Major Project Thesis - Intuitive Level Design in Game Development

Ben Kershaw N00221365

Declaration

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

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

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

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

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

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

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

DECLARATION:

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

Student :

Signed 

Date_____01/05/2026_____

Acknowledgments

Before beginning I would like to thank the people who helped to contribute to this project, without their support this project would not have turned out as well as it did.

Firstly, I would like to thank my project supervisor: Joachim Pietsch and my second reader John Dempsey. With their help, I was able to elevate this project to a level I would not have reached otherwise. Their continued support, guidance and assistance was critical in the completion of my thesis project.

I would also like to thank my family and my girlfriend Angelina for their constant support and love not only throughout this project, but always. Thank you all for always pushing me to be the best I can be.

Abstract

This thesis focuses on “intuitive level design”, and how it was implemented into my game “Species Sprint”. Production of this game was completed in Unity with C#. The intention with this project was to create a game with level design that was intuitive to the player, meaning that they would be able to understand core aspects of the world design without it needing to be explained to them. They should be able to understand the core concepts of the design intuitively.

This type of level design is quite theoretical, borrowing concepts from shape theory, colour theory and abstract lighting concepts. All of these were researched prior to the start of this project and then implemented accordingly.

The broad description for “Species Sprint” is that it is a racing game in which the player must choose an animal to play as, each with their own pros and cons, and then complete a race. Each race consists of three laps around whichever course the player chooses, although for the purposes of this thesis only one sample level was created.

This project will be a success if the level design concepts mentioned above are implemented effectively not only in my opinion, but in the opinion of the testers employed to experiment with the game. They will be asked to each complete a sample race and their opinions will be recorded and used to alter the course of development in the latter half of our assigned time.

Contents

Chapter 1: Introduction

1.1 Project Aim & Objectives

To design a video game there are many factors a developer has to take into account. What is the graphical style of your game? What are your game mechanics? Is it 3d or 2d? One of these key considerations a developer will have to put a significant amount of thought into is about the level design of their game. Jessica Plowman, a lead game developer at Epic Games, describes level design as the developers ability to create a level for a game which "*must be able to be read by the player in a short amount of time while still providing an entertaining experience*"(Plowman, 2026)

This project investigates the research question: *“What are the elements of good level design?”*. On the surface, this may seem like a simple question but there are many layers to this that the passive observer may not take into account.

Ultimately, the aim of this project is to create a fully functioning game while utilizing the design elements and research conducted in preparation for this project. The game should be centered around its highly stylised, intuitive and comprehensible level design.

The game that is being developed for this major project is called “Species Sprint”. It is a game where you will compete in a 3 lap race, while playing as and against a selection of different animals. Each of these animals have different pros and cons, such as different abilities and speeds. It is the players goal to complete a race as quickly as possible, making sure they are the first to cross the finish line.

1.2 Success Criteria & Scope

This project will be considered a success assuming the following conditions are met:

- The game is fully functional & playable
- Best practice level design elements are included
- These design elements reflect positively with testers
- Tester feedback is implemented effectively
- This report reflects the research conducted and implemented for the project

In the context of this project, as much as any other, it is critical to maintain a realistic scope of what is achievable within the set time frame. The way this was managed within this project is through the SCRUM methodology. This is a concept which originates from rugby, with the idea of the developer splitting all the tasks required to complete their project into “sprints”, each typically being 1-2 weeks long. Within each of these sprints, there will be a small list of tasks that the developer expects to get done within that timeframe. You can see my sprints below in Figures 1&2:

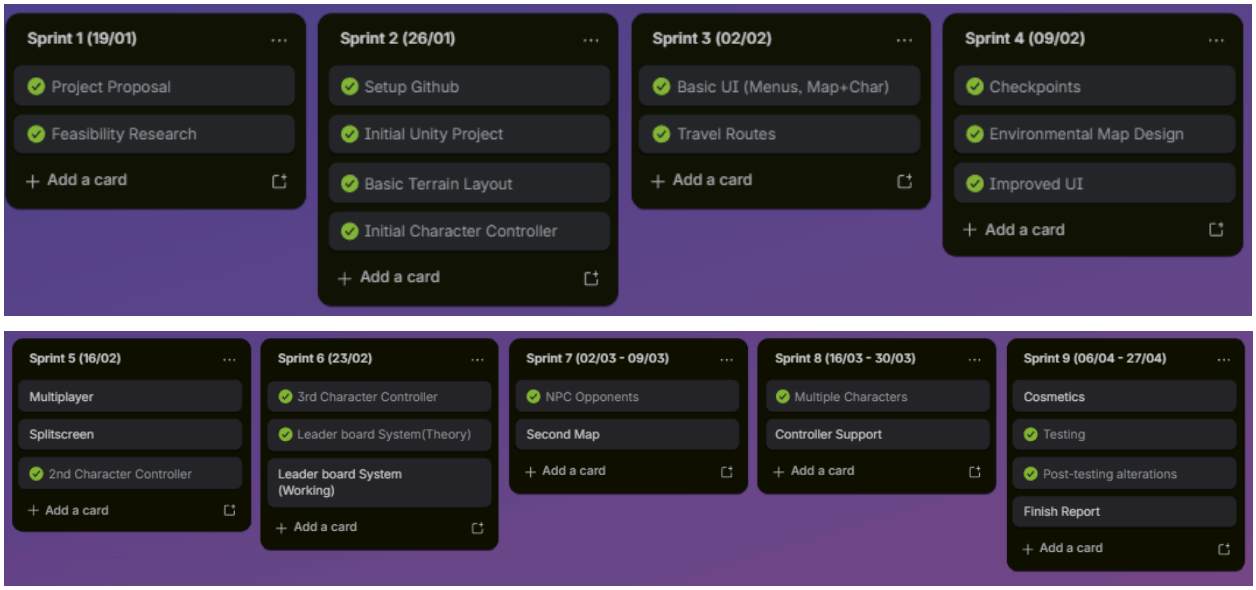

Figure 1 & Figure 2 - Trello Board for thesis project

You can see all of the tasks required to complete the project above. As you can see some of these are not done but that will be discussed further later in the report.

Chapter 2: Research & Background

2.1 Introduction

This section of my paper will be discussing many different concepts which will enhance the players experience while engaging with video games. The idea behind this project is that by learning the theory behind these concepts now, they will be applicable to future projects, more specifically this thesis project which will be completed alongside this report. All going well the end product will be one crafted with the skills and theoretical knowledge necessary to create a highly effective level and that this report will explain its effectiveness through industry standard concepts and examples from highly successful titles and publishers from within the gaming industry.

2.2 Elements of Effective Level Design

As mentioned previously, there are many different concepts that go into designing a level for a video game. A lot of these will come down to the genre of the game. For example, a shooter game should have a level that allows for strategic gunplay whereas a driving simulator will typically have a wide open level allowing the player to take advantage of the speed of the vehicle. However, there are a few fundamentals that will hold true across all genres, only they are manipulated in order to fulfill the expectations that genres will demand. These being:

- Lighting
- Shape Theory
- Colour Theory

These four concepts allow the developer to fully manipulate the player's perception of the level. Two levels that are functionally exactly the same but lacking consistency in the above concepts will ultimately confuse the player and lead to a negative experience. Throughout the rest of the research section of this report, all four of these concepts will be discussed in detail, explaining the inner workings of each of them, listing what are both good and bad examples of both in existing media as well as providing quotes and insights from industry professionals regarding how these aspects of game development should be implemented.

2.3 Lighting

Lighting in all media, but in this case specifically video games, is the process of placing light sources of various intensities and hues at specific locations and angles in order to convey information to the player. This can be used to convey a narrative meaning or perhaps to lead the player in a certain direction.

2.3.1 Wayfinding Lighting

In his book “The Level Design Book”(Yang, 2025) Robert Yang discusses the concept of “Wayfinding” lighting. This is the concept of using lighting to direct the player to move to a certain location or along a certain route. In their article “The Psychological Impact of Light and Colour”(Kaplan & Kaplan, 2017) the company TCP Lighting explains the reasons humans will naturally gravitate towards light. They say:

“This is where lighting can come in. It can be used to highlight building elements, spaces, paintings, textures, etc. that people may find familiar.”

“ Highlighting areas of an environment to help draw people’s attention can help them find familiar objects, spaces and structures.”

In the case of this article they are discussing how their lighting company sets up lights when renovating offices and homes, but the same logic naturally holds true for games too. In an otherwise dark room, it would only make sense to walk towards the most visible point in the room.

There are some very effective examples of this in some of the most popular video games of recent years. One example that comes to mind is in a game called “Call of Duty: Black Ops 2”. In Figure 1 below, you will see an example of this.

Figure 2 - Example of Wayfinder lighting in Call of Duty: Black ops 2

The beam of light you see shooting into the sky in Figure 1 is indicating the location of the “Mystery Box”. Its purpose isn't important for the sake of this example but just know that it is a core component of the game that the player will almost certainly need to go to at some point. Now, the player is never explicitly told the location of the purpose of the Mystery Box, however

they are under the assumption that they should go towards the beam of light they see in the sky. This is a prime example of Wayfinding Lighting.

Another example can be seen below in Figure 2. This example comes from the critically acclaimed game “Dark Souls 3”:

Figure 3 - Example of Wayfinder lighting in Dark Souls 3

Some may take issue with the fact that it is very hard to see anything in Figure 2, however that is the point of this level. In Dark Souls 3, those small glowing orbs are what indicate the location of items the player can pick up to aid them on their journey, so naturally they will run towards it to gather it. However, the developers are expecting the player to do this and in a subversion of expectations, going within a radius of this item will trigger a boss fight. In this instance, the developer is using Wayfinder lighting and its instinctual influence on the player to their advantage to play a trick on the player and the first time a player experiences this remains one of the most memorable points of the game.

2.3.2 Narrative Lighting

Another way lighting can be used in video games is to display narrative meanings. This could be many things, though usually it is used to display a character or location's morality. A fantastic example of this can be found in Figure 3 below, this image being from the game “Little Nightmares 2”:

Figure 4 - Example of Narrative lighting in Little Nightmares 2

In Figure 3 above we see the player character “Mono”(right) greeting a new character called “Six”(left). The lighting in this scene displays a clever piece of foreshadowing that most would not pick up on their first playthrough. For almost all of the game, Six follows the player on their adventure and is very much an ally, only to betray them in the final scene of the game. With that in mind, let's take a look at this image from a narrative lighting perspective. Mono being the protagonist and by extension the “good guy” in this game is represented here by how the light is shining on him, he is the focus of the game and is shown as such. He is also trying to help Six throughout the course of the narrative, literally pulling her out of the dark place she was in. In contrast to this, Six is presented fully in the dark, only starting to come into the light when coaxed by Mono. This small detail lets the characters true intentions be shown from very early on in the game.

2.4 Shape Theory

As defined by Nahzanin Shahbazi in her article ”Shape Language: What It Is and Why it Matters!”[4]**:**

“Shape language involves deliberately using geometric forms, such as circles, squares, and triangles, to express emotion, meaning, and character traits.
In animation and game design, these shapes are visual cues that audiences instinctively understand.”

In other words, shapes have an inherent meaning to people, something that every person naturally understands without having to be told. This concept can then be implemented into game design in order to direct the player without having to explain certain elements of the game to them. This section of this report will discuss the three most common shapes to which this theory applies, (squares, circles and triangles)and how we perceive it, as well as examples of how it is used in some popular games.

2.4.1 Square

As per Nahzanin**:**

“Squares symbolize strength, stability, and trustworthiness. Their even sides and solid forms convey a sense of security.”(Shahbazi, 2025)

With this in mind we can start to inspect how making certain elements of your level design using square components can elicit certain reactions and motivations in the player. With squares being used to display stability and power, below is an example of how other developers use these ideas:

In Blizzard Entertainment’s game Overwatch, the player can choose the character they want to play as from a large roster of heroes. One of these heroes the player can play as is a character called Reinhardt. In Overwatch, every character has a gimmick, something that makes them unique. In Reinhardt’s case, this is his large shield, as seen in Figure 4:

Figure 5 - Example of squares being used with shape theory in Overwatch

As you can see in Figure 4 above, Reinhardt's shield is a large square. Now there are other shield users in the game with varying shapes and sizes relative to the character but Reinhardt's being this shape is very important to the games design. You see in Overwatch, characters can be classified into three main categories:

Damage: Focused on damaging players on the other team
Tank: Focused on protecting their own team from incoming damage
Support: Focused on providing aid to their team, typically via healing or buffs

Particularly in the early days of Overwatch, Reinhardt was considered by most players to be the quintessential tank, mainly for his value gameplay wise but by looking at his design it is clear why many would perceive him like this. As mentioned earlier, Squares convey power and stability and this is exactly what Reinhardt needs to be. So even on their first time playing, players will instinctually know that they should rally around this character for cover.

2.4.2 Circle

In her article titled "Shape Language in Character Design"(Daineko, 2025) Anastasia Daineko calls circles**:**

"- like a warm hug. Their smooth, endless curves make us feel cozy and safe, like a bubble or a full moon."

Circles typically convey a much softer image than squares, stable but in a much different way. They are often used to show something is bouncy or otherwise safe to come into contact with. There are many examples of this in games, but below is one which illustrates this very well:

Figure 6 - Example of circles being used with shape theory in Mario Kart Wii

Figure 5 shows a level from Mario Kart Wii, titled Mushroom Gorge. In it, the players race their karts around this map, with the theme of this particular map being that it takes place in a sprawling valley filled with these enormous mushrooms, as seen in Figure 5. Now, the game never explicitly tells the player this but if you were to ride off that wooden ramp over the gorge and land on the mushroom, the player would discover that they are in fact bouncy and act as a shortcut! This would not be so intuitive had they used an object of another shape such as a square or triangle, as these shapes portray negative feelings in terms of making progress.

2.4.3 Triangle

Triangles are characterized by their points. This often resonates as dangerous or sharp to people, given that in video games, triangles are typically associated with things like swords and spikes. The player will instantly be wary of going near anything triangle shaped for this reason. Some examples I can offer are:

Figure 7 - Example of triangles being used with shape theory in Hollow Knight

Figure 8 - Example of triangles being used with shape theory in Bloodborne

Figures 6 & 7 above illustrate two examples of triangles being seen as threatening to the player in games. Figure 6 comes from a game called “Hollow Knight”. It sees the player exploring a vast underground kingdom. Upon reaching this section, the player will simply know not to make contact with this portion of the stage because triangles are seen as sharp and threatening, whereas if these were say circles the player would most likely be less apprehensive.

Figure 7 shows “The Moonlight Greatsword” from “Bloodborne”. Any number of swords or other weapons could have been used for this example but this image was used as it has the most defined shape for the purposes of making this point. Its large triangular shape tells the player this weapon is dangerous and most likely a very high damage item, again as opposed to something more circular.

2.5 Colour Theory

Without most people even knowing it, colour theory dominates almost every aspect of our lives. It is defined by Klara Pranke and Theodore Jensen in their paper “Colours and Context” (Plowman, 2026) as:

“Colour can be considered a non-linguistic visual stimulus that can symbolically communicate information.”

This is to say that nearly every person regards certain colours to have intrinsic meanings and emotions that are attached to them. Most people will subconsciously pick up on these colours' meanings without even realising and this reaction can be exploited for the purposes of level design and game development as a whole.This paper will not discuss the meanings attached to every colour but below you will find listed some of the most prominent colours, their meanings as well as examples of how they can be used in games.

2.5.1 Red

Red is the colour of passion, typically this being violent passion or loving passion, depending on the context. It is quite a striking colour and is often used as the primary colour of a scene/component of a scene. A game developer could use this to display a character's intentions towards the player. A quite on the nose example of this can be seen below:

Figure 9 - Example of red being used with colour theory in Angry Birds

In Figure 8, the main character of the game Angry Birds can be seen. He is also the logo and mascot of the game. His red hue is no coincidence as considering he is the mascot, he needs to be the epitome of anger. He is quite literally “red in the face” with anger here.

Another idea which the colour red can relay, which more so relates to level design, is heat. A level being shown in predominantly red/orange hues will often signal to the player that that area is quite hot and that they will most likely take damage if they interact with it. This can be seen in the below example:

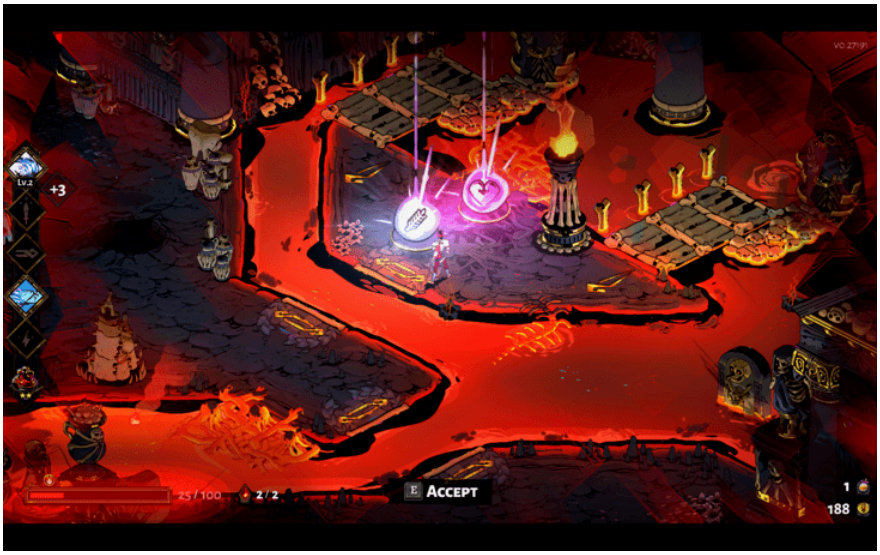

Figure 9 - Example of red being used with colour theory in Hades

Figure 9 above shows the second level of the game “Hades”. This is a level dominated by large lakes of lava that will damage players upon contact. The player is not told this but given the context of the level and colour choice, they know to try not to interact with it.

2.5.2 Blue

Blue is the colour of calmness and peace. Blue locations in video games will typically be places of tranquility and perhaps to take a break. In level design, by far blue’s most relevant inclusion comes in the form of water. This simple hue difference is the reason we are able to differentiate water in a game such as Subnautica as opposed to the lava seen in Figure 9.

Figure 10 -Example of blue being used with colour theory in Undertale

In Figure 10, we can see the “Snowden” level of “Undertale”. Although the material covering the ground is supposed to be snow, typically a white material, the developer has instead chosen to give it a blue tone. It can be assumed that this stylistic choice was done in order to make the area seem more calm, as it is one of the game's opening levels and the developer wants to ease the player in.

2.6 Conclusion

To conclude this section of my report, I would like to recap everything that has been studied and how I intend to implement each of these concepts into my thesis project.

2.6.1 Lighting

From this section of the report you can see just how important lighting is in the development of any game, from directing the player to a narrative perspective, there are many different ways in which the developer can use lighting to their advantage.

The intention for this project is to use lighting as a tool to direct the player to certain key locations and items without having the use of a map. Assuming this is done properly, the intended route for the player should be quite obvious despite the lack of hands on guidance

2.6.2 Shape Theory

Shape theory shows how important something as small as shapes can be when designing your level and how you need to heavily consider these elements when engaging in level design.

This project should lean into these design tropes as much as possible ensuring the user will have an understanding of how to safely navigate the world.

2.6.3 Colour Theory

Colour theory will be kept in mind at all times when designing all the elements of the world. From letting the player know which sections are safe and which aren't, to helping to make the world feel more alive, colours will be doing a lot of the heavy lifting in all aspects of the game.

Chapter 3: Requirements Analysis

3.1 Requirements Gathering

Part of building a game that people will enjoy playing is looking at games that are similar to yours to understand what the competitors are doing right and wrong. For this project, this section of the report will be discussing one of the most iconic racing games to better understand what users enjoy about these kinds of games and then implementing it into this project.

3.1.1 Similar Applications

Mario Kart 8 is the most sold game on the Nintendo switch with an astonishing 70 million copies sold to date (Nintendo, 2025). When people think of a racing game, the Mario kart series is typically what comes to mind for most, and it is clear to see why. Mario Kart games, in particular the more recent ones, are all incredibly entertaining and very well polished, making them the quintessential racing game. But what is it about them that players seem to enjoy so much? The following seem to be the most common praises people give to the games:

- Recognisable Characters
- Fun Track Design
- Intuitive and Responsive Controls
- Appealing Visuals
- High Speed Racing
- Overall Polish

All of these seem like very basic critiques of the games at first, but Nintendo has spent years earning this acclaim, with every entry in the series improving on the last. They have laid the blueprint for what many modern racing games do and this project will be no different. Naturally, there cannot be any recognisable characters in this game. The rest of these opinions though are quite realistic to implement. Taking inspiration from Mario Kart, users enjoy medium length, fast paced courses with twists, turns and obstacles to keep them engaged, quick movement with firm and responsive controls and finally, they want the maps they're racing on to look good.

3.2 Requirements Modeling

3.2.1 Personas

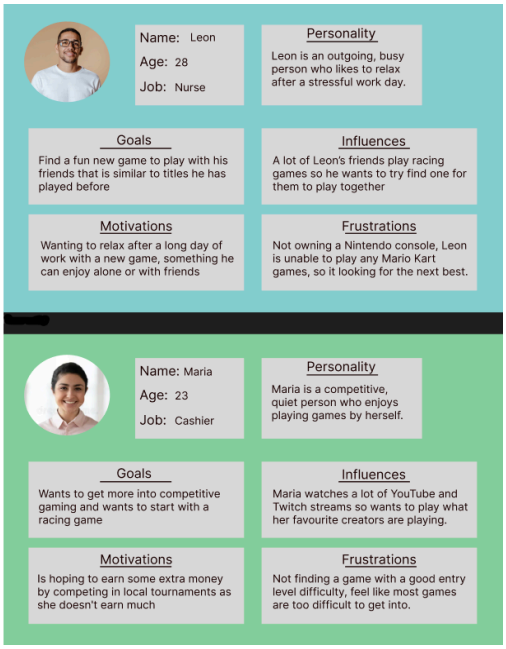

Figure 11 - Personas

Figure 10 shows the personas used for this project: Leon and Maria. Personas are tools that help developers figure out what kind of users will potentially be accessing their applications. In this can, the personas reflect people in their twenties looking for something new and fun to play after a day at their stressful jobs. It is very important to keep personas in mind when working on a project as being able to predict what your potential users will want out of your app will improve the design process and allow you to create an application that is more likely to be enjoyed by users.

3.1.2 Functional vs Non-Functional Requirements

The functional requirements for this project are as follows:

- Allows the player to navigate a series of menus to set up a race
- Contains a map with clear directions
- Allows the player to complete a three lap race
- Contains a checkpoint system to keep track of progress in a race
- Contains a “deadzone” system to keep players on the track
- Implements a character controller to allow the player to move
- Implements an opponent to race against, whether that be an AI or another player

The non-functional requirements for this project are listed below:

- Implements intuitive design choices, using previous research, into the level design
- Implements aesthetically pleasing design choices into the level design
- Provides useful information such as lap count, time and a minimap in the UI
- Includes choice of multiple characters with unique traits
- Implements sound design

Chapter 4: Design

4.1 System Architecture

Unique sections of a game in unity are divided into Scenes. Scenes are unique domains which contain all of the stages for a game. This system is used so that the application does not need to load all of the assets at once and slow down the RAM (Random Access Memory). The scenes present in Species Sprint are as follows:

1. Main Menu - The launch page of the game
2. Level Selector - Where the player chooses which level they want to play on
3. Character Selector - Where the player chooses which character they want to play as
4. Treetop Dash - The sample level made for this project on which players can race
5. Finish Screen - The screen that is displayed to the player on completion of a race

Typically, the player will move through these scenes in the above order, with some exceptions. Scenes are often moved between via buttons with code attached to them which are triggered under certain conditions. For example, in the Level Selector scene, the player is presented with a series of levels as seen in Figure 11 below. Each of these panels is a button and attached to them is a script telling Unity to move the player to the corresponding level, in this case clicking the button labeled Treetop Dash will change to the Treetop Dash scene.

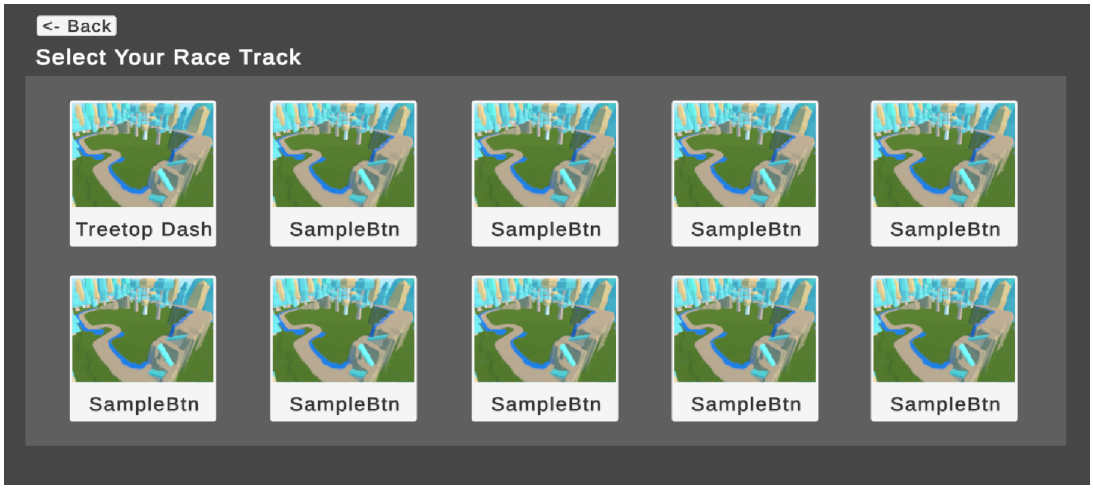

Figure 12 - Level Select page from Species Sprint

The other element of system architecture present in a Unity project is in the hierarchy. This system allows the developer to see a concise list of all of the assets present in the current scene, as well as allowing them to be grouped together. Having a well managed file system is crucial for any project as being able to quickly navigate your hierarchy will vastly speed up development time and minimise the amount of mistakes. In Figure 12 you will see the hierarchy for the main level of Species Sprint, Treetop Dash:

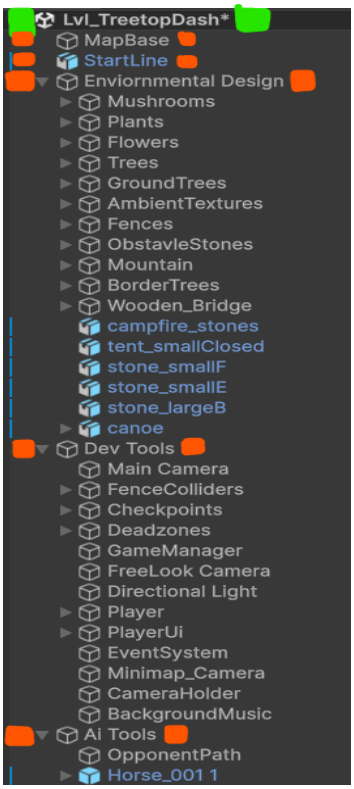

Figure 13 - Hierarchy in Unity for Species Sprint

The folder annotated with green boxes in Figure 12 is the top layer of the hierarchy, it contains all other other assets related to this scene. The folders annotated with orange boxes are the “general folders”. These are folders that contain all of the assets for a specific concept, such as all of the assets for the environmental design. Finally everything within these folders are the individual assets which make up the game world, such as trees and bridges.

4.2 Interface Design

In game development, interface design refers to visual elements of the game typically used to display information to the player or to perform an action. Examples of this would be a game's menus or the player's HUD(Heads-Up Display), which refers to the information being pinned to the screen during gameplay. Below you will find my design and implementation process of all of the interface elements of Species Sprint

4.2.1 Menus

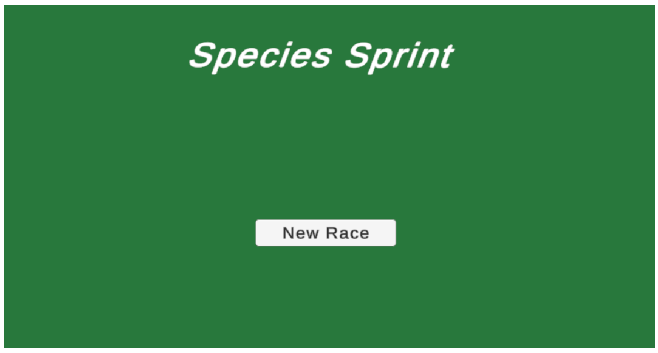

Figure 14 - Main Menu of Species Sprint

Above in Figure 14 is the Start Menu. THis is the page the player is placed on immediately after launching the game. It is very simple, only showing the button for a new race. Pressing this button moves you to the Level Selector screen. There is no reason to make this screen more complicated than it already is.

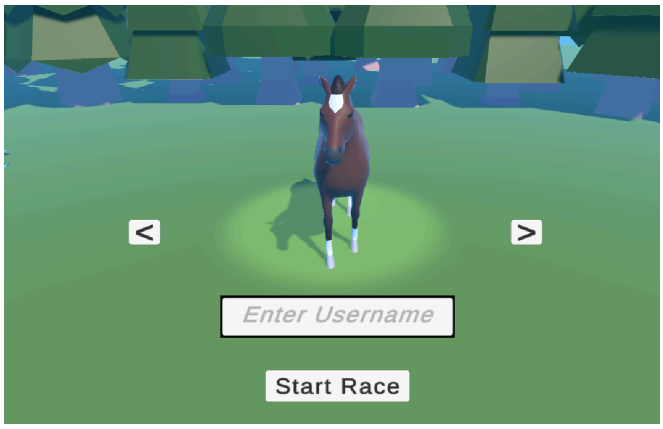

Figure 15 - Character Selection screen in Species Sprint

In Figure 15 you can see the Character Select screen. It should be noted that the Level Select screen takes place before this in the gameplay loop, however because it was already discussed in a previous section it did not seem necessary to include again here.

The Character Select screen is where the player chooses which character they want to play as. The arrow buttons on the left and right of the screen cycle through an array of the character models and the one the player is currently on is set as the active model. There is a text box where the player can set a username for themselves. This will be displayed above their head in the race itself. Finally, there is a button at the bottom of the screen which allows the player to move onto the race with the username and character they selected here.

Figure 16 - Race over screen in Species Sprint

In Figure 16, you can see the Race Over screen. This screen only displays once one of the racers has completed their third lap, ending the race. Here it displays how long it took you to complete the race, as well as giving you the option to restart another race or to go back to the main menu. There is also a prototype for a leaderboard system on this page, however this is non-functional in the finished project as, when doing research into how a leaderboard system in Unity would be achieved, it seemed too complicated for something that was quite far outside the scope of the thesis statement.

4.2.2 Player HUD

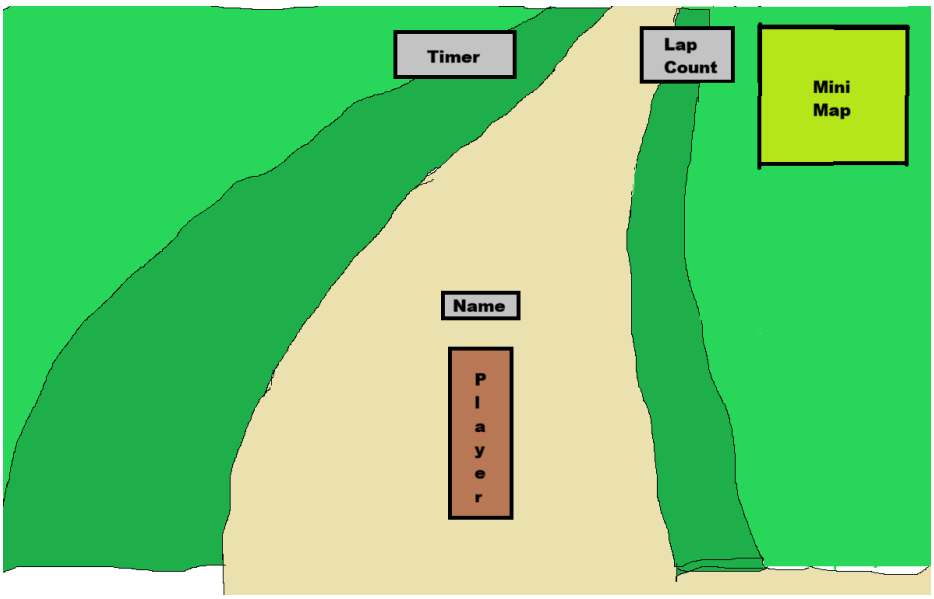

Figure 17 - Low-fidelity prototype of player HUD for Species Sprint

Figure 18 - Screenshot of player HUD in Species Sprint

Figure 13 & Figure 14 show the HUD system for Species sprint. In FIgure 13, you can see a low-fidelity prototype of what HUD may look like. The timer in the middle showing how long the course is taking you, the minimap on the right showing yours and other racers current positions and just to the left there is the lap count, to help you keep track of your progress.

Being satisfied with this design, you can then see how it was implemented into the game itself. The UI is simple, readable and informative. It is important to give the player all of the necessary information without overwhelming them and this was achieved here.

4.3 Level Design

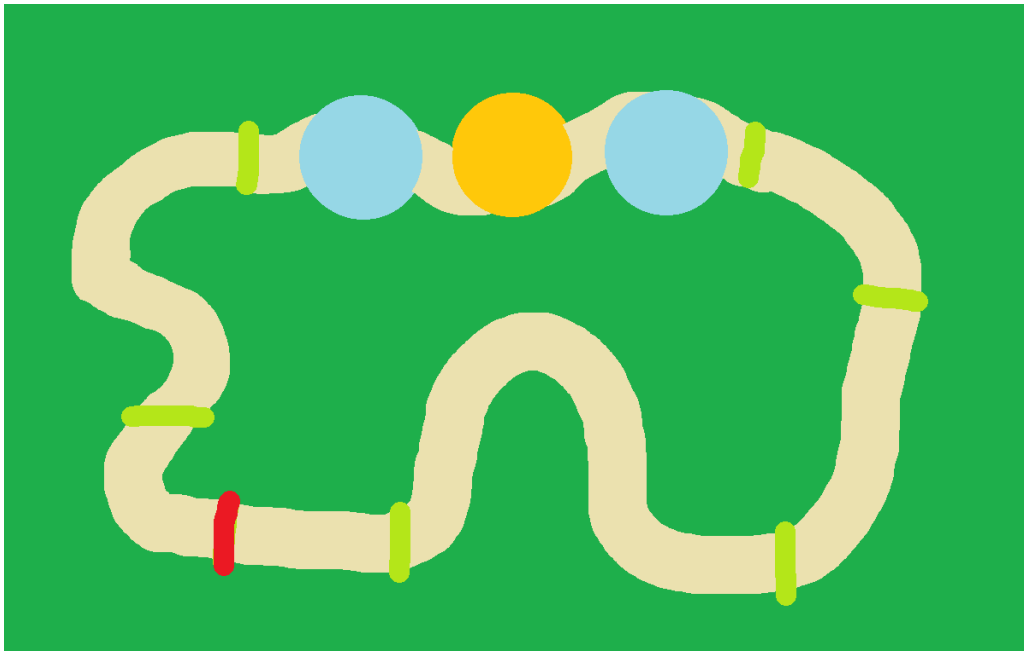
Figure 19 - Low-fidelity prototype of map design in Species Sprint

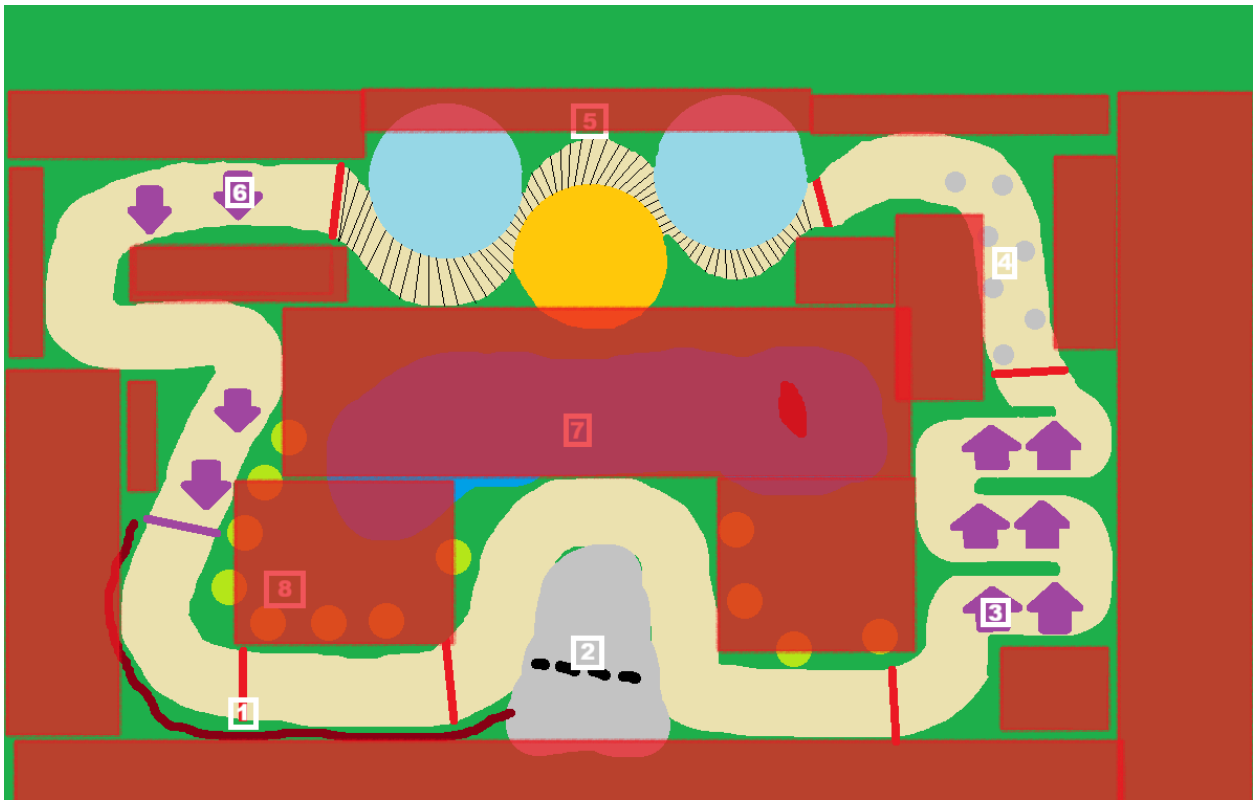

Figure 20 - High-fidelity prototype of map design in Species Sprint

Level design is the main focus of this project and is what the research portion of this thesis paper looked into. With that in mind, getting this portion of the game right was crucial. Above, in Figure 15 & Figure 16, you can see the high-fidelity and low-fidelity prototypes on the level design. You can see each of the level design elements numbered in Figure 16, these are all explained in detail below:

4.3.1 #1:Checkpoints

The Checkpoint system is what dictates whether or not the player is cheating and bypassing certain sections of the racecourse. The system works by assigning an integer to the player called checkpoint_count, which starts at 0. When the player passes through the first checkpoint, it checks if the checkpoint_count is set to 0. If it is, it increases the checkpoint_count to 1 and allows the player to progress. At the next checkpoint, it will see if the checkpoint_count is set to 1 and if it is, it increases it to 2 and allows the player through. This repeats for all of the checkpoints on the map until you reach the finish line. If your checkpoint_count is what the game expects, it is reset to 0 and your lap_count is increased to 1. After repeating this process another 2 times, assuming you enter the finish line checkpoint with the correct checkpoint_count and lap_count scores, the race will end and you will be moved to the RaceOverMenu scene. In Figure 17 below you can see an example of a checkpoint and in Figure 18 you will find a snippet of the code which provides the checkpoints their functionality.

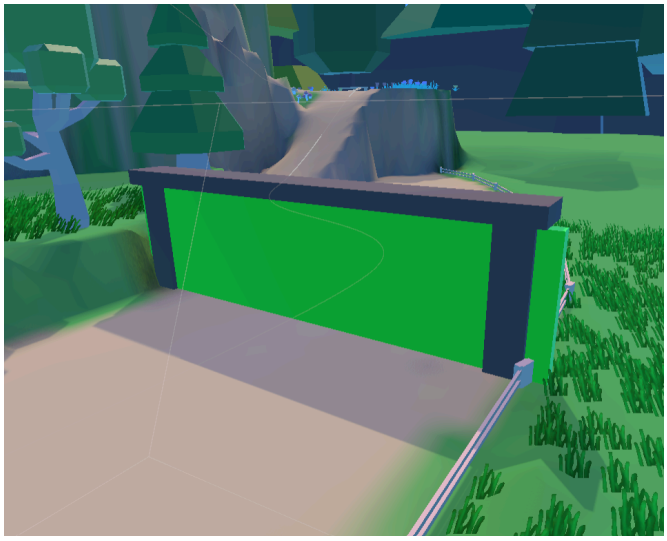

Figure 21 - Screenshot of checkpoint in Species Sprint

```
private void OnTriggerEnter(Collider other)
{
    if (other.gameObject == Checkpoint1 && currentCheckpoint == Checkpoint1_count)
    {
        currentCheckpoint++;
    }
    else if (other.gameObject == Checkpoint2 && currentCheckpoint == Checkpoint2_count)
    {
        currentCheckpoint++;
    }
    else if (other.gameObject == Checkpoint3 && currentCheckpoint == Checkpoint3_count)
    {
        currentCheckpoint++;
    }
    else if (other.gameObject == Checkpoint4 && currentCheckpoint == Checkpoint4_count)
    {
        currentCheckpoint++;
    }
    else if (other.gameObject == Checkpoint5 && currentCheckpoint == Checkpoint5_count)
    {
        currentCheckpoint++;
    }
    else if (other.gameObject == Checkpoint6 && currentCheckpoint == Checkpoint6_count)
    {
        currentCheckpoint++;
    }
    else if (other.gameObject == Finishline && currentCheckpoint == Finishline_count)
    {
        completedLaps++;
        Debug.Log($"Lap {completedLaps} Completed");
        currentCheckpoint = Checkpoint1_count;
        Debug.Log("Current Checkpoint: " + currentCheckpoint);
    }

    if (completedLaps == 3)
    {
        Debug.Log("Race Finished!!!");
        FindObjectOfType<Timer>().StopTimer();
        SceneManager.LoadScene("RaceOverMenu");

    }
```

Figure 22 - Snippet of code for checkpoint system functionality

4.3.2 #2: Mountain Shortcut

The mountain shortcut is placed near the beginning of the level and is a way for players with a keen eye to get a leg up. It is also the first example of using intuitive level design in Species Sprint. This shortcut can be seen in Figure 19 below:

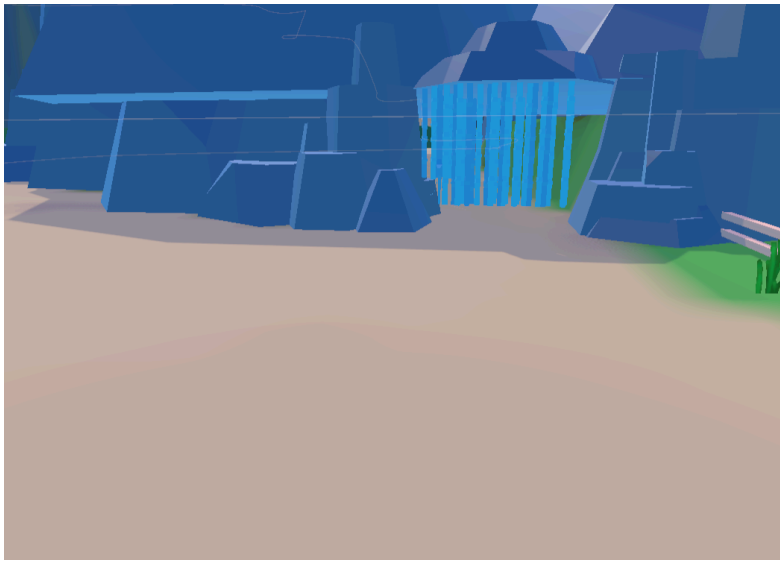

Figure 23 - Mountain shortcut in Species Sprint

The environment surrounding the entrance to the shortcut are quite dark colours. Dark greys were used and this portion of the map is deliberately not well lit. In contrast to this, the vines covering the entrance to the shortcut are quite bright and draw the eye. They were placed sparsely enough that the player will be able to see that there is an alcove behind them, hopefully prompting them to investigate.

4.3.3 #3: Elevated Ramps

Following the mountain shortcut referenced in Figure 23 and the second checkpoint. The player will encounter a set of three short hills to ascend. These can be seen below in Figure 24:

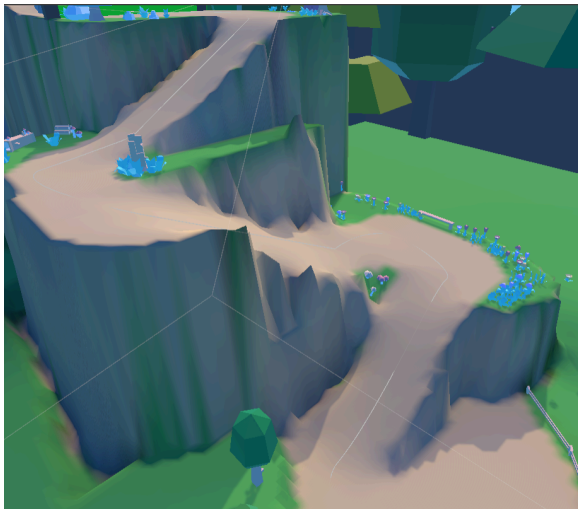

Figure 24 - Hills the player ascends in Species Sprint

Again here intuitive level design is being used to direct the player in a certain direction. The path the player is intended to follow is the beige/dirt coloured road, so in order to make sure they

don't go the wrong way, we have to interrupt that line. This was done by introducing grass patches throughout to make areas where the player is not supposed to go, which they will have learnt from the previous section where there were fences preventing them from accessing grass. Environmental design was also used here as on the grassy patches, obstacles such as benches, tents and flowers were placed in order to discourage the player from attempting to travel that way and, failing this working, they will interact with a “deadzone” (explained further in section 4.3.7).

4.3.4 #4: Rock Formations

At the top of the hill is a long, straight stretch of road. Originally this was empty and, as you will see in the user testing in section 6.2, testers did not enjoy this section of the map as they felt it was too empty and was the most boring section of the map. Naturally, this was very critical feedback and needed to be addressed. The rock formations (seen below in Figure 25) were added so that it isn't possible for the player to simply run in a straight line anymore. They have to duck and weave and also be careful not to bump into another racer who is travelling the same path, as they could get knocked off the edge if they aren't careful. This last minute addition has made what was originally a more boring section of the level much better and second time tester echoed this opinion.

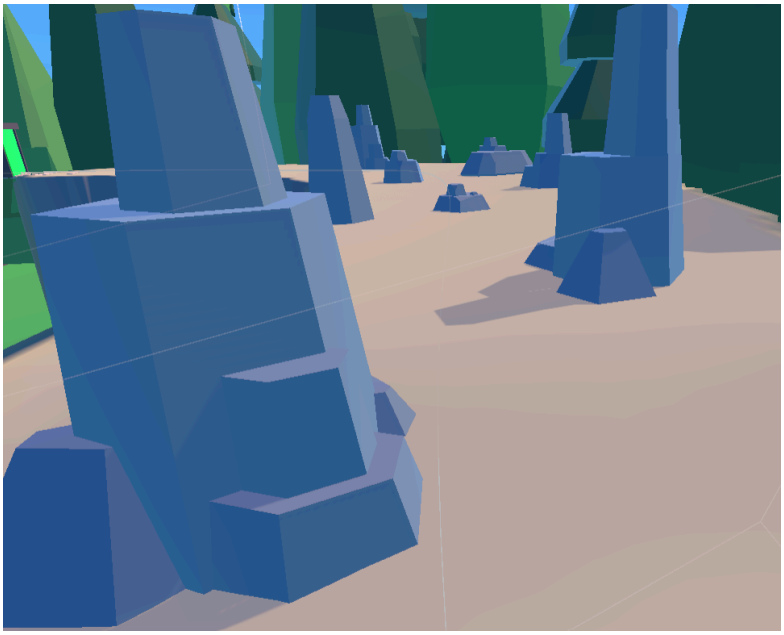

Figure 25 - Rocks the player must avoid in Species Sprint

4.3.5 #5: Treetop Bridge

The treetop bridge is the key feature of the map and its namesake. The player is able to see it through a gap in the treeline in the early stages of the level. It is a series of three large trees with a rope bridge suspended from them, which the players must run along. The bridge curls between the trees and is quite narrow, meaning players have to be careful not to fall off while traversing it. A shortcut is planned here by using the deer's planned ability which would let him

jump onto the branches of the trees for a shortcut (Note: This was not implemented, only planned). Below you can see the treetop bridge in Figure 26:

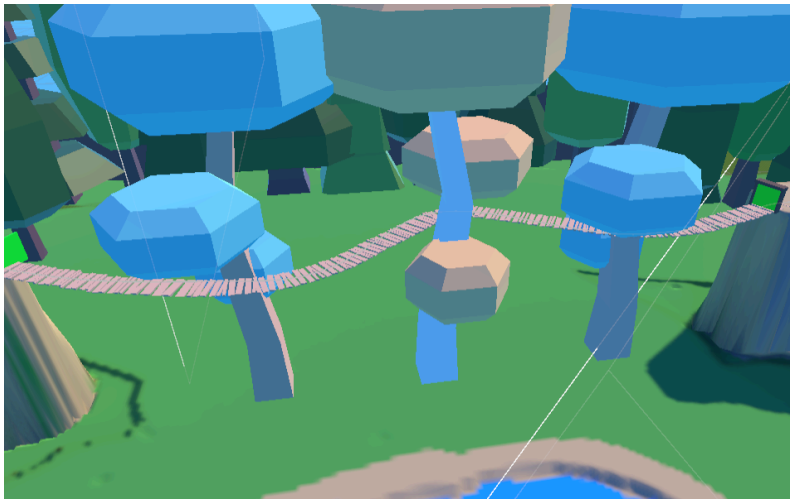

Figure 26 - Treetop Bridge in Species Sprint

4.3.6 #6: Downhill Slope

In order to get back to the finish line to complete a lap, players must descend from the height they gained from the hills in 4.3.3. This is done via another set of ramps. This time there are only 2 and they are much longer and will take the player a significant amount of time to complete. There is a possible shortcut here though. The edges on the first section of this ramp are quite spikey and can be used as a ramp to propel the player character to a lower section of the course. Doing this shortcut can be a massive time save if executed properly. Upon reaching the bottom of this hill, turning one more corner will lead the player back to the finish line, thus completing a lap. Below in Figure 27 is an image of the downwards slope.

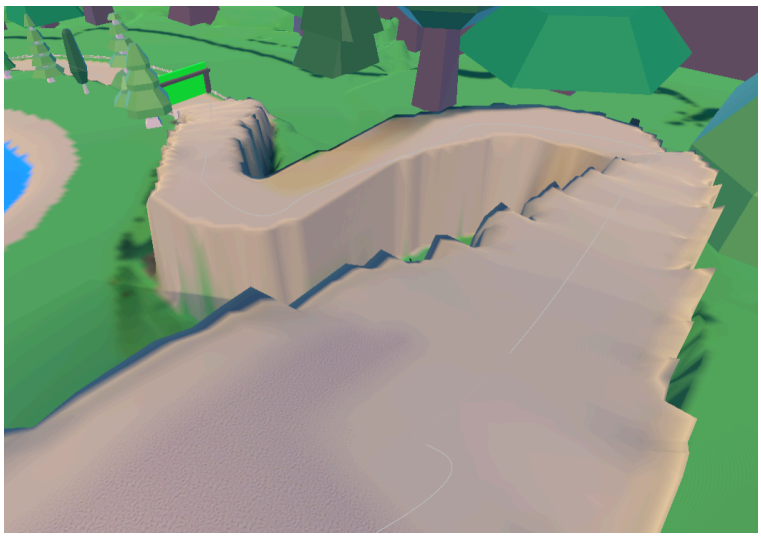

Figure 27 - Downwards slope in Species Sprint

4.3.7 #7: Deadzones

Deadzones are large invisible boxes surrounding the edge of the racetrack. The reason they are there is to prevent the player from leaving the racecourse. This is far from an original concept, with similar map exit detection being used in many racing games. The system works by placing these large boxes around the track and giving them the “deadzone” tag. In the code, it is established that if the player's collider interacts with anything with a “deadzone” tag, they are teleported to the last checkpoint they passed through. This information is retrieved from the checkpoints script by reading the value of currentCheckpoint. With these deadzones covering most of the map that isn't the track, this system is mostly foolproof, though it should be noted that during testing some players found gaps in the deadzones. This was later patched. Though they are invisible to the player during gameplay, the deadzones exist in the world as red boxes that are simply occluded from the player's camera. A rough diagram of their layout can be seen in Figure 20, however a screenshot of the game shows them properly in Figure 28 below:

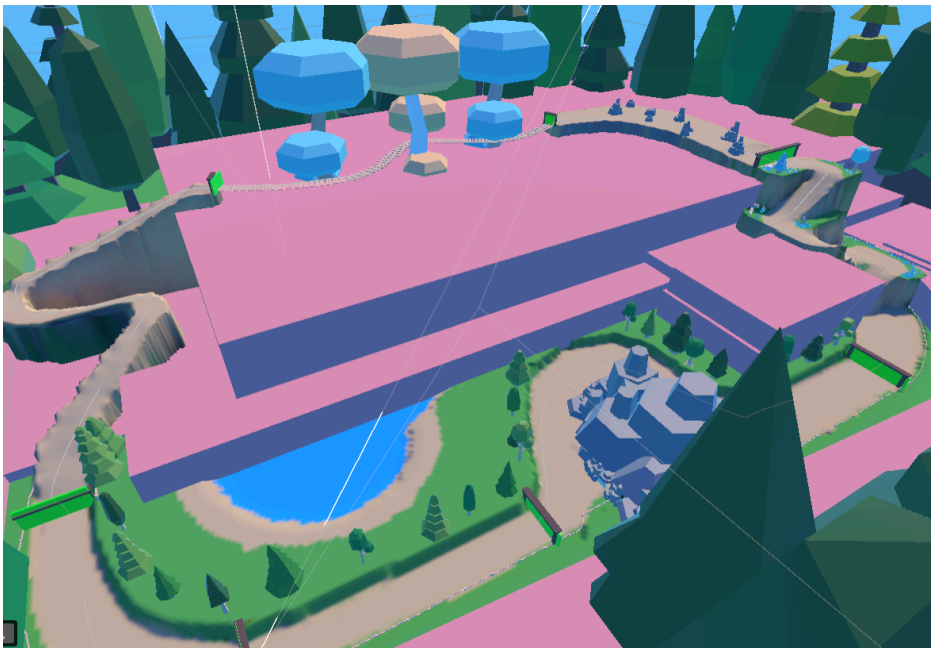

Figure 28 - Deadzones to avoid exiting race track in Species Sprint

4.3.8 #8: Environmental Design

Environmental design was included to make the world feel more alive. These were simple things like adding trees, rocks and water throughout the map. This makes the map feel more like a real place as opposed to just a level in a game. Examples of this can be seen in FIgure 21, Figure 24 and Figure 28.

Chapter 5: Implementation

5.1 Development Process

5.1.1 Sprint 1 (19/01)

- Project Proposal✅
- Feasibility Research✅

5.1.2 Sprint 2 (26/01)

- Setup Github✅
- Initial Unity Project✅
- Basic Terrain Layout✅
- Initial Character Controller✅

5.1.3 Sprint 3 (02/02)

- Basic UI(Menus, Maps)✅
- Travel Routes✅

5.1.4 Sprint 4 (09/02)

- Checkpoints✅
- Environmental Map Design✅
- Improved UI✅

5.1.5 Sprint 5 (16/02)

- Multiplayer❌
- Splitscreen❌
- 2nds Character Controller✅

5.1.6 Sprint 6 (23/02)

- 3rd Character Controller✅
- Leaderboard (Theory)✅
- Leaderboard (Working)❌

5.1.7 Sprint 7 (02/03 - 09/03)

- NPC Opponents✅
- Second Map❌

5.1.8 Sprint 8 (16/03 - 30/03)

- Multiple Characters✅
- Controller Support❌

5.1.9 Sprint 9 (06/04 - 27/04)

- Cosmetics❌
- Testing✅
- Post-testing alterations✅
- Finish Report✅

5.2 Major Challenges

There were multiple challenges faced throughout the course of this project, some which were possible to overcome and which weren't. Highlighted below are three such cases.

Timer

The timer is something that unfortunately was not possible to get functional to its fullest extent. It was possible to get it to the stage that the timer was able to count from the start of a race until the end and then display it on a leaderboard. Unfortunately, it wasn't possible to implement the starting timer that would play when at the beginning of each race. This is something that with more time would definitely be possible however I don't believe that it is a priority right now

Deadzones

The deadzones were quite difficult to get fully working. For a long time, they would work around half the time. The problem was they would very frequently glitch and ultimately not teleport you to the previous checkpoint. This was an issue which persisted throughout most of the development process until eventually it was possible to resolve the issue. In the end the issue was that because the player was still moving through the character script when they were trying to teleport, so the game was getting confused on which movement script should take priority. This was solved by disabling movement as soon as the player interacts with a deadzone and then attempting to teleport after.

NPC Animations

NPC animations are another thing which was not possible to get working. I still do not fully know exactly why, but the issue comes from the fact that the isMoving boolean in AIAnimationCheck was never switched to true. In hindsight, it may have been better to take a whole different approach to not only the NPC Animation but the whole NPC system in general. This is something that could be done with more time to develop.

Chapter 6: Testing & Evaluation

6.1 User Testing

User testing for this project went very well. Testers responded very well to the game. They found it very fun and all said that level navigation was very intuitive, which is exactly what I was hoping for. The main issues that seemed to come up though are that the controls could be a bit difficult sometimes and that some parts of the level were a bit sparse.

6.2 Error Handling

The main ways I dealt with the feedback that was given was to deal with the main issue that was presented in the feedback. I did this by:

- Adding rocks to the most empty section of the map to add some variety
- Alter the controls so that it feels more steady, the camera sits at a nicer angle and speed is quicker while turning feels more fluid. Overall the game feels a lot better

Chapter 7: Project Management

7.1 Methodology

The methodology for the project management was using the SCRUM method. I mentioned this previously but to recap, this methodology involves using Sprints in order to delegate work across multiple weeks, aiming to only move on to the next set of tasks when the current weeks are completed. Overall I think this process worked very well for the purposes of this project.

7.2 Tools

The main tools I used for project management in this thesis were:

- Github
- Trello
- Figma
- Miro

Miro and Trello were used to keep track of all of my assets and work progress. They were incredibly helpful in keeping me up to date with what I should have done at each point in the development cycle

Github helped me keep all of my code backed up and safe, which definitely saved me a few times as there were issues with some of my files after closing, meaning I couldn't just undo it. Thankfully, I had working versions of my file on Github so was able to resolve it without issue.

Finally Figma was used to make low and high fidelity prototypes for map design. This allowed me to get a very good idea of how to organise the level design and the UI.

Chapter 8: Conclusion

Overall I am very happy with how this project went. I learned a lot of new techniques throughout the course of the development process and I believe that this is the best game I have made to date. I received a lot of help from classmates and lecturers alike and this kind of collaboration was incredibly helpful.

There are a few things I wish I could have gotten done that I wasn't quite able to , such as multiplayer, leaderboards and animations for the NPC’s, but unfortunately I wasn't able to. In the future I would like to go back and continue development on this project.

References

Daineko, A. (2025, April 15). *Shape Language in Video Games: Character Design in 2025*. Ejaw.net. Retrieved April 29, 2026, from https://ejaw.net/shape-language-in-character-design/

Kaplan, R., & Kaplan, S. (2017, December 12). *The Psychological Impact of Light and Color*. TCPI. Retrieved April 29, 2026, from https://www.tcpi.com/psychological-impact-light-color/

Plowman, J. (2026, March 1). *Level Design Fundamentals | Epic Developer Community*. Epic Games Developers. Retrieved April 29, 2026, from https://dev.epicgames.com/community/learning/tutorials/3VKJ/unreal-engine-fortnite-level-design-fundamentals

Pranke, K., & Jensen, T. (2022, January 1). *Colours and Context*. Colours and Context. Retrieved April 29, 2026, from https://www.diva-portal.org/smash/get/diva2:1666624/FULLTEXT01.pdf

Shahbazi, N. (2025, August 28). *Shape Language: What It Is and Why It Matters!* Pixune. Retrieved April 29, 2026, from https://pixune.com/blog/shape-language-technique/

Yang, R. (2025, November 01). *The Level Design Book*. The Level Design Book. Retrieved April 29, 2026, from https://book.leveldesignbook.com/introduction