
T7	Strongly agree
T8	Strongly agree - "Settings contain what you would expect, apart from controls as mentioned earlier containing camera sensitivity. But it also contains key binds as expected."
T9	Strongly agree - "Good UI."
T10	Strongly agree

Table 15: Feedback for Statement 7 of User Test 1 questionnaire

Positive Feedback

- "All were labelled well and made sense as to were each sub section was kept."
- "Settings contain what you would expect, apart from controls as mentioned earlier containing camera sensitivity. But it also contains key binds as expected."

Constructive Feedback

- "Same reticle thing...[mentioned in previous statement]... - as it wasn't immediately obvious what it does when first loading the game."

Statement 8: The tutorial helped me understand the game's controls

Tester	Feedback
T1	Strongly agree
T2	Slightly agree - "I would slightly say that the tutorial didn't feel like a tutorial at all, which is natural for a game that is still in development, but keep in mind for later stages of development."
T3	Strongly agree - "Run well as it introduces a mechanic then adds to it. introduces another mechanic and then combines them to make a more complex solution."
T4	Strongly agree
T5	Strongly agree* - "Having a focus on single aspects in the early tutorials allowed for easy learning."
T6	Strongly agree - "Introduces a feature in each exercise, which helps the player build and combine their knowledge."
T7	Strongly agree - "The progression of the levels introduced mechanics in a way that made a lot of sense, and it was very easy to understand how to play the game."
T8	Strongly agree - "Tutorial levels gradually added different control elements as to not overwhelm the player."
T9	Slightly agree - "I feel like the tutorial text could've stayed a bit longer if anything."
T10	Slightly disagree

Table 16: Feedback for Statement 8 of User Test 1 questionnaire

Positive Feedback

- "Run well as it introduces a mechanic then adds to it. introduces another mechanic and then combines them to make a more complex solution."
- "Having a focus on single aspects in the early tutorials allowed for easy learning."
- "Introduces a feature in each exercise, which helps the player build and combine their knowledge."
- "The progression of the levels introduced mechanics in a way that made a lot of sense, and it was very easy to understand how to play the game."
- "Tutorial levels gradually added different control elements as to not overwhelm the player."

Constructive Feedback

- "I would slightly say that the tutorial didn't feel like a tutorial at all, which is natural for a game that is still in development, but keep in mind for later stages of development."
- "I feel like the tutorial text could've stayed a bit longer if anything."

Statement 9: The controls were easy to understand

Tester	Feedback
T1	Strongly agree
T2	Strongly agree - "Controls felt SUPER intuitive, amazing work here!"
T3	Strongly agree - "Simple but good movement."
T4	Strongly agree
T5	Strongly agree* - "There were no strange controls and the main set was small which made it easy to grasp quickly."
T6	Slightly agree - "Typical video game controls. Very intuitive."
T7	Strongly agree - "The controls felt very intuitive, and the gravity and feel of the character felt great too. One control that I think could be a great addition is something that makes the player boost downwards back towards their ground."
T8	Strongly agree - "Controls were conventional, what you would expect. Not too complicated or too many."
T9	Strongly agree - "Standard WASD pretty good."
T10	Slightly agree

Table 17: Feedback for Statement 9 of User Test 1 questionnaire

Positive Feedback

- "Controls felt SUPER intuitive, amazing work here!"
- "Simple but good movement."
- "There were no strange controls and the main set was small which made it easy to grasp quickly."
- "Typical video game controls. Very intuitive."
- "The controls felt very intuitive, and the gravity and feel of the character felt great too."
- "Controls were conventional, what you would expect. Not too complicated or too many."
- "Standard WASD pretty good."

Constructive Feedback

- "...One control that I think could be a great addition is something that makes the player boost downwards back towards their ground."

Question 6: What do you like the most about the game?

Tester	Feedback
T1	"Simple gameplay, simple objective."
T2	"The creativity, the smooth transition to different gravities, and the controls."
T3	"The concept and the puzzles."
T4	"It was easy enough to pickup with the simple idea and well thought out mechanic."
T5	"The transitioning between surface rotations."
T6	"The concept. I enjoyed the 3D nature of the game; I thought changing the players orientation was creative."
T7	"After the tutorial, the game felt very fun to play- I really like how the gravity in the game isn't constant and the play must figure out how to effectively move around the space. If this game was completed or there was an online aspect, I'd certainly love to play this game!"
T8	"The movement, double jump and walking on walls."
T9	"Throwing the ball was pretty funny."
T10	

Table 18: Feedback for Question 6 of User Test 1 questionnaire

Positive Feedback

- "Simple gameplay, simple objective."
- "The creativity, the smooth transition to different gravities, and the controls."
- "The concept and the puzzles."
- "It was easy enough to pickup with the simple idea and well thought out mechanic."
- "The transitioning between surface rotations."
- "The concept. I enjoyed the 3D nature of the game; I thought changing the players orientation was creative."
- "After the tutorial, the game felt very fun to play- I really like how the gravity in the game isn't constant and the play must figure out how to effectively move around the space. If this game was completed or there was an online aspect, I'd certainly love to play this game!"
- "The movement, double jump and walking on walls."

Question 7: What do you think could be improved?

Tester	Feedback
T1	"More varied environments."
T2	"Everything I mentioned already, but I also recommend some QOL changes to the platform shapes and to the way G0 is introduced. Overall loved it."
T3	"More levels."
T4	"The movement in regards to jumping is very bad. When jumping it is hard to aim and when mid air the acceleration and speed feels too fast compared to how much you move. Which leads to over compensating. When switching surface types or when switching the way of movement it sometimes jitters out the camera and takes longer than expected to switch."
T5	"Traversing the menu backwards through the use of a control."
T6	"The graphics and maybe more realistic physics when throwing the ball."
T7	"The main thing I think that could improve the game could be to colour code the objects in the scene- especially the platforms that allow you to change your gravity. Initially I tried to switch my gravity on the platforms and regular walls, however once I learned what the gravity changing platforms looked like it and intuitive and really great. Thank you!"
T8	"The transparent walls can be a little confusing. They can look kind of empty, like there's nothing to land on. A level select setting would be good, so that you could return to previous levels. The graphics are quite basic and the colours are not particularly exciting."
T9	"When I stuck to the walls I thought that was pretty jarring on the eyes, maybe slow that interaction around, I personally felt disorientated."
T10	

Table 19: Feedback for Question 7 of User Test 1 questionnaire

Constructive Feedback

- "More varied environments."
- "...I also recommend some QOL [(quality of life)] changes to the platform shapes and to the way G0 is introduced."
- "More levels."
- "The movement in regards to jumping is very bad. When jumping it is hard to aim and when mid air the acceleration and speed feels too fast compared to how much you move. Which leads to over compensating. When switching surface types or when switching the way of movement it sometimes jitters out the camera and takes longer than expected to switch."
- "Traversing the menu backwards through the use of a control."

- "The graphics and maybe more realistic physics when throwing the ball."
- "The main thing I think that could improve the game could be to colour code the objects in the scene- especially the platforms that allow you to change your gravity. Initially I tried to switch my gravity on the platforms and regular walls, however once I learned what the gravity changing platforms looked like it and intuitive and really great. Thank you!"
- "The transparent walls can be a little confusing. They can look kind of empty, like there's nothing to land on. A level select setting would be good, so that you could return to previous levels. The graphics are quite basic and the colours are not particularly exciting."
- "When I stuck to the walls I thought that was pretty jarring on the eyes, maybe slow that interaction around, I personally felt disorientated."

B User Test 2 - Online Functionality (Raw Data)

Question 1: On average, how many hours per week do you spend playing video games similar to the one you just played? (3D first-person, competitive online multiplayer)

Tester	Response
T1	2-5 hours
T2	<1 hour
T3	2-5 hours
T4	1-2 hours
T5	>10 hours
T6	1-2 hours
T7	1-2 hours
T8	2-5 hours

Table 20: Feedback for Question 1 of Online Playtest questionnaire

Question 2: Which form of input did you use during the test?

Tester	Response
T1	Keyboard + Mouse
T2	Keyboard + Mouse
T3	Keyboard + Mouse
T4	Keyboard + Mouse
T5	Keyboard + Mouse
T6	Keyboard + Mouse
T7	Keyboard + Mouse
T8	Keyboard + Mouse

Table 21: Feedback for Question 2 of Online Playtest questionnaire

Question 3: Which operating system did you use to play the tutorial?

Tester	Response
T1	Windows
T2	Windows
T3	Windows
T4	Windows
T5	Windows
T6	Windows
T7	Windows
T8	Windows

Table 22: Feedback for Question 3 of Online Playtest questionnaire

Statement 1: The interface was easy to navigate.

Tester	Response
T1	Strongly agree - "Very clear, nothing was hard to read."
T2	Strongly agree
T3	Slightly agree - "Easy to find its just that when clicking through it can be a bit janky."
T4	Strongly agree - "Clean easy to tell what everything did."
T5	Strongly agree - "Very to the point very easy."
T6	Strongly agree - "The interface was easy to navigate. simple to find options, settings are good. Was alright."
T7	Strongly agree - "The interface was very well designed, and the progression and buttons acted as expected. The button sound made it more interactive."
T8	Strongly agree - "Everything was categorised well and had obvious labels."

Table 23: Feedback for Statement 1 of Online Playtest questionnaire

Positive Feedback

- "Very clear, nothing was hard to read."
- "Clean easy to tell what everything did."
- "Very to the point very easy."
- "The interface was easy to navigate. simple to find options, settings are good. Was alright."
- "The interface was very well designed, and the progression and buttons acted as expected. The button sound made it more interactive."
- "Everything was categorised well and had obvious labels."

Constructive Feedback

- "...clicking through it can be a bit janky."

Statement 2: I understood which UI button/slider was highlighted

Tester	Response
T1	Strongly agree - "Very responsive sliders/answers were highlighted."
T2	Slightly disagree - "Seems a bit buggy, buttons darkened despite not being selected."
T3	Strongly agree - "Easy to navigate due to how it was highlighted."
T4	Strongly agree - "It was coloured differently, makes it easier to differentiate the sliders."
T5	Strongly agree - "Same as last time, easy and to the point."
T6	Strongly agree - "It was clear and easy to use."
T7	Slightly agree - "The UI worked perfectly, however initially in menus the top button was auto selected (for use for controllers), when I was using keyboard and mouse."
T8	Strongly agree - "The darkening was easy to see and the colour scheme chosen also meant that the mouse was easily seen."

Table 24: Feedback for Statement 2 of Online Playtest questionnaire

Positive Feedback

- "Very responsive sliders/answers were highlighted."
- "Easy to navigate due to how it was highlighted."
- "It was coloured differently, makes it easier to differentiate the sliders."
- "Same as last time, easy and to the point."
- "It was clear and easy to use."
- "The darkening was easy to see and the colour scheme chosen also meant that the mouse was easily seen."

Constructive Feedback

- "Seems a bit buggy, buttons darkened despite not being selected."
- "...initially in menus the top button was auto selected (for use for controllers), when I was using keyboard and mouse."

Statement 3: Finding and joining a lobby (either through a code, or clicking on one of the public lobbies) felt straight-forward and easy

Tester	Response
T1	Slightly agree - "Crash after every game. Game itself worked perfectly fine."
T2	Slightly disagree - "A bit buggy, but possible to connect if restarting the game."
T3	Strongly disagree - "Was very difficult and sometimes just didn't work."
T4	Slightly agree - "Had to restart a couple times to get it to work."
T5	Slightly agree - "I realize that this is a bug, but having to relaunch the game after a round was tedious. When setting up a lobby worked, it was very easy."
T6	Slightly agree - "We had a couple problems initially. Everyone needs to restart their game when the host creates the lobby, once everyone has restarted, the lobby appears and is joinable. same applies for joining private games."
T7	Strongly agree - "It was very easy to join a game, and the amount of button clicks was very minimal."
T8	Strongly agree - "The system was easy to use and to navigate. All the information needed to join was there and it was very obvious as to what to do."

Table 25: Feedback for Statement 3 of Online Playtest questionnaire

Positive Feedback

- "...Game itself worked perfectly fine."
- "When setting up a lobby worked, it was very easy."
- "It was very easy to join a game, and the amount of button clicks was very minimal."
- "The system was easy to use and to navigate. All the information needed to join was there and it was very obvious as to what to do."

Constructive Feedback

- "Crash after every game."
- "Was very difficult and sometimes just didn't work."
- "Had to restart a couple times to get it to work."
- "...having to relaunch the game after a round was tedious."

Question 4: What did you like the most about the lobby setup process?

Tester	Response
T1	"Very simple."
T2	
T3	"Having the text to speech for chat."
T4	"The chat in the lobby screen."
T5	"The pins were pretty good."
T6	"I like the pre-game text chat and being able to clearly see whos in the lobby."
T7	"The creating lobby UI was very simple and easy to understand, I liked the simple nature of the toggle buttons for options."
T8	"I liked that the menu showed already created lobbies. This made it accessible for people that may be playing the multiplayer mode alone."

Table 26: Feedback for Question 4 of Online Playtest questionnaire

Positive Feedback

- "Very simple."
- "Having the text to speech for chat."
- "The chat in the lobby screen."
- "The pins were pretty good."
- "I like the pre-game text chat and being able to clearly see whos in the lobby."
- "The creating lobby UI was very simple and easy to understand, I liked the simple nature of the toggle buttons for options."
- "I liked that the menu showed already created lobbies. This made it accessible for people that may be playing the multiplayer mode alone."

Question 5: What do you think could be improved?

Tester	Response
T1	"Add an option to set/change name. Also display what teams people are on."
T2	
T3	"Lobby selection and joining [could be improved]."
T4	"The voice chat stays connected after the games have ended and you are kicked to the menu screen."
T5	"Chat text limit, bug fixes, mic muted by default."
T6	"Joining process wasn't smooth. also tts can get annoying. an option to turn it off would be cool."
T7	"I think UI to change your name could help greatly, being able to set username code could also be really cool. Another very small thing could be adding some spinning "loading" UI during loading screens."
T8	"The connections were on and off. But after a bit of messing around it did work in the end. So it wasn't that big of an issue."

Table 27: Feedback for Question 5 of Online Playtest questionnaire

Constructive Feedback

- "Add an option to set/change name. Also display what teams people are on."
- "Lobby selection and joining [could be improved]."
- "The voice chat stays connected after the games have ended and you are kicked to the menu screen."
- "Chat text limit, bug fixes, mic muted by default."
- "Joining process wasn't smooth. also tts can get annoying. an option to turn it off would be cool."
- "I think UI to change your name could help greatly, being able to set username code could also be really cool. Another very small thing could be adding some spinning "loading" UI during loading screens."
- "The connections were on and off. But after a bit of messing around it did work in the end. So it wasn't that big of an issue."

Statement 4: The score HUD (heads-up display) was easy to read, so I always knew what the score was.

Tester	Response
T1	Strongly agree - "Score was clear, what team we were on wasn't."
T2	Slightly agree - "I think it would be better if it was more clear to see the teams, who is on what team and what team I am on."
T3	Strongly agree - "Was easy to see."
T4	Strongly agree - "Stood out from the rest of the environments."
T5	Strongly agree - "I always knew the score, not what team i was on though."
T6	Strongly agree - "I knew what the score was and how long was left on the clock. Design could be worked on but gave useful info."
T7	Strongly agree - "It was very easy to understand the HUD, one thing that could be call would be adding team colours to each team's text."
T8	Slightly agree - "To me the scores didn't stand out on the screen and kind of blended in with the game as it's quite monotone. Possibly could have had a slight tint of colour in order to make it stand out a bit more."

Table 28: Feedback for Statement 4 of Online Playtest questionnaire

Positive Feedback

- "Score was clear."
- "Was easy to see."
- "Stood out from the rest of the environments."
- "I always knew the score."
- "I knew what the score was and how long was left on the clock."
- "It was very easy to understand the HUD"

Constructive Feedback

- "I think it would be better if it was more clear to see the teams, who is on what team and what team I am on."
- "To me the scores didn't stand out on the screen and kind of blended in with the game as it's quite monotone. Possibly could have had a slight tint of colour in order to make it stand out a bit more."

Statement 5: The voice and text chat was easy to use.

Tester	Response
T1	Strongly agree - "Very simple system on muting and sending messages."
T2	Slightly agree
T3	Strongly agree - "Was very effective."
T4	Strongly agree - "Simple and works cleanly."
T5	Strongly agree - "Easy to use, voice chat sometimes broke."
T6	Strongly agree - "Text chat worked in lobby creation, but I couldn't figure out how to get it working in-game."
T7	Slightly agree - "It worked very well, however when adding large amounts of text the text to voice could continue for a long time. One small issue I found was in a lobby by myself, my messages sent twice."
T8	Strongly agree - "It was very obvious where to type messages and the voice chat worked well."

Table 29: Feedback for Statement 5 of Online Playtest questionnaire

Positive Feedback

- "Very simple system on muting and sending messages."
- "Was very effective."
- "Simple and works cleanly."
- "Easy to use."
- "It was very obvious where to type messages and the voice chat worked well."

Constructive Feedback

- I couldn't figure out how to get it [the text chat] working in-game."
- "...adding large amounts of text the text to voice could continue for a long time. One small issue I found was in a lobby by myself, my messages sent twice."

Statement 6: I knew which team I was on.

Tester	Response
T1	Strongly disagree - "No way of seeing who was on what team."
T2	Slightly disagree
T3	Strongly disagree - "I had no clue as it didn't tell me or show me."
T4	Slightly disagree - "Bit hard to tell which team you were on and who was on your own team."
T5	Strongly disagree - "Hard to tell what team I was on, no indicator/ fixed spawn."
T6	Strongly disagree - "Hard to tell, other than where you spawn (which i forgot quickly)."
T7	Slightly disagree - "I think it was a little bit difficult to tell which team I was on initially, some UI or user tags of others (coloured by their team) could help! After playing I was able to tell by which goal I spawned next time."
T8	Slightly disagree - "There was not much indication as to what team I was on, other than possibly the spawn location. I think it would be good if your team had a different colour for the score at the top."

Table 30: Feedback for Statement 6 of Online Playtest questionnaire

Constructive Feedback

- "No way of seeing who was on what team."
- "I had no clue as it didn't tell me or show me."
- "Bit hard to tell which team you were on and who was on your own team."
- "Hard to tell what team I was on, no indicator/ fixed spawn."
- "Hard to tell, other than where you spawn (which i forgot quickly)."
- "I think it was a little bit difficult to tell which team I was on initially, some UI or user tags of others (coloured by their team) could help! After playing I was able to tell by which goal I spawned next time."
- "There was not much indication as to what team I was on, other than possibly the spawn location. I think it would be good if your team had a different colour for the score at the top."

Statement 7: [If you haven't played the tutorial] The controls were easy to pick up.

Tester	Response
T1	Strongly agree - "Very simple mechanics and controls, took about 30 seconds to get to grips with how the game worked."
T2	Strongly agree
T3	
T4	
T5	
T6	
T7	Slightly agree - "The controls were very easy to use and transferable from other first person games I've played."
T8	

Table 31: Feedback for Statement 7 of Online Playtest questionnaire

Positive Feedback

- "Very simple mechanics and controls, took about 30 seconds to get to grips with how the game worked."
- "The controls were very easy to use and transferable from other first person games I've played."

Question 6: What do you like the most about the gameplay?

Tester	Response
T1	"I love the idea of a rotating field."
T2	
T3	"How chaotic it is with 5+ players."
T4	"The throwing of the balls."
T5	"It felt really fluid, I liked the sound effects."
T6	"The core gameplay was really fun. This is a very good proof of concept for the movement feature. Multiplayer certainly made the experience better - and once we got everything working it was really good fun."
T7	"I liked the fast pace of the game, and the ability to throw the ball to your teammates- I think with practise it would be cool to try and setup team plays."
T8	"I like the movement, it feels very fluid and the camera rotations are not at all choppy or sickening (which some games similar can be)."

Table 32: Feedback for Question 6 of Online Playtest questionnaire

Positive Feedback

- "I love the idea of a rotating field."

- "How chaotic it is with 5+ players."
- "The throwing of the balls."
- "It felt really fluid, I liked the sound effects."
- "The core gameplay was really fun. This is a very good proof of concept for the movement feature. Multiplayer certainly made the experience better - and once we got everything working it was really good fun."
- "I liked the fast pace of the game, and the ability to throw the ball to your teammates- I think with practise it would be cool to try and setup team plays."
- "I like the movement, it feels very fluid and the camera rotations are not at all choppy or sickening (which some games similar can be)."

Question 7: What do you think could be improved?

Tester	Response
T1	"I think it would be really clever to either add items, or characters with unique abilities/skills."
T2	
T3	"Just lobby selection and joining."
T4	"The voice chat stays connected."
T5	
T6	"The most fundamental changes are knowing what team you are on and completely fixing the joining lobby feature, as that was a bit of a struggle at times. After, energy could be put into graphics. One suggestion would be to make the players slightly larger to help encourage teamwork."
T7	"I think the game worked really well, one idea I think that could be nice would be to make it easier to save shots (if it possible to!), since when an opponent has the ball the only way to stop them from scoring would be to tackle them."
T8	"As mentioned, adding more obvious indicators of teams and adding some colour to the important information in my opinion would be beneficial."

Table 33: Feedback for Question 7 of Online Playtest questionnaire

Constructive Feedback

- "I think it would be really clever to either add items, or characters with unique abilities/skills."
- "Just lobby selection and joining."
- "The voice chat stays connected."

- "The most fundamental changes are knowing what team you are on and completely fixing the joining lobby feature, as that was a bit of a struggle at times. After, energy could be put into graphics. One suggestion would be to make the players slightly larger to help encourage teamwork."
- "...make it easier to save shots (if it possible to!), since when an opponent has the ball the only way to stop them from scoring would be to tackle them."
- "As mentioned, adding more obvious indicators of teams and adding some colour to the important information in my opinion would be beneficial."

C Weekly Log

week beginning (w/b) - 9/10/23 Had an introductory meeting with my supervisor, outlining the key details I need to know for the individual project. Also (briefly) discussed project idea.

w/b - 16/10/23 Wrote and completed project proposal document, which included brief research into available information and resources that I may be able to use in my project. I then discussed the project proposal with my supervisor and went into greater detail about the structure of development for my software and why I wanted to pursue this particular project.

w/b 23/10/23 Developed a working proof of concept to be able to visualise the core mechanic of my idea. Also learnt more about accessibility through Microsoft's 'Gaming Accessibility Fundamentals' course.

/b 30/10/23 Began writing interim report. Also had a group meeting with my supervisor where I briefly discussed the progress I had made since our last meeting - we then went through the requirements of the interim report and looked at an example of someone who achieved a high mark for their submission in a previous year.

w/b 6/11/23 Spent the week finishing the first draft of the interim report. Had a 1:1 meeting with my supervisor to check the progress of my report, as well as the state of my proof of concept. We discussed the ethical concerns that may arise with testing the software on people who have an impairment - the approach to take in this regard will be discussed in a future meeting.

w/b 13/11/23 Finalised work on the interim report, using feedback from my supervisor to help improve the report. I was then able to submit the report before the deadline.

w/b 20/11/23 Did not complete work this week due to being occupied with other priorities in my course, specifically coursework in my 'introduction to computer security' module.

w/b 27/11/23 Created list of boards representing a part of the project (examples: UI research; gameplay tasks) in HacknPlan app. Also researched scriptable object technology found in the Unity game engine. The modularity that scriptable objects provides will be useful when the game scales in complexity.

w/b 4/12/23 Refactored character controller so that player can change direction of gravity independent of the surface used as the ground. Also included extra mechanics in the character controller: sprinting, wall running.

w/b 11/12/23 Designed basic UX flow chart for the menu interface. Also gathered research on the features available in similar commercially available games.

w/b 18/12/23 Researched unity's online service 'Netcode for Gameobjects'.

w/b 25/12/23 Holiday break - no significant work completed

w/b 1/1/24 Started work on getting basic online functionality, using host-client design. The player is allowed to use their machine as a server (making them the host) and let other people (clients) join.

w/b 8/1/24 Exam period - no significant work completed

w/b 15/1/24 Found 'Game Usability' book, with excellent reference material for usability design and implementation. Decided to read and take notes.

w/b 22/1/24 Continued to read Game Usability book for most of the week. Also implemented the game's timer as well as a main menu interface using UX flow chart

w/b 29/1/24 Continued to read through and make notes of Game Usability text-

book.
Programming and other research:

- Included sound effects for input commands. Audio implementation is not complete though so will need to come back and finish this.
- Found out about Unity's DOTS (Data-Oriented Technology Stack), which includes Netcode for Entities. Apparently a useful tool for server-authoritative games. Spent a few hours on it before deciding the complexity was too much, and I am too late into production of the game to switch over to Netcode for Entities (it would also require having to learn Data-Oriented Programming (I am currently using Object-Oriented Programming)).
- Implemented offline 'training' functionality. This allows the player to re-familiarise themselves with the controls of the game. Technically a cognitive accessibility feature for players who have forgotten the controls, however this is underdeveloped.

w/b 5/2/24 Had a meeting with my supervisor, showcasing the work I had made so far. We discussed a roadmap of the next 2-3 weeks, as well as what I was going to do to make the game more accessible. I said I would work on implementing a variety of settings that can be controlled through a user interface, and then construct a tutorial for the controls. This tutorial will then be tested (alongside testing the settings menu) in the first round of playtesting.

w/b 12/2/24 Finished implementing settings UI (for the list of settings I created last week) as well as creating the 8 levels the player can play through to learn the controls of the game. I found that the solution I had for the settings UI was really hard to scale (i.e. add more settings), so after some thought I decided to refactor the backend for the settings UI. I used scriptable objects so that settings can be stored across scenes, and then PlayerPrefs (a class in Unity) for storing player customisations.

w/b 19/2/24 Finished work on the settings UI refactoring from last week. Then added gamepad support for the Player Character as well as the settings UI. In doing this, I encountered a problem with the frontend. The current solution required players to use a scrollbar to scroll through the list of available settings: The gamepad UI navigation system provided by Unity did not like this. As a result, I had to change the frontend so that a scrollbar would not be necessary.

w/b 26/2/24 Spent the week focused on re-designing the tutorial levels. I felt that the current solution I had was rushed when I developed it, so instead I spent more time thinking about what I'd like the player to learn in different levels, and how that learning progresses. I also added text on-screen so the player is aware of new controls.

w/b 4/3/24 Created User Test 11, focused on the usability of the UI as well as the effectiveness of the tutorial for teaching the player the controls. Then spent the rest of the week promoting the playtest.

w/b 11/3/24 Continued to promote User Test 1, while also focusing on refactoring the online functionality - this included re-designing the front end for creating and joining lobbies.

w/b 18/3/24 Started implementing online voice and text chat using Unity's Vivox service. Also briefly looked at results from User Test 1. Will need to spend more time on this though and make a proper evaluation of the results.

w/b 25/3/24 - Submission Began writing first draft of the final report, alongside finishing implementation of online voice and text chat.

w/b 1/4/24 Continued writing first draft of the final report.

w/b 8/4/24 Continued writing first draft of the final report.

w/b 15/4/24 Sent first draft to supervisor and received feedback. Also conducted User Test 2 which involved testing the game's online functionality and receiving qualitative feedback.

w/b 22/4/24 Wrote second draft of the final report.

w/b 29/4/24 - **Submission** Sent second draft to my supervisor and received feedback. Wrote third draft and submitted.

D User Testing Compliance Form

See next page.

User Testing Compliance Form for UG and PGT Projects*
School of Engineering and Informatics
University of Sussex

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

Prior to conducting your project, you and your supervisor will have discussed the ethical implications of your research. If it was determined that your proposed project would comply with **all** of the points in this form, then both you and your supervisor should complete and sign the form on page 3, and submit the signed copy with your final project report/dissertation. Note points are written in past tense, as you are effectively confirming, once your project has been completed, that it was conducted in a way that complied with all of the points.

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

---

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

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

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

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

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

---

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

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

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

4. No incentives were offered to the participants.
   The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle. People volunteering to participate in research may be compensated financially e.g. for reasonable travel expenses. Payments made to individuals must not be so large as to induce individuals to risk harm beyond that which they would usually undertake.

5. No information about the evaluation or materials was intentionally withheld from the participants.
   Withholding information from participants or misleading them is unacceptable without justifiable reasons for doing so. Any projects requiring deception (for example, only telling participants of the true purpose of the study afterwards so as not to influence their behaviour) are deemed high risk and require approval from the relevant C-REC.

6. No participant was under the age of 18.
   Any studies involving children or young people are deemed to be high risk and require ethical approval through the relevant C-REC.

7. No participant had a disability or impairment that may have limited their understanding or communication or capacity to consent.
   Projects involving participants with disabilities are deemed to be high risk and require ethical approval from the relevant C-REC.

8. Neither I nor my supervisor are in a position of authority or influence over any of the participants.
   A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any study.

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

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

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

11. The evaluation was described in detail with all of the participants at the beginning of the session, and participants were fully debriefed at the end of the session. All participants were given the opportunity to ask questions at both the beginning and end of the session.
    Participants must be provided with sufficient information prior to starting the session, and in the debriefing, to enable them to understand the nature of the investigation.

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

Project title: 3D Game Development

Student's Name: Benjamin Peter Conway

Student's Registration Number: 22101609

Student's Signature:

Date: 24th April 2024 (24/04/2024)

Supervisor's Name: **Dr Kingsley Sage**

Supervisor's Signature:

Date: **25 April 2024**

E Project Proposal

See next page.

3D Game Development - Producing an Accessibility-Driven Competitive Online Multiplayer Title

University of Sussex
Candidate Number 246535
Supervisor Dr Kingsley Sage
October 2023

What makes a multiplayer game difficult? Is it when developers implement control schemes that require an extensive tutorial just to know the right button combinations? Is it a steep learning curve, that divides the player-base into 'beginner' and 'expert'? Perhaps the game becomes too challenging due to it not accommodating for the player's physical impairments, or limited hardware due to their financial situation?
This project will place a primary focus on understanding how to design, develop, and test an online multiplayer game that offers equal accessibility for all players.

1 Aims

- Discover what accessibility-related tools and design ideas are currently available in the industry.
- Understand how to design game mechanics that feel intuitive for the player.
- Explore [and use] methods that test the effectiveness of accessibility features.
- Repeatedly improve the game's accessibility (when necessary).

2 Objectives

2.1 Primary

1. Research and evaluate the accessibility features currently available in a range of similar titles to assist Primary Objective 4.
2. Determine the balance between teaching the player how to interact with the game, and letting them develop their own understanding of the interface/mechanics.
3. Interview at least 5 individuals who play competitive multiplayer video games to understand what features they most enjoy (as well as what they don't enjoy), and to assist in completing Primary Objective 4.
4. Use the information gathered in Primary Objectives 1-3 to produce a specification for the software.
5. Use the specification in Primary Objective 4 as a guide to the software's design (and subsequent development), including gathering functional and non-functional requirements.
6. Research human-computer interaction techniques to assist the design of the software.
7. Use agile development to give the flexibility necessary to make continuous improvements.
8. Create multiple prototypes with increasing complexity during cycles of design and development.
9. Implement offline and online functionality.
10. Ensure that all accessibility features can be used, for all offline and online functions.
11. Design and implement a basic player agent (A.I.) that can be used in offline matches.
12. Research a range of popular methods to test the effectiveness of accessibility features.
13. Record and evaluate player testing during cycles of the development phase to determine the effectiveness of accessibility features.

2.2 Extensions

1. Extend Primary Objective 11 so that the player agent can be used in online matches.
2. Extend the online functionality in Primary Objective 9 to include text and voice communication between players.
3. Use a large, varied base of play-testers to yield more accurate results in demonstrating the effectiveness of the the game's accessibility features.
4. Ensure the cycles strictly adhere to the agile methodology, where each cycle is given a deadline and focuses on a specific set of features to implement.

2.3 Fallback

1. Scale back Primary Objective 9 to only have core online functionality (i.e. managing connection/disconnection to server).
2. Create a list of 'crucial' and 'optional' accessibility features, then ensure the crucial features are implemented.

3 Relevance

I have had numerous assignments over the past two years of my course that have taught me how to manage a project centred on delivering robust software: the skills developed as a result of completing these assignments can no doubt be transferred into this project, in addition to the knowledge gained in the accompanying lectures and seminars. Evidence of an understanding of modules such as further programming, human-computer interaction, computer networks, and software engineering, will be seen in this project.
There is also a personal element attached to this project, as I used to compete for Sussex University's eSports team and plan to enter the video game industry after graduation. I have spent time outside of the courses' modules learning about how to create games with the physics-based Unity engine. Not only will the completion of this project give me a title which I can add to my portfolio and show to employers in the future, but the research into accessibility will be of equal value as the industry sees a growing demand for this technology to be implemented into upcoming games.

4 Resources Required

- Unity Personal licence to develop software using the Unity game engine
- Personal computer and access to lab computers for development
- May require funding to purchase the licence of existing assets that can aid the development of the software
- May require funding to incentivise play-testers

5 Timetable

Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
9:00–10:00	Coursework	Lecture	Coursework	Coursework	Coursework	Coursework	—
10:00–11:00	”	”	”	Lecture	”	”	—
11:00–12:00	Lecture	Coursework	Lab	Coursework	”	”	—
12:00–13:00	”	Lunch break	”	Lunch break	Lunch break	Lunch break	—
13:00–14:00	Lunch break	Lecture	Lunch break	Lab	”	”	—
14:00–15:00	Coursework	Coursework	Coursework	Workshop	Coursework	Coursework	—
15:00–16:00	Seminar	”	”	Coursework	”	”	—
16:00–17:00	Coursework	”	”	”	”	”	—

Table 1: Weekly Timetable (Monday to Sunday) *NB: Coursework includes the Individual Project*

6 Bibliography

1. Anderson, S. L. (2023). *Video Game Accessibility Defined Through Advocacy: How the Websites AbleGamers.org and CanIPlayThat.com Use the Word Accessibility.* Games and Culture, 0(0) [Online] Available at: Click here. [Accessed: 18th October 2023]

2. Benvenuti D, Ferro L S, Marrella A and Catarci T (2023) *An Approach to Assess the Impact of Tutorials in Video Games Informatics* 10 6 [Online] Available at: Click here.

3. Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994). *Design patterns: elements of reusable object-oriented software.* Boston: Addison-Wesley. Available at: Click here.

4. Isbister, K. and Schaffer, N. (2008). *Game usability: advice from the experts for advancing the player experience.* Amsterdam: Elsevier/Morgan Kaufmann. Available at: Click here.

5. *League of Legends* (2009). Riot Games Inc, USA.

6. Mark Brown (Game Maker's Toolkit) (2021). *How Accessible were 2021's Games?.* 10th December. Youtube. [Online Video]. Available at: Click here. [Accessed: 18th October 2023]

7. Mark Brown (Game Maker's Toolkit) (2020). *How Accessible were 2020's Biggest Games?.* 11th December. Youtube. [Online Video]. Available at: Click here. [Accessed: 18th October 2023]

8. Mark Brown (Game Maker's Toolkit) (2019). *How Accessible were 2019's Biggest Games?.* 22nd November. Youtube. [Online Video]. Available at: Click here. [Accessed: 18th October 2023]

9. Mcconnell, S. (2004). *Code complete.* Redmond, Washington: Microsoft Press. Available at: Click here.

10. Microsoft (no date) *Gaming accessibility fundamentals.* Microsoft. [Online]. Available at: Click here. [Accessed 18th October 2023]

11. *Overwatch 2* (2022). Blizzard Entertainment Inc, USA.

12. Preece, J., Rogers, Y., & Sharp, H. (2015). *Interaction design: beyond human-computer interaction (Fourth edition.). John Wiley & Sons Ltd* - Click here.

13. *Rocket League* (2015). Psyonix LLC, USA.

14. Salen, K. and Zimmerman, E. (2004). *Rules of play: game design fundamentals.* Cambridge, Mass. The Mit Press. Available at: Click here.

15. Schell, J (2019). *The Art of Game Design : A Book of Lenses, Third Edition*, CRC Press LLC, Milton - Click here [Accessed 18 October 2023].

16. Scope (no date) *Accessibility in gaming report.* Scope. [Online]. Available from: Click here [Accessed 18th October 2023].

17. Unity Technologies (no date) *Gaming Accessibility.* Unity Technologies. [Online]. Available from: Click here [Accessed 18th October 2023].

18. *Valorant* (2020). Riot Games Inc, USA.

19. Y. Liu *Human-Computer Interface Design Based on Design Psychology*, (2020). IEEE [Online] Available from: Click here [Accessed 18th October 2023].

7 Interim Log

Meeting #1 (11.10.2023):
Introductory meeting outlining the key details we need to know for the individual project.