

# SPEEDWAY



35239979

## TABLE OF CONTENTS

<b>STATEMENT OF INTENT</b>	<b>1</b>	<b>MODELLING SOFTWARES</b>	<b>19</b>
MOTIVATION	1	Introduction	19
INTENDED SOFTWARE	1	Evaluation	20
TARGET AUDIENCE	2	<b>STORYBOARDS</b>	<b>20</b>
REQUIRED SKILLS	2	Introduction	20
LIMITATIONS	2	TRADITIONAL	20
TIME FRAME	3	THUMBNAIL	21
Evaluation	3	DIGITAL	21
<b>RESEARCH</b>	<b>4</b>	<b>RIGGING SOFTWARE</b>	<b>21</b>
FILE TYPES	5	Introduction	21
Introduction	5	Evaluation	22
Evaluation	7	<b>TUTORIALS</b>	<b>23</b>
<b>RENDERING</b>	<b>7</b>	Introduction	23
Introduction	7	Evaluation	23
BIASED	7	<b>PERSPECTIVES</b>	<b>24</b>
UNBIASED	8	Introduction	24
Evaluation	8	FIRST PERSON	24
<b>RENDERING SOFTWARE</b>	<b>9</b>	THIRD PERSON	24
Introduction	9	OMNISCIENT	25
SOFTWARES	9	Evaluation	25
Evaluation	10	<b>MUSIC</b>	<b>25</b>
<b>GPU vs CPU RENDERING</b>	<b>10</b>	Introduction	25
Introduction	10	DIEGETIC AND NON-DIEGETIC SOUND	25
GPU	11	<b>PROGRAMMING LANGUAGES</b>	<b>26</b>
CPU	11	Introduction	26
Evaluation	12	PROGRAMMING LANGUAGES PROS AND	
<b>FRAMERATE</b>	<b>12</b>	CONS	26
<b>PIXELS</b>	<b>13</b>	<b>CROSS-PLATFORM</b>	<b>27</b>
Introduction	13	<b>MULTIPLAYER</b>	<b>27</b>
DPI AND PPI	14	CLIENT-SIDE CONNECTION	27
<b>ASPECT RATIO AND DISPLAY RESOLUTION</b>	<b>14</b>	<b>SELECTION AND JUSTIFICATION</b>	<b>28</b>
<b>VIDEO GAME GENRES</b>	<b>15</b>	<b>FILE TYPES</b>	<b>28</b>
Introduction	15	<b>RENDERING</b>	<b>28</b>
Evaluation	17	<b>RENDERING SOFTWARE</b>	<b>29</b>
<b>GAME ENGINES</b>	<b>18</b>	<b>GPU vs CPU RENDERING</b>	<b>29</b>
introduction	18	Evaluation	29
		<b>ASPECT RATIO AND DISPLAY RESOLUTION</b>	<b>30</b>
		<b>VIDEO GAME GENRE</b>	<b>30</b>

<b>GAME ENGINE</b>	<b>31</b>	<b>FINANCE PLAN EVALUATION</b>	<b>45</b>
<b>MODELLING AND ANIMATION SOFTWARE</b>	<b>31</b>		
<b>PERSPECTIVE</b>	<b>32</b>	<b><u>RECORD OF PRODUCTION</u></b>	<b>46</b>
<b>MUSIC</b>	<b>32</b>		
<b>PROGRAMMING LANGUAGE</b>	<b>32</b>	<b>BLENDER</b>	<b>46</b>
<b><u>DEVELOPMENT OF IDEAS</u></b>	<b>33</b>	Evaluation	46
<b>CONCEPT SKETCH</b>	<b>40</b>	CAR MODELLING	46
Overall Evaluation	36	Front Of Car	46
Track Evaluations	36	Roof of Car	47
Car Evaluations	36	Back Of Car	47
Cut Scene Evaluation	36	Door of Car	48
Hierarchy Evaluation	36	Bottom of the car	49
<b>TESTING</b>	<b>37</b>	Wheels of Car	49
Car Function Script	37	Tires	50
Car Modelling First Attempt	37	Evaluation	50
Handling	38	ACCESSORY MODELLING	51
Timers	38	Checkpoint Modelling First Attempt	51
Best Time Display	38	Checkpoint Modelling Final Result	51
Collisions	38	Starting Lights	52
<b>WORKING DRAWINGS</b>	<b>39</b>	TRACK MODELLING	52
<b><u>DESIGN MODIFICATION</u></b>	<b>41</b>	Learning how to use the Modifiers	52
<b>STATEMENT OF INTENT</b>	<b>41</b>	Using a reference	52
<b>GAME IDEAS</b>	<b>41</b>	Creating the ROAD	53
<b>CAR MODEL</b>	<b>41</b>	Creating the barriers	53
<b><u>TIMEPLAN</u></b>	<b>42</b>	Editing The Track	54
TimePlan Evaluations	44	<b>UNITY</b>	<b>54</b>
Term 4 2021 Evaluation	44	Evaluation	54
Summer Holiday Evaluations	44	IMPORTING TRACK	54
Term 1 2022 Evaluation	44	Adding The Landscape	55
Term 1 Holiday Evaluation	44	Scaling The Map	55
Term 2 2022 Evaluation	44	Adding Trees And Skybox	55
Term 2 Holiday Evaluation	44	IMPORTING CAR MODEL	56
<b><u>FINANCE PLAN</u></b>	<b>45</b>	Evaluation	56
<b>THE FINANCE PLAN</b>	<b>45</b>	PROGRAMMING/USER INTERFACE	56
		UI Script	56
		Save Script	56
		RumbleSound Script	57
		Speed and Gear Display	57
		MiniMap Display	57
		Time, Position and Lap Displays	58
		Lap Banner	58
		Checkpoint Banner	58
		Buttons	59

Car User Controller	59	<b>FINAL EVALUATION</b>	<b>68</b>
Starting lights	60		
Time Trial Script	60	<b>STATEMENT OF INTENT</b>	<b>68</b>
Practice Script	61	<b>RESEARCH</b>	<b>69</b>
Race Script	61	<b>PLANNING</b>	<b>69</b>
Leaderboard	61	<b>CONSTRUCTION</b>	<b>69</b>
Car AI Control	62	<b>RACETRACKS</b>	<b>69</b>
Wrong Way	62	<b>CARS</b>	<b>70</b>
Multiple Cars	62	<b>IN-GAME USER INTERFACE</b>	<b>70</b>
<b>WORK, HEALTH &amp; SAFETY</b>	<b>63</b>	<b>TITLE SCREEN</b>	<b>70</b>
<b>ERGONOMICS</b>	<b>63</b>	<b>MAIN MENU</b>	<b>71</b>
<b>COVID-19</b>	<b>64</b>	<b>RACETRACK MENU</b>	<b>71</b>
<b>HAZARDS</b>	<b>65</b>	<b>STATS MENU</b>	<b>71</b>
<b>SIGNAGE</b>	<b>66</b>	<b>CAR SHOWROOM MENU</b>	<b>72</b>
<b>FLOODING</b>	<b>67</b>	<b>HANDLING</b>	<b>72</b>
Evaluation	67	<b>MENU NAVIGATION</b>	<b>72</b>
		<b>OVERALL</b>	<b>72</b>



## STATEMENT OF INTENT

For my Year 12 major work, I intend to code and create a video game using a game engine such as Unity, Unreal Engine or Gadot. This game will use a 3D model, which I will model, texture, and animate and will be driving around in a 3D environment which I will also design. This 3D game will also include features such as music, multiple levels, title screen, save and load functions, high-scoring system.

## MOTIVATION

I have created games previous to the one I will be working on and found it extremely challenging and enjoyable. My passion for creating games has developed over time from my time spent playing online games playing hundreds of hours developing ideas and opinions on what makes a great game. My experience coding is very limited only having made 3 games in the past.

Doing multimedia at Killara high has given me the opportunity to improve and get feedback on projects I am extremely passionate about which will allow me to achieve the highest possible mark. I have spent hours watching videos on coding languages such as c# and game engines like Unity developing my skills to prepare me for my year 12 multimedia major work and to eventually be able to pursue a career in the multimedia industry, or to complete a course at university or other forms of work in the multimedia industry.

## INTENDED SOFTWARE

- ❖ 3D Modelling Software – I plan to create an environment, player model and objects for my game using a 3D modelling software such as blender or maya.
- ❖ Game Engine – I aim to use a game engine such as Unity, Gadot or Unreal Engine to help design my game.
- ❖ Animation Software – I intend to use a program with animation functions such as armatures and bones which I have previously used in blender.
- ❖ Graphics Software – I plan to use graphic design software such as adobe illustrate and photoshop to design and create a UI as well as other features of my game.
- ❖ Programming Software – I will be using Visual Studios to write all my code for things such as movements and interactions within the game.
- ❖ Sound Editing Software – I aim to use audio-based software's such as audacity to create the noises that will be found within my game.

## TARGET AUDIENCE

My target audience will be teenagers to young adults who enjoy games, due to the fact that I intend my game to be based on functions that may be difficult for younger and less experienced players to understand and though I aim to add levels scaling in difficulty so anyone who would like to pick up the game could learn.

## REQUIRED SKILLS

- Ability to code
- Ability to animate
- Ability to 3D model
- Understanding of the game engine
- Ability to design textures
- Ability to design sound effects

## LIMITATIONS

- ❖ Computer – There are certain limitations that will come from my computer and school computers such as performance and portability to remedy this I will be using a USB to transport my work in between school and home but I will still be faced with the problem with the performance of the school computers so most likely at school I will priorities working on my folio for my project allowing me to commit the majority of my work at home to designing and editing the game I am making for my major work.
- ❖ Coding – When it comes to coding, I am limited in my ability so I will need to be very active in the coding community when it comes to finding tutorials and potentially post faulty code online for advice.
- ❖ 3D Modelling – My 3D modelling is probably the most advanced skill out of what will be required for this project and even that has quite a lot of area to improve but with sufficient time I am sure I will be able to create and design everything I will need.
- ❖ Animation – I have only animated characters a few times prior to this project, and they have all be very beginner and simple animations so I will need to dedicate quite a bit of time to watching tutorials and perfecting my animations to look smooth and make sure there isn't any clipping with the animation.
- ❖ Cost – Resources such as programs and assets may be difficult to obtain due to the price of them.
- ❖ Time – Despite having a year to work on my game not 100% of that time can be dedicated to working on it, this is because I have other commitments such as school family, friends, and other subjects so I will need to make sure that I provide myself with enough time that I can submit it on time.

## TIME FRAME

Managing my time will be extremely important especially considering the core focus of the project my game will only be able to be edited when I am at home due to it needing my computers performance as well as the fact the school computers don't always have Unity installed on them. Another thing I will need to take into consideration is that I will only have 3 lessons a week where I will be able to seek out teacher advice and feedback real-time. I will need to take into consideration all the features of the project and work on them according to a planned schedule which I will create using a Gantt chart, when forming the Gantt chart, I will need to also keep in mind things which might slow down my progress such as rendering.



## EVALUATION

Now that I have investigated the different aspects of game development, I now know what areas I will need to be researching more in depth. And that is what will make up the research section of my portfolio.

# RESEARCH



## FILE TYPES

### INTRODUCTION

I am researching file types in order to determine which file type will be most effective and suit my needs for my project. File types vary in quality and generally files which are better in quality will take up more space on a hard drive, but file sizes shouldn't be much of a factor I will need to consider because I have plenty of space on my hard drive and usb.

File Type	File Name	File Extension	Description
Image	Joint Photographic Expert Group	.JPG, .JPEG	JPEG or JPG is a commonly used method of lossy compression for digital images. The degree of compression can be adjusted, allowing a selectable trade-off between storage size and image quality.
	Graphics Interchange Format	.GIF	A GIF is a bitmap image format that is designed to play silent short videos in an infinite loop.
	Bitmap Image File	.BMP	Bitmap images are widely used on digital cameras, smartphones and online. They are determined by the number of pixels and colour depth.
	Tagged Image File Format	.TIFF	A popular file format used in the graphic design community to store raster images.
	Portable Network Graphic	.PNG	Raster image graphic system designed for transferring images across the internet.
	Truevision TGA	.TGA	A raster image file used across the animation and film industry intended for large screens.

Audio	Audio Interchange File Format	.AIFF, .AIF, .IEF	Common format for transferring audio samples but does not support compression.
	MPEG-1/ MPEG-2 Audio Layer III	.MP3	An audio format commonly used due to its form of lossy compression.
	Waveform Audio File Format	.WAV	A Microsoft format usually used on windows and is most of the time uncompressed.
	Vorbis	.OGG, .OGA	A free open-source project using lossy audio compression.
3D	Filmbox	.FBX	a 3D model saved in the Filmbox format that can be imported and exported across many different programs.
	OBJ	.OBJ	Standard 3D image format that can be exported between programs that can store object code and related data.
	Blender Format	.BLEND	A file format which is used by the modelling, rendering and animation software Blender capable of storing scenes, objects, materials, textures, sounds, images and post production effects within the file.
	3DS	.3DS	This file format is used by Autodesk an animation, modelling and rendering software to store projects.
	Polygon File Format/ Stanford Triangle Format	.PLY	A file format that is used to store three-dimensional data from 3D scanners.

## EVALUATION

After researching and learning about different file types for images I will now be able to import and export files efficient and effectively. I still need to pick my program of choice but from my research I am leaning towards blender because of its great compatibility with unity which is one of my potential game engine choices.



## RENDERING

### INTRODUCTION

Rendering is the process of converting 3D models into 2D images on computer. There are many different elements that you can use to change the result of the render which I will need to take into consideration when I am thinking about the graphics for my game. Rendering will play an integral part of developing my game because I will need it to design any cut scenes or animations I add to my project.

## BIASED

Biased rendering engines prioritize speed over accuracy and higher quality, which is why biased rendering engines are faster at producing results when up against unbiased rendering engines. Biased rendering engines also tend to leave you with less settings that over complicate your project if you don't know exactly what you are doing. Biased rendering engines are generally used in things such as vfx in live action scenes because unbiased engines won't have enough to work with to perform the necessary calculations. A good example of a biased rendering engine is blender internal.



VRay = Unbiased

Octane = Biased

## UNBIASED

Unbiased rendering unlike biased rendering focuses on realistic detail and high-quality projects. Unbiased rendering engines give you more options for more specific settings that will allow you to adjust the image to come out exactly how you want, all though you need to be well educated on what all the settings do otherwise it can lead to an image coming out that isn't visually appealing. These settings can scare people away from using Unbiased rendering engines. Unbiased rendering engines also require more powerful computers for the rendering to be finished in any suitable amount of time.

## EVALUATION

For my project I will most likely use a biased rendering engine such as eevee in blender, not because of the stress it may put on my computer but because of the requirements of my project and the time it will save for working on more important features such as the code and models. The simpler settings is also a feature which will benefit me in the long run as I won't need to waste time making sure the light rays are bouncing perfectly to look as realistic and good quality as possible.

## RENDERING SOFTWARE

### INTRODUCTION

In this section of my folio, I will be weighing the pros and cons of different rendering software's/engines. Rendering software's are going to be important in the process of designing my project, I will need to render things such as animations and cut scenes to make my game the best game it can be.

### SOFTWARES

Software	Pros	Cons
	<ul style="list-style-type: none"><li>- Both Biased and Unbiased Rendering Engines</li><li>- Free</li><li>- Great for modelling</li><li>- Easy to setup</li><li>- Previous Experience</li></ul>	<ul style="list-style-type: none"><li>- Requires a good computer</li><li>- Seemingly Cluttered Menus</li><li>- Learning Curve</li></ul>
	<ul style="list-style-type: none"><li>- Lots of user-created scripts and plugins</li><li>- More specific for modelling</li></ul>	<ul style="list-style-type: none"><li>- Animation software in 3D's isn't very good</li><li>- \$1,470 per year</li><li>- Compatibility issues</li><li>- Lots to learn</li><li>- Bad UV map tool</li></ul>
	<ul style="list-style-type: none"><li>- Great interface</li><li>- Feedback when modelling</li><li>- Simple camera positioning</li><li>- Visual material/shader construction</li><li>- Easily customisable</li></ul>	<ul style="list-style-type: none"><li>- “Non-Realistic” Rendering results</li><li>- aesthetic rather than quantifiable and scientific</li><li>- A\$1968 annually</li></ul>

 <b>octane</b> render	<ul style="list-style-type: none"> <li>- GPU rendering</li> <li>- Good options for tweaking and refinement of models</li> <li>- 3D instances</li> <li>- Good variety of plugins</li> </ul>	<ul style="list-style-type: none"> <li>- \$1,074 per year</li> <li>- Poor user interface</li> <li>- Learning Curve</li> <li>- Node-based</li> <li>- Can't import FBX files</li> <li>- Not always stable</li> </ul>
	<ul style="list-style-type: none"> <li>- Extremely flexible</li> <li>- Fast render times</li> </ul>	<ul style="list-style-type: none"> <li>- Complex settings</li> <li>- Spots and Flicker in rendering results</li> </ul>

## EVALUATION

Based on my research blender will be the ideal program for my project, this is due to factors such as previous experience, price, easy to set up and gives me access to both styles of rendering engines both biased and unbiased.

## GPU VS CPU RENDERING

### INTRODUCTION

When rendering there are many different ways to get your project rendered using different programs, outsourcing, biased or unbiased, but no matter what the process will required to be run by either a GPU or CPU. “The difference between CPU and GPU microprocessors is the way each one handles different tasks. A CPU performs different calculations to process tasks while a GPU has the ability to focus all computing abilities on a specific task”.

## GPU

A GPU uses thousands of tiny extremely efficient cores that can handle multiple tasks simultaneously which allows for extremely efficient rendering. GPUs were used as a response to graphically intense applications that would burn out CPUs and slowed down computer performance.

Pros	Cons
<ul style="list-style-type: none"><li>- You can use SLI setups for rendering</li><li>- GPU rendering uses less power</li><li>- Significantly Faster</li><li>- Lower hardware costs</li></ul>	<ul style="list-style-type: none"><li>- GPUs are reliant on CPUs</li><li>- GPUs depend on driver updates</li></ul>

## CPU

CPU rendering is a style of rendering that renders images only using the CPU. This method is popular due to it being cheaper in the short term as GPUs tend to be extremely expensive although fluctuating. CPU rendering is a good choice for users who would like to render a single image but users will run into problems if trying to render complex scenes or animations.

Pros	Cons
<ul style="list-style-type: none"><li>- The CPU isn't reliant on a GPU</li><li>- CPU programs tend to be more stable</li><li>- Not constrained by amount of VRAM like GPU is</li></ul>	<ul style="list-style-type: none"><li>- CPUs do not stack well</li><li>- Slower</li><li>- CPUs designs constantly changing leading to more required upgrades when trying to increase performance e.g., motherboards.</li></ul>

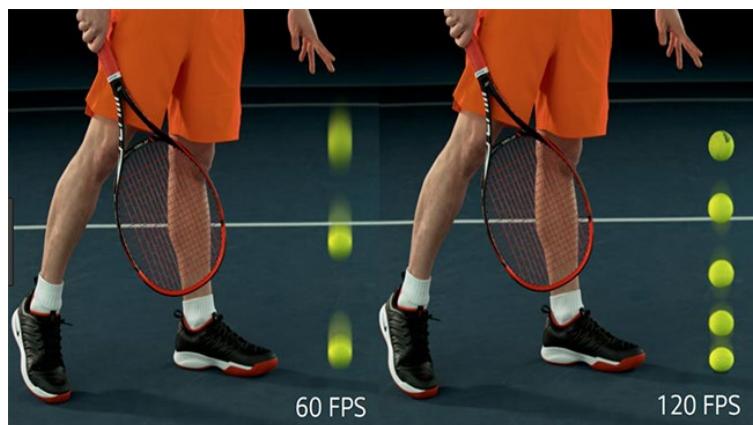


## EVALUATION

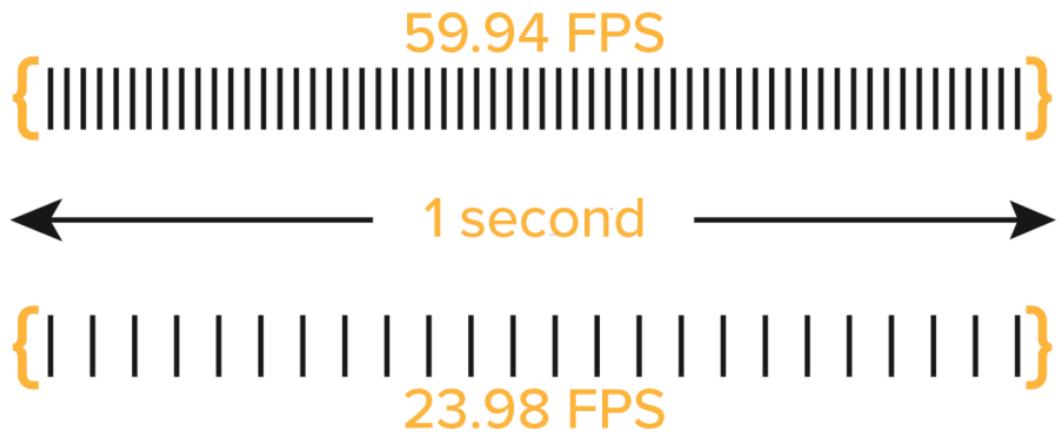
After doing research and taking everything into consideration I decided I will use GPU rendering and do my rendering at home on my pc which is stocked with a ryzen 5 3600x and a RTX 2070 Super. These computer parts are high end and should not have any problems rendering the required cut scenes and animations for this project.

## FRAMERATE

While watching videos everything is extremely smooth and put together, this is achieved by compiling many images with slight differences over each other in under a second this is referred to by the term Framerate or more commonly in the gaming world, FPS. Generally, when it comes to videos the more frames played a second the smoother the video comes out. For example, at 60 fps footage will look clean and smooth with very limited stutter-y actions and beneath 60 fps is 30 fps which is used in some cases to make videos load quicker due to them sending less data but the trade-off for the faster load times is worse footage which can appear jagged and jolty. Above both of those and generally the highest framerate used for videos is 120 fps which creates an extremely smooth video that is enjoyable to watch but similarly to 30 fps it has its trade-offs, 120 fps while being a lot better quality may take longer to load and will have bigger file sizes. Framerate has also been used to create videos in slow motion by getting a camera that take thousands of images per second and then playing them back to you at a rate such as 60fps or even slower.



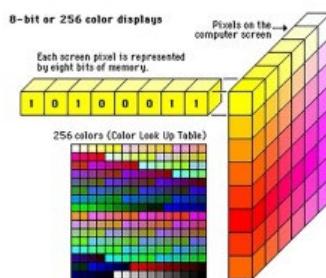
FPS	Quality	Size
120	Great	Large
60	Good	Medium
30	Poor	Small



## PIXELS

### INTRODUCTION

A pixel is the smallest element in a raster image and comes from the term picture element. A pixel is the smallest item of information in an image generally made up of three colours red, green and blue however sometimes can be made up by cyan, yellow, magenta and black that change strength to make up different colours.



Bits per pixel	Colours
1	2 (Monochrome)
2	3
3	8
4	16
8	256
16	65,536 (High Colour)
24	16,777,216 (True Colour)

## DPI AND PPI

Pixels are also used by computer mouse manufacturers to create a scale of sensitivity; this is done by measuring how many pixels the mouse moves over by inch this is referred to as dpi or ppi which can be used interchangeably. On top of this pixels are also used when it comes to printing using the same dpi/ppi system to measure out the quality of the printer. So, the higher the dpi/ppi the better the quality.



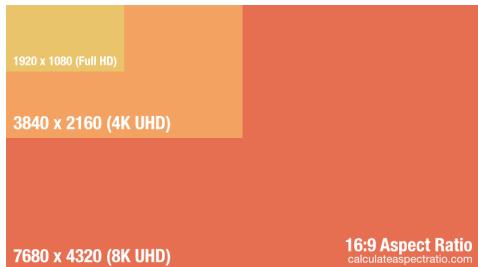
## ASPECT RATIO AND DISPLAY RESOLUTION

Display resolution is the visual dimensions of any display in terms of width and height measured using pixels. Resolution is a big factor along with fps that affects the quality of a video, the higher resolution of a photo the greater the file size will be but the greater quality it will be.

Aspect ratio is the result of measuring the number of pixels on the width and height of the screen which gives you the resolution and then dividing it to get the lowest common multiple so for example  $1920 \times 1080 = 16:9$

Common Example of Aspect Ratios and Resolutions

Resolutions	Aspect Ratios
800 x 600	4:3
1024 x 768	4:3
1152 x 648	16:9
1680 x 944	16:9
1920 x 1080	16:9
2048 x 1152	16:9
3840 x 2160	16:9



Example of how resolutions scale



Comparison between different resolutions

Different aspect ratios are used:

Aspect Ratio	Use
9:16	iPhone
4:3	Standard Definition TV
16:9	HD and widescreen TV, majority of computer displays
9:19	'Vertical Video' (Recording on smartphones)
8:3	Cinematography, Most movies, Widescreen computer screens
18:5	SuperUltra wide TV

## VIDEO GAME GENRES

### INTRODUCTION

Video games are computer programs designed to challenge the player through interaction with a user interface or input device such as controller, mouse, or keyboard. Because of how general this term is it is broken up into multiple genres to help people out when purchasing and choosing which game they want to play.

Genre	Description	Example
Sandbox	A sandbox is an environment which enables the player to do anything they would like to do with the game generally breaking physics and doing this which wouldn't be possible or legal in our real world.	 Garry's Mod
First Person Shooter (FPS)	First person Shooters commonly referred to as FPS are games that try to immerse the player into a world of violence and danger designed to test their aim and decision making by firing weapons at either computer-generated players also known as bots, or fire at other players to see who comes out on top with most kills and least deaths. This genre can also be split across many different game modes such as team death match, battle royale, death match. These game modes adjust the aim of the game to challenge players in different ways some introducing rng to balance the game between newer less skilled players and older better players by adding a luck-based component.	 Counter Strike Global Offence
Role-playing (RPG)	A role-playing game commonly referred to as RPG is game where the player takes control of a character set in a fictional world and take responsibility for acting out these roles within the narrative.	 Skyrim

Multiplayer	Multiplayer games are games that use online servers to connect its players so they can play together, this is commonly seen in game genres such as Shooters, Role-playing, Adventure and Survival.	 Grand Theft Auto Online
Action	An action game is designed to test the hand eye coordination, and reaction-time of players by throwing unexpected scenarios at them.	 Battlefield 2042
Adventure	An adventure game is a game that puts the player in the role of protagonist in an interactive story that challenges using puzzles and exploration.	 Star Wars Jedi: Fallen Order

## EVALUATION

After researching different game genres, I now have a better understanding of games and their different genres. This will help me in my selection and justification when deciding what game, I will be making for my project.



## GAME ENGINES

### INTRODUCTION

A game engine is a software designed to help programmers when creating their own games generally offers help of general libraries and support programs. They make it easier for programmers by offering a set of software tools or API's built to optimize game development.

Software	Pros	Cons
	<ul style="list-style-type: none"><li>- No royalty</li><li>- Good level of control</li><li>- Multiple platforms</li><li>- Great Community Support and Documentation</li><li>- Good asset Management</li><li>- Asset store</li><li>- Ability to make own tools</li><li>- Consistent rendering engine</li><li>- Great for 2D and 3D games</li><li>- - Great for Beginners</li></ul>	<ul style="list-style-type: none"><li>- Weak lighting engine</li><li>- Compatibility issues between updates</li><li>- Poor animation tool</li><li>- - Basic material editor</li></ul>
	<ul style="list-style-type: none"><li>- Open source</li><li>- Multi-platform</li><li>- - only 20MB</li></ul>	<ul style="list-style-type: none"><li>- Primarily 2D game engine</li><li>- Community is small</li><li>- - 3D features are incomplete</li></ul>
	<ul style="list-style-type: none"><li>- Great for PC games</li><li>- Better powered for 3D games</li><li>- Next gen graphics</li><li>- Great rendering technology</li><li>- C++</li></ul>	<ul style="list-style-type: none"><li>- Designer friendly only</li><li>- Suited to specifically FPS style games</li><li>- - 5% tax on games you make</li></ul>

## MODELLING SOFTWARES

### INTRODUCTION

Modelling software's are computer programs designed to allow the user to build simulations and other models. I will need to use a modelling software to design models and animations for my game and will be an essential part of my game's development process, this is why I will need to choose the best software for my project.

Software	Pros	Cons
	<ul style="list-style-type: none"> <li>- Free</li> <li>- Great for Animation</li> <li>- Keyboard shortcuts</li> <li>- Blender addons</li> <li>- Great modifiers</li> <li>- Great community</li> <li>- Previous experience with blender</li> </ul>	<ul style="list-style-type: none"> <li>- Learning curve</li> <li>- Seemingly cluttered menus</li> <li>- Selection system</li> <li>- Extrusion tool</li> </ul>
 AUTODESK 3DS MAX	<ul style="list-style-type: none"> <li>- Cornucopia of user-created scripts and plugins</li> <li>- 3DS Max is geared more towards modelling whereas others like Maya are more towards animation.</li> </ul>	<ul style="list-style-type: none"> <li>- There is a lot to learn</li> <li>- Bad normal align tool and UV mapping</li> <li>- Compatibility issues</li> <li>- Not cross-platform</li> <li>- \$1,470 per year</li> </ul>
	<ul style="list-style-type: none"> <li>- Intuitive interface</li> <li>- Visual feedback when modelling</li> <li>- Easy camera positioning</li> <li>- Visual material/shader construction</li> <li>- Easily customisable using Maya Embedded Language</li> </ul>	<ul style="list-style-type: none"> <li>- “Non-Realistic” Rendering</li> <li>- results aesthetic rather than quantifiable and scientific</li> <li>- A\$1968 annually</li> </ul>

## EVALUATION

Thanks to this research I will be more confident when it comes to choosing what program I should base my project on. This project will be reliant on my ability to model and create the necessary animations for my game to achieve the highest possible marks and provide the best experience possible.

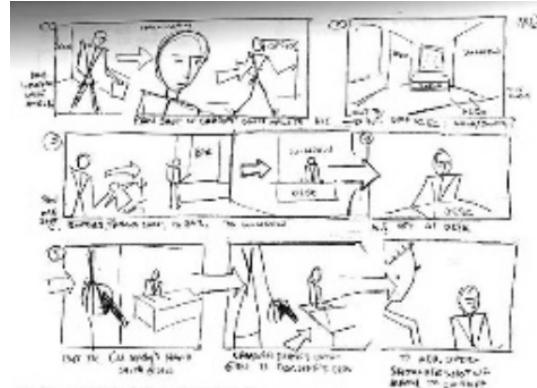
## STORYBOARDS

### INTRODUCTION

Storyboards are sequences of drawings that lay out and represent the key points of the story or process. I need to research storyboards so I can decide on what storyboard I will use to conceptualize my game and lay out the bones of the game so that the process of designing each scene is easier because I will already know what order I should be doing it in.

### TRADITIONAL

Traditional storyboarding is a series of general sketches that help gives the writer, producer, and director a clearer understanding of how they want their work by allowing them to visualize it. These storyboards are still very common in the TV and movie industry as they allow for changes to be made quickly and without any cost.



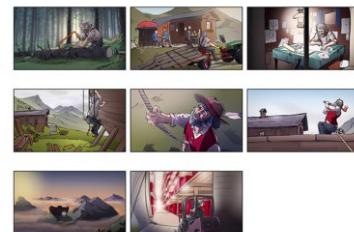
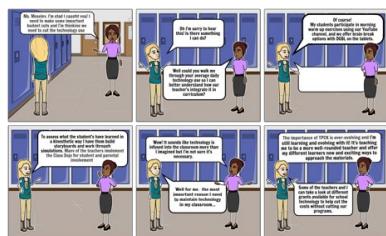
## THUMBNAIL

Thumbnail storyboarding is similar to traditional storyboarding but instead usually has no text for context and is a lot quicker to put together generally used when the creators of the project already have a better idea of how each scene will play out.



## DIGITAL

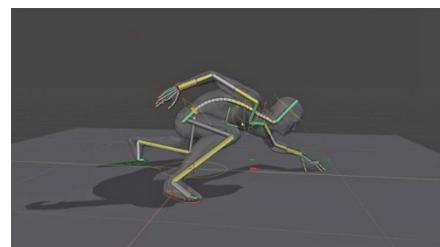
Digital storyboarding is perfect for animations because you are able to place the exact graphic that will be used in the final production in each sequence. This allows everyone to get a peek at what the final video will look like. However, doing this means the process of making these storyboards will take longer and be more expensive to correct any errors.



## RIGGING SOFTWARE

### INTRODUCTION

Rigging is the process of creating a skeleton within the 3D model that has an effect on the surrounding mesh allowing you to easily adjust the bones which in turn makes the mesh around it. Any animation my project may include will require me to be able to set up rigs to my models to do this I will need to be able to use the best possible software for this part of the process that is creating my project.



Software	Pros	Cons
	<ul style="list-style-type: none"> <li>- Wide assortment of tools for designing scenes</li> <li>- Free to use</li> <li>- Multiple Platforms (Windows, Mac, Linux)</li> <li>- Large community who are extremely active</li> <li>- Many Add-ons available</li> <li>- Loads of tutorials</li> <li>- Prior Experience</li> </ul>	<ul style="list-style-type: none"> <li>- Takes time to learn well</li> <li>- Known to Crash</li> <li>- Complex UI can make it difficult for new users</li> </ul>
 AUTODESK 3DS MAX	<ul style="list-style-type: none"> <li>- Used by professionals</li> <li>- Very flexible</li> <li>- Somewhat easy to use</li> <li>- Great tools for rigging</li> <li>- UI is great and easy for beginners to pick up</li> </ul>	<ul style="list-style-type: none"> <li>- Costs \$1,700 per year</li> <li>- Not Cross-Platform</li> <li>- Doesn't support programs outside Autodesk library</li> <li>- I have never used it before</li> </ul>
	<ul style="list-style-type: none"> <li>- Commonly used in the multimedia industry</li> <li>- Fantastic rigging</li> <li>- Cross platform (Windows, Mac and Linux)</li> <li>- Lots of tutorials</li> </ul>	<ul style="list-style-type: none"> <li>- Challenging to learn</li> <li>- Expensive (\$1,986 per year)</li> <li>- I have no previous experience with maya</li> </ul>

## EVALUATION

This research has given me insight into an area of the project I will need to take into consideration and will largely be determined whether I can get a school provided copy of the more expensive programs.

## TUTORIALS

### INTRODUCTION

Tutorials will play a large part in this process and will take up a considerable amount of time as I will still be learning a large amount of the skills that I will need to use for this project. Luckily for me thanks to YouTube I won't have to pay money on websites such as square space and I will be able to learn all the skills I need for free.

Tutorial	Channel	Videos that may help
Modelling	Blender Guru	Donut tutorial Beginners Guide to Learning 3D Computer Graphics
	Blue Inversion	Smart Shaders: Grounds & Roads   Blender 3D Car in Blender 2.8x - PART I - Modelling [ Beginners ]
	Imphenzia	Let's model A SCIFI BIKE in 10 MINUTES Learn Low Poly Modelling in Blender 2.9 / 2.8
	CG Geek	How to Create a Low Poly Character in Blender 2.8 1 Minute Blender Tutorials
Coding	Brackeys	HOW TO PROGRAM - Getting Started! MAKE YOUR GAME RUN SMOOTH - Unity LOD Tutorial
	Alex Lee	Learn Java in 14 Minutes (seriously) Java Basics 1
Animation	CG Geek	Become a PRO at Animation in 25 Minutes   Blender Tutorial Create Satisfying 3D Animations   Easy Blender Tutorial
	CBaileyFilm	Blender 2.91 Character Animation Tutorial for Beginners! Character Animation 101

### EVALUATION

These tutorials will be extremely useful when I am designing and coding my game. Thanks to this research I will be able to find good channels easily and quickly whenever I need help hopefully making my project process extremely efficient.

## PERSPECTIVES

### INTRODUCTION

Perspective is how you represent an object or situation within three-dimensional space so height, width, and depth. Games come in many forms a one factor which can change the way a game can feel or is played is the perspective of the game.

### FIRST PERSON

First person games are games where you view the world as the character would so through their eyes this type of perspective is commonly seen in shooter games and realistic simulation games. This is generally because of the sense of immersion that it provides the user as well as making certain games translate more with real life skills. This perspective is also used in a lot of games where the character isn't designed by the developers to give a sense of the protagonist being you.



Star Wars Battlefront II



Grand Theft Auto V

### THIRD PERSON

Third person games are used for games which are designed for less realistic mechanics having the camera in third person can help with making sure the user isn't as disorientated. This type of perspective can commonly be found in games which aren't as competitive and in games where the main character has their own story for you to play through commonly seen in games such as superhero games.



Batman Arkham Knight



Grand Theft Auto V

## OMNISCIENT

Omniscient games are generally found in strategy games where you are required to look at the full map so you can plan out and develop your group. Games such as Stellaris use this perspective to control an empire and battle others, this perspective gives you an idea of how the world is progressing around you which can help you plan out your next moves.



Stellaris



Clash of Clans

## EVALUATION

This research will make it easy for me to determine what camera settings I will need for my game depending on its style and genre. It is also worth noting that some games have multiple perspectives, games such as Grand Theft Auto V allow the player to switch between first and third person as they wish although first person was a feature released later in the game's success to attract buyers when the latest generation of console came out in 2013.

## MUSIC

### INTRODUCTION

Having the right music is an integral part of designing a game but for it to work there are many factors I will need to consider. It is important for players to have music within their games as to make sure they don't get bored by nothing by environment sounds.

### DIEGETIC AND NON-DIEGETIC SOUND

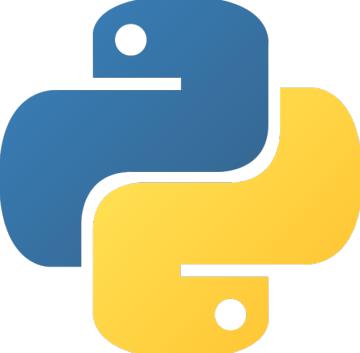
Diegetic sound is any sound that comes from within the story world. This type of sound can be heard by the characters in the story and can affect the way the story develops. For example, if a radio turns on and an enemy hears it leading to the capture of your character. Non-Diegetic sound however is a noise that comes from outside of the story world and is a sound the characters cannot hear. This can be used to manipulate the emotions of the viewer whilst keeping the game's development continuing along the same path. Depending on the project I try to create this will be a factor I can use to change the way the story unfolds as well as something I can use to control the emotions of the user.

## PROGRAMMING LANGUAGES

### INTRODUCTION

Programming languages are artificial languages that can be used to control the behaviour of a machine. There are many different languages that you can learn and use for this some more affective and some less most of them are specific to different areas. I need to research this so I can see which language will benefit me when creating my project.

### PROGRAMMING LANGUAGES PROS AND CONS

Programming Language	Pros	Cons
	<ul style="list-style-type: none"><li>- Easy to learn</li><li>- Development and maintenance are easier</li><li>- Very fast</li><li>- Good for game design</li><li>- Lots of tutorials</li><li>- Previous experience</li></ul>	<ul style="list-style-type: none"><li>- Based on Microsoft .Net framework making the language not flexible</li></ul>
	<ul style="list-style-type: none"><li>- Client-side</li><li>- Used everywhere on the web</li><li>- Simple to learn</li><li>- Great for designing user interface</li><li>- Little demand on servers</li></ul>	<ul style="list-style-type: none"><li>- Not flexible at all</li><li>- Mainly for browsers</li></ul>
	<ul style="list-style-type: none"><li>- Open source</li><li>- Easy to read</li><li>- Easy to learn</li><li>- Large libraries</li><li>- Powerful control capabilities</li></ul>	<ul style="list-style-type: none"><li>- Code is executed line by line making it slower than others</li><li>- Database access layer is underdeveloped</li><li>- Uses large amounts of memory</li></ul>

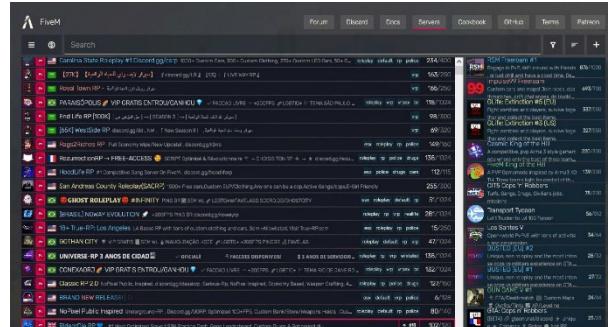
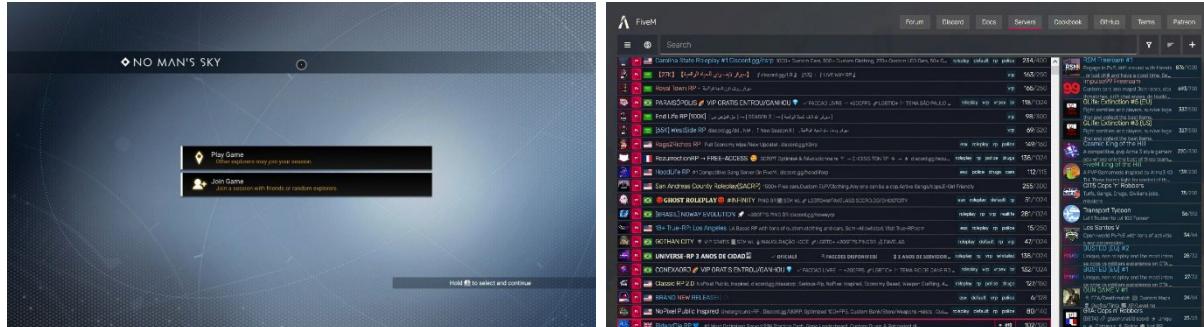
## CROSS-PLATFORM

Cross Platform is a feature within many games that can include online and just allowing the game to be on multiple platforms. Multi/Cross-Platform is used to reach a larger market when distributing your games by profiting off of multiple industries such as Windows, Linux, Macintosh and Mobile Devices.

## MULTIPLAYER

### CLIENT-SIDE CONNECTION

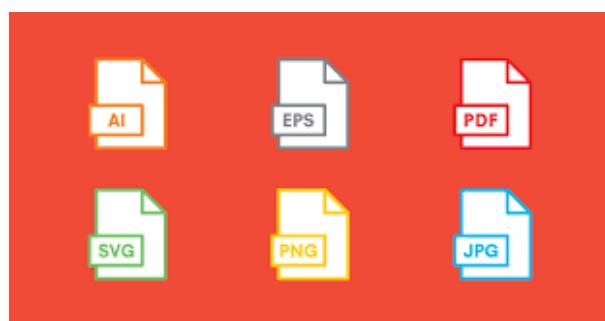
Client-side connection is a feature that exists within many games allowing players to play online together on the same server. Client-side connection also known as player to player or p2p allows the players playing the game to run servers off of their own computers reducing lag and costs for me as I don't need to invest in running a server. However, this feature has its own limitations, these limitations include a limited competitive possibility this is due to the fact that there are no public servers for anyone to join meaning that most likely you would only be able to compete against friends instead of strangers online who may be better at the game. This style of multiplayer is seen in big games such as No Mans Sky, Grand Theft Auto, Warframe and Elden Ring.



## SELECTION AND JUSTIFICATION

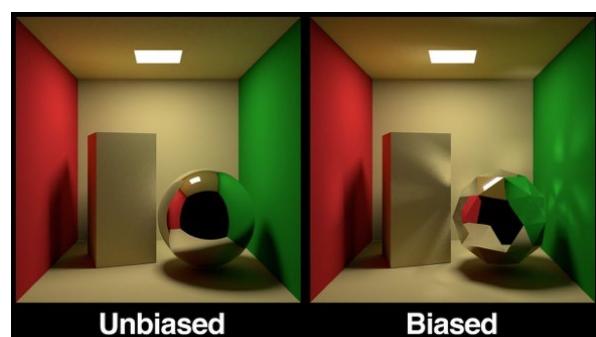
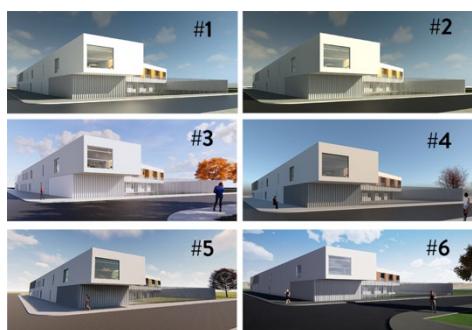
### FILE TYPES

Throughout my project I will use a variety of file types which will change depending on what I need the file for due to factors such as quality, size, programs all varying so I will need to choose the correct file type most suited to its requirements. Researching the different file types has given me a better understanding of them and I will now be able to decide between them a lot more efficiently and effectively.



### RENDERING

For my project I will be switching between biased and unbiased rendering so that the most important sections are the best quality, and the less important sections will be rendered in slightly worse quality to conserve time. Time will be the main factor in deciding which type of rendering I will use and for what because unbiased rendering takes a considerable amount of more time for only minimal improvement in quality.



## RENDERING SOFTWARE

The Rendering Software I have decided to use will be blender as it most suitably fits my needs as well as being free which will save me money. My previous experience with the program also had a large impact in my decision due to the fact that I know how to use it and I know it works perfectly for my needs.



## GPU VS CPU RENDERING

For my project I will be rendering everything with a GPU as they are much more efficient and effective. The only downside to using GPU rendering is the cost of it but this isn't not a problem that I have to deal with due to the fact that I already own a RTX 2070 Super which will make render time considerably faster than if I was using a CPU.

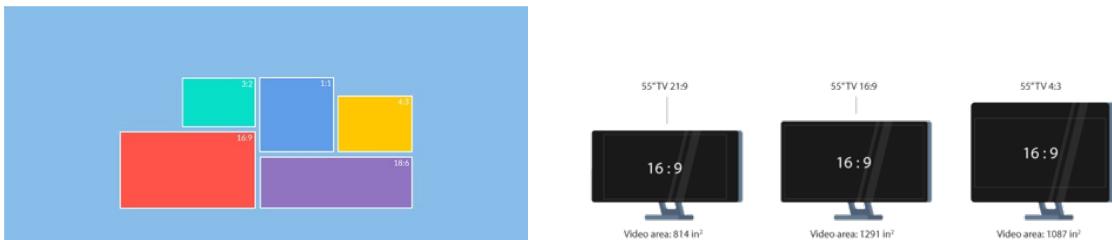


## EVALUATION

Deciding what to use for my major project has been relatively easy the main thing I have been unsure of so far was whether I should use unity or unreal engine. With the release of UE5 in April that has caught my interest, but I think the release is too late as it would only have me with 3 months to code and make sure the project works together so I will most likely end up using unity as I have the most experience with it.

## ASPECT RATIO AND DISPLAY RESOLUTION

I will be using 16:9 1920 x 1080 as it seems to be the most common for gaming on the computer which will be my target platform. 16:9 is also in general a very common aspect ratio which will make translating between display resolutions simple to do for the game. 1920 x 1080 is also relatively easy to run for most systems whereas high resolutions will be difficult for the computer to keep a consistent framerate on. The lowest resolution I would suggest playing the game on would be 1280 x 720 which is the lowest display resolution to the game which provides good quality while being easy to run on computers.



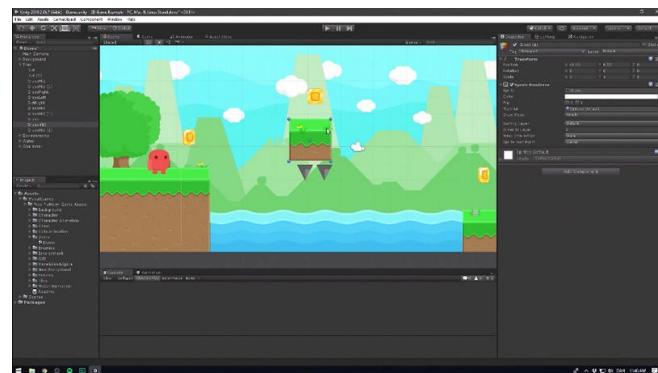
## VIDEO GAME GENRE

The game I will be making will be a racing game where you will drive a car through different checkpoints either around a racetrack or city. I will aim to design 3+ maps in my game for the player to race around on one being a race track another a city and the last one a floating island. I aim to add different obstacles and mechanics to make the driving challenging but also playable.



## GAME ENGINE

I will be using Unity to develop and design my game as it is free, and I have previous experience with it having made two previous projects using unity as my game engine. Another benefit of unity that has made me decide to pick it was its huge community supplying both tutorials and assets for me to use when creating my project. Using these assets and tutorials will allow me to focus on the functionality and development of my game as the coding is the core focus of this project.



## MODELLING AND ANIMATION SOFTWARE

I have chosen blender to be the software I will use to model and animate. This is because of my previous experience with the software meaning I won't need to waste time learning a new software. The fact it is free is also a key reason for my choice of modelling and animating software the quality blender offers for the price makes blender stand significantly above other software's especially for a high school project. Blender will be able to offer me all the necessary tools and functions.



## PERSPECTIVE

The game will be in a third perspective point of view this is because of the genre being racing meaning first person would not be enjoyable for the player. Also being able to see the vehicle you are driving is a big part of games like these. Due to this being in third person I will need to make the animations and models in much more detail as the user will be able to view the whole world much easier than if the game was in first person.



## MUSIC

I will be using diegetic sound in my game as I can utilize the fact that the car has a radio potentially allowing me to setup a feature that will allow the player to control the music that is playing. Hopefully this feature will add a bit more functionality for the user.

## PROGRAMMING LANGUAGE

I will be coding my game in C# as I have used it previously when coding games on unity meaning there will be less, I will need to learn how to do for my game to be its best possible version. C# is also a very fast language which means the game will run significantly smoother than if I were to use a different language. C# also has a great community who are always able to help on forums when stuck on any problems to do with my coding.



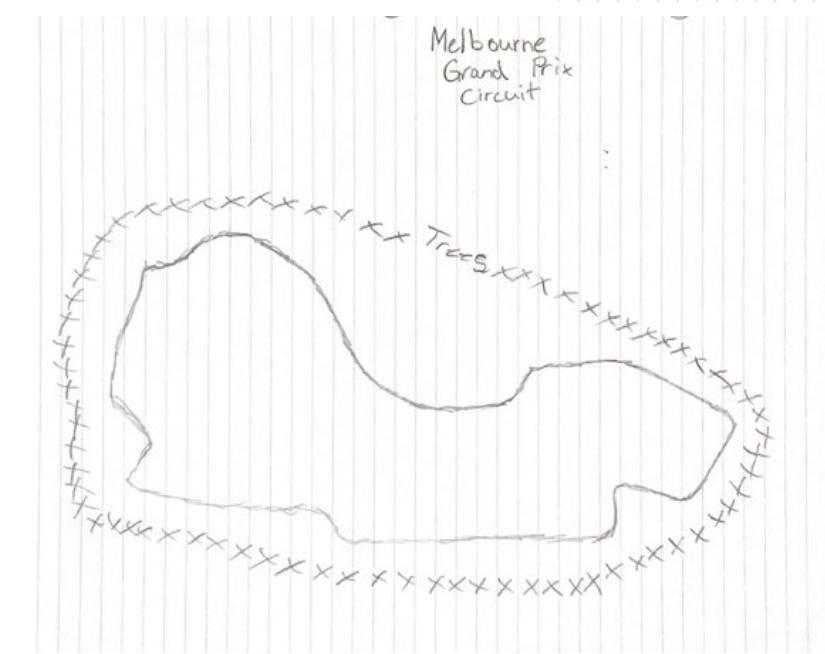
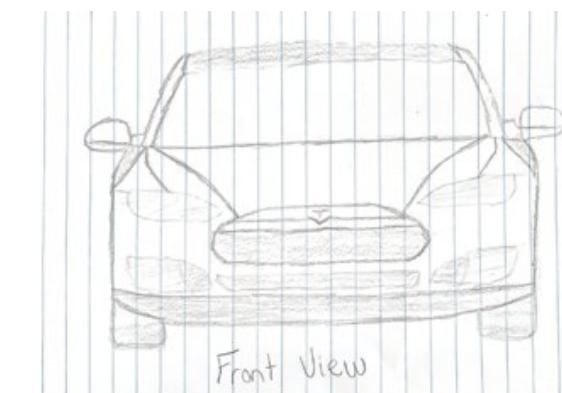
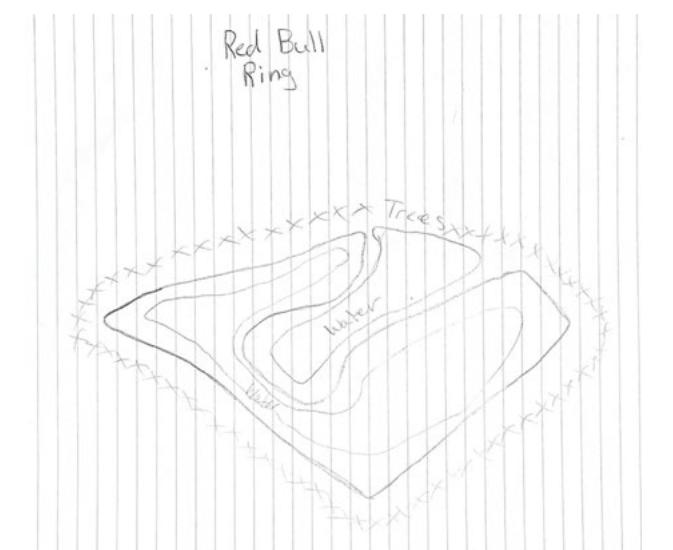
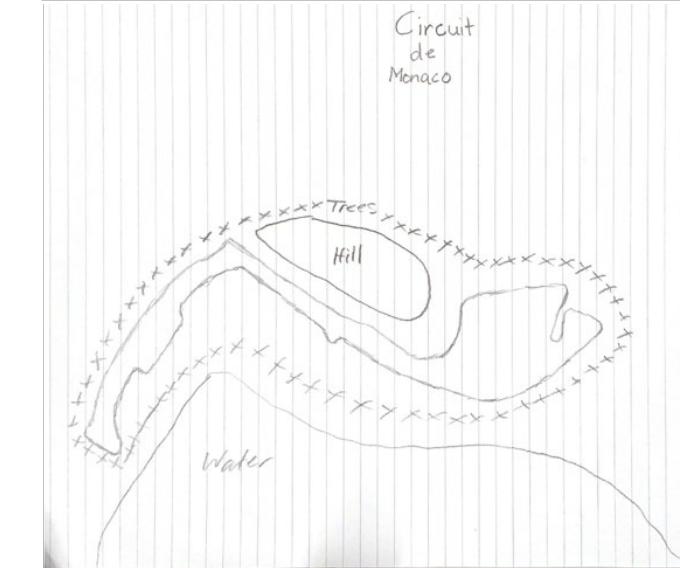
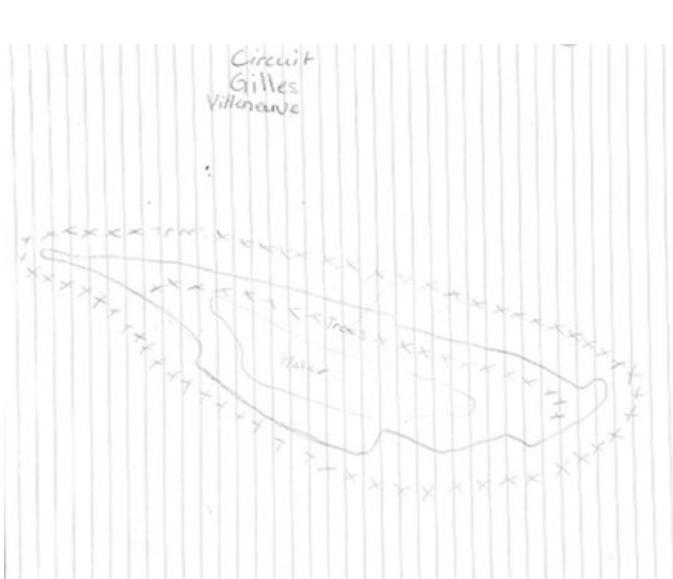
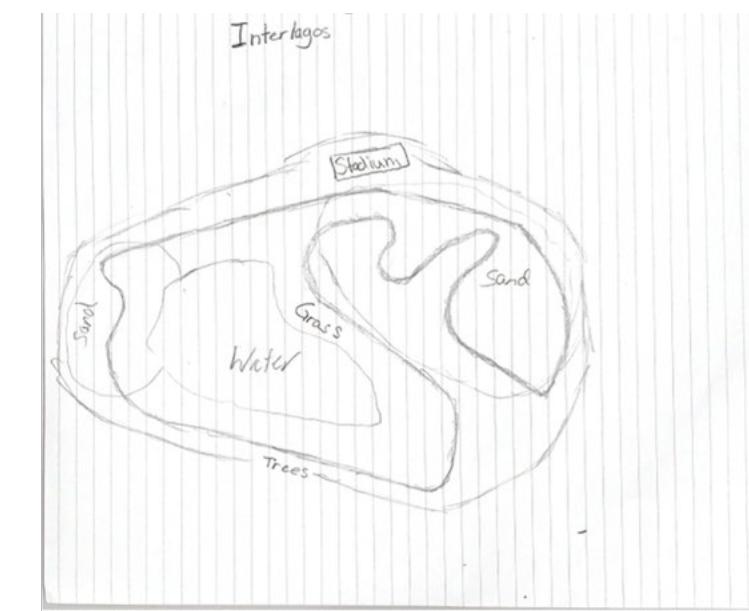
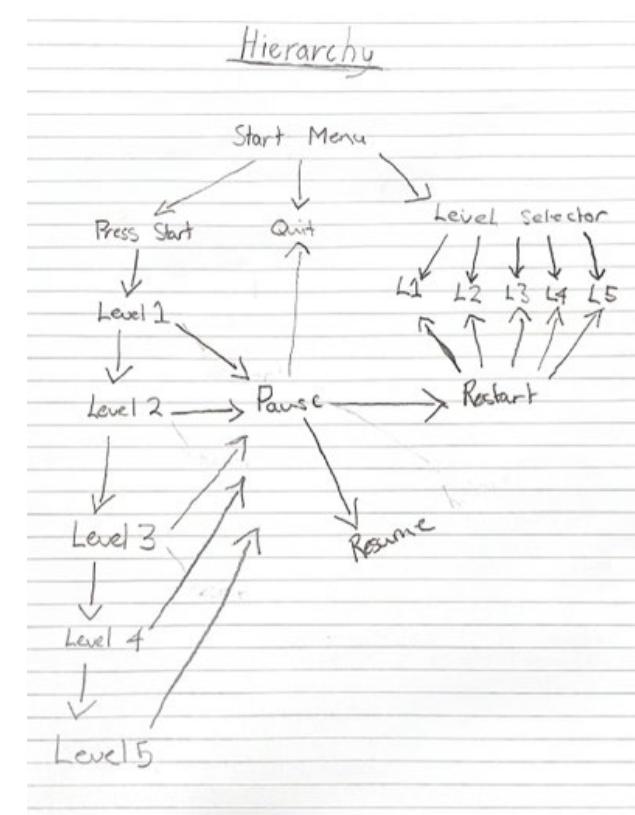
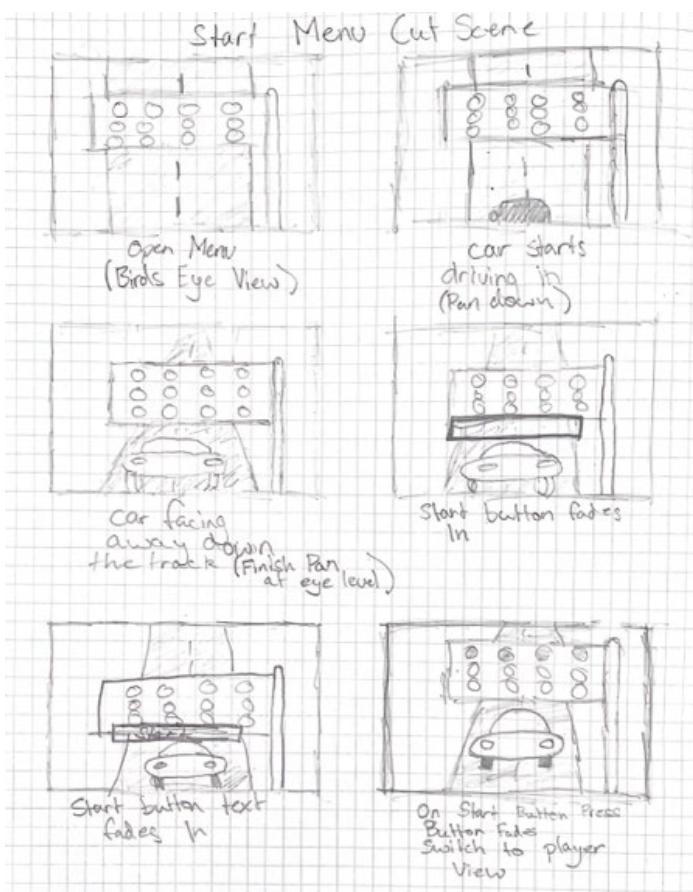
# DEVELOPMENT OF IDEAS



My original idea for this project was to create an animation story of a character following me around in a 3D world I designed while I walk around freezing time in difficult situations. But with further consideration I decided I wanted to do work on an area I'd like to pursue into university and work in, in the future, so I came up with the idea of a game where you were in a plane flying through obstacles but that seemed too inspired. My next idea was to try and make a game similar to 2008 Supersonic Acrobatic Rocket-Powered Battle-Cars but then I was more interested in a game like Forza Horizons focused more on a Grand Theft Auto time trial mode essentially creating a game similar to Trackmania.



## CONCEPT SKETCH



## OVERALL EVALUATION

---

These Sketches have given me a rough idea of what I want the final product of my game to look like and include I am going to rework and redraw them so they are more organised and have a more professional look so I am better able to understand them.

## TRACK EVALUATIONS

---

The design of these tracks are based on popular f1 tracks making the final design of these tracks significantly easier as I don't need to research ideal twists and turns and can just focus on the modelling and coding of the game and tracks.

## CAR EVALUATIONS

---

The many car models have helped me to decide on the specific ones I want in the game and what I will add to my final storyboard.

## CUT SCENE EVALUATION

---

My concept design for the cut scene was initially to difficult which is why I decide on something different for the final product.

## HEIRARCHY EVALUATION

---

This is the template for my hierarchy that the game will follow it should provide a good reference on how the player will navigate between menus.

## TESTING

### CAR FUNCTION SCRIPT

```
using System.Collections;
using.Collections.Generic;
using UnityEngine;

public class Testing : MonoBehaviour
{
    public bool carLocked = false;
    public bool carStarted = false;
    public bool engineStarted = false;

    void Start()
    {
        lockCar();
    }

    private void Update()
    {
        if (Input.GetKeyDown(KeyCode.P) && carLocked = false)
        {
            carStarted = true;
        }

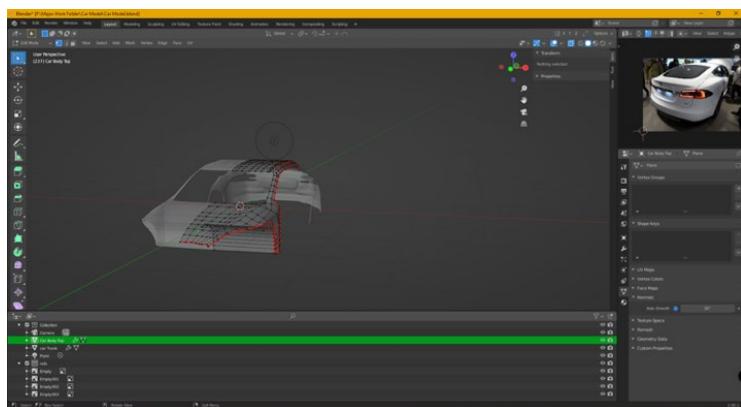
        if (carStarted = true && Input.GetKeyDown(KeyCode.L))
        {
            EngineOn();
        }
    }

    private void lockCar()
    {
        carLocked = true;
    }

    private void EngineOn()
    {
        engineStarted = !engineStarted;
    }
}
```

This script is the base and an idea of what will eventually become the code that will control the functions of the car in my game. The idea behind this code is to set up the necessary requirements for the car in my game to move. By determining states, the car is in I can add features such as locking car and control over the ignition. This script is just a rough draft and idea and will need to be worked on further before final submission of my multimedia major work. This code essentially works by creating multiple different classes which will run set lines of code which switch the class between true and false, this is achieved using a bool which creates a binary state of only two possible choices true or false. Within the Update class it will check every frame to see if I have called the other class within to run and switch it from true to false or vice versa.

### CAR MODELLING FIRST ATTEMPT



Modelling this car is a project I have undertaken that has proven to be quite challenging. I have never modelled anything in quite as much detail. The way I have decided to go about modelling this car is by using blueprints of a tesla car and using specific views to adjust vectors so they match up along the edges of where the image says they should be. I started this off with a plain rectangle which I fit to the images and made transparent so I could use it as a reference of dimensions when modelling. Blender crashed and I now have to start again... I forgot to save.

## HANDLING

The most important feature of a car racing game is obviously how the car works because of this I spent many hours slowly tweaking each individual variable of the cars controls until I was satisfied with the result of the handling. The next part of the handling I had to adjust was the change in handling when you are driving on the terrain and not the road which was a much quicker process to figure out due to my previous testing of the on-road handling.

## TIMERS

The timers for race time, lap time and best time was a difficult task to set up as it required a bit more challenging math and a lot of code which made it challenging to identify issues in the code but after some research and re-evaluation of how I was dividing and multiplying the milliseconds I realised the problem was in a separate part of code where I was setting the point which the hundredths of seconds was turning to tenths of a second which created issues with the timers display.

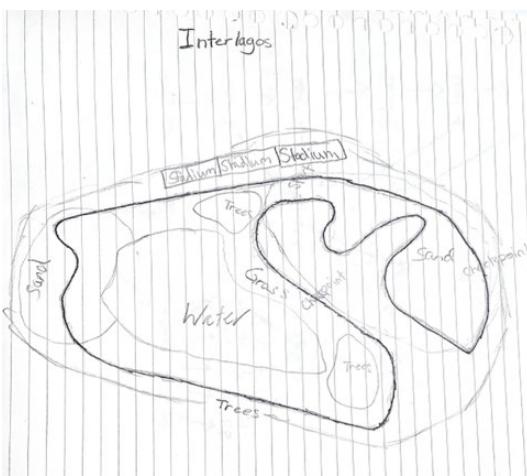
## BEST TIME DISPLAY

The displaying of the best time was a feature that I was extremely set on adding it. I had issues with the updating of the best time after each lap. The basis for the code was quite logical where all it needed to do was on the first lap set current lap time to best lap time then from the second lap onwards measure the difference between the previous best lap time and the current lap time and see if current lap time was lower in which case, it would set the previous lap time to string and display it. But I ran into some issues with the displaying of the best lap time from the second lap onwards and after a lot of testing, analysing and rewriting code I finally found that in one line of the code I was setting the last lap time to the previous best lap time before it was doing the calculations therefore not adjusting the best lap time display. This issue also created issues with the displaying of the new record banner that would appear across your screen when you had beaten your previous best lap time.

## COLLISIONS

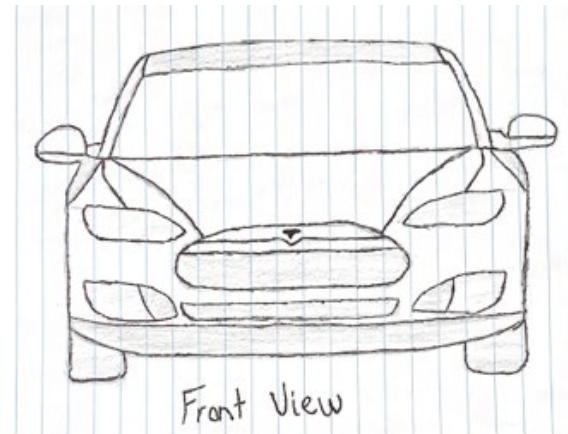
When I first started testing my game I noticed that when the car would ram into the metal barriers it would start flipping and flying through the air it took me a while to fix this but through evaluating different possible causes I eventually determined that the cause was due to the wheel hitbox colliders driving up the bumps on the metal barriers to fix this I increased the size of the car body collider which prevented the car wheels from ramping up the barriers.

## WORKING DRAWINGS



First Track: Interlagos

The black line highlights the area where the road runs then for the rest of the map the areas where there are circles relate to the text nearby such as sand trees and water.



Tesla

### Material Descriptions

Car body: White, Shiny and Metallic

Tires: Black Rubber Texture

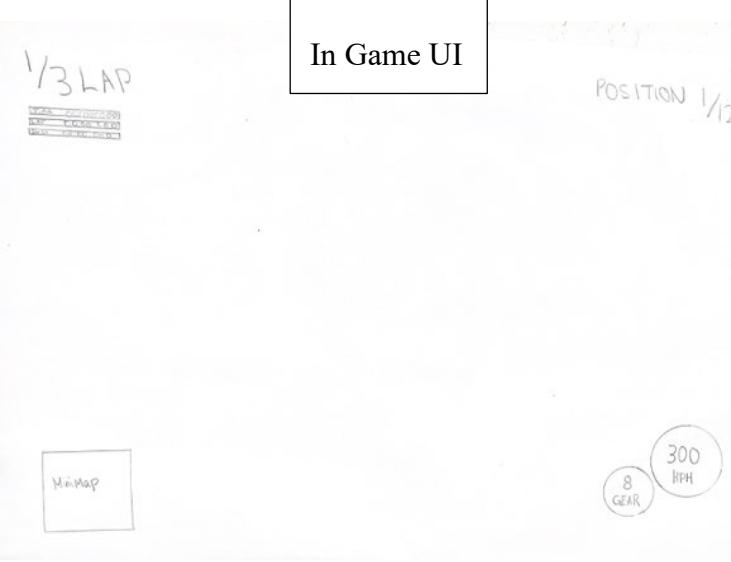
Windows: Glass Textures that will provide a bit of a reflection whilst still appearing see through

Grill: Silver, Shiny and Metallic

Bumpers: Black Carbon Texture

Wheel: Silver, Shiny and Metallic

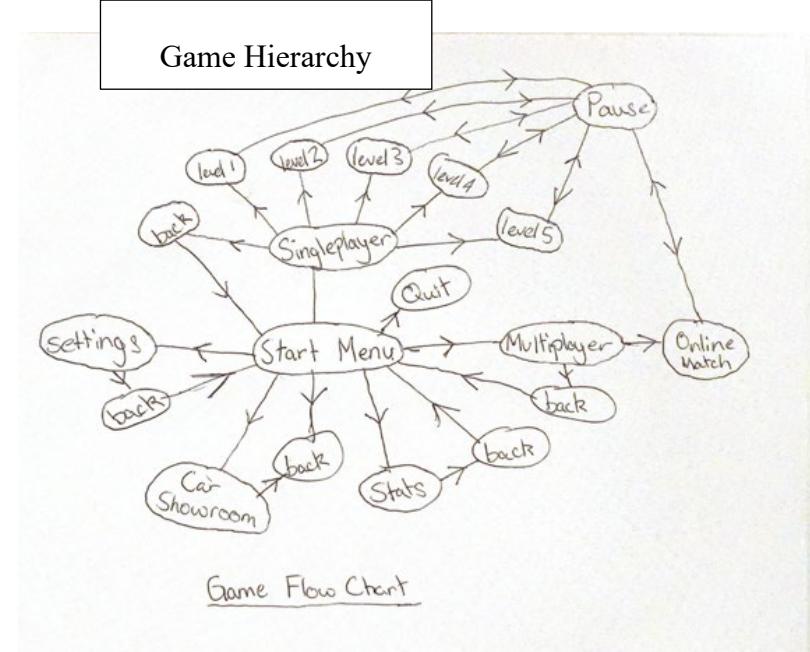
Lights: Glass Textures that will provide a bit of a reflection whilst still appearing see through



In Game UI

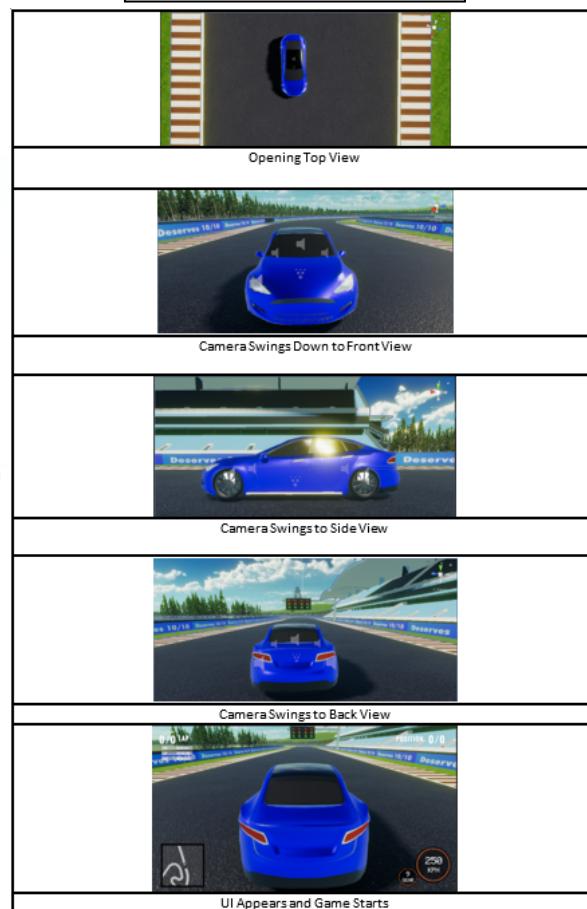


POSITION 1/12

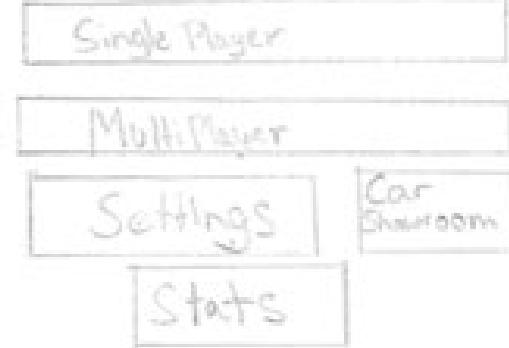


Game Flow Chart

Cut Scene Storyboard



Start Menu UI



# DESIGN MODIFICATION

## STATEMENT OF INTENT

My original plan for my major work was to create an animation that traversed between multiple art styles as well as switching between 2D and 3D. This idea was inspired by the upcoming movie Spider-Man: Across the Spider-Verse which within the trailer demonstrates a similar concept which you can see as the main characters jump between universes. A similar idea was demonstrated within the recent movie Doctor Strange in the Multiverse of Madness in which you see the graphics change as they move between worlds.



## GAME IDEAS

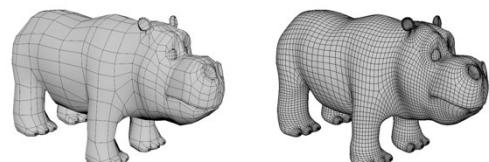
My original idea for my game was to create a 3D runner similar to the game like Vexter. You would have been a car moving down the road trying to avoid different obstacles that would be procedurally generating obstacles such as other cars people and cones. You would have been able to collect power ups and boosters that would give you different advantages making it easier to get further and thus increase your high score.



## CAR MODEL

The first plan I had for my major work cars design was to create it all in low poly but after careful consideration I ended up deciding to model the car in high poly, the difference being in detail of the model. Due to my decision to model in higher quality it means that the process will take longer

Low poly - High Poly





## TIMEPLAN EVALUATIONS

---

I have made this time plan to help organize myself throughout the process of creating this project. Hopefully it will allow me to keep up to date on my Major Work and if I am behind instructing me on where I need to be. Currently (Term 4 week 9) I am on track with everything, and my time plan has assisted me as to not get lost in my process of developing this project.

## TERM 4 2021 EVALUATION

---

The majority of this term was planning and preparing for my project as you can see on the timeplan, I also managed to start modelling and working on my folio.

## SUMMER HOLIDAY EVALUATIONS

---

Throughout the summer holidays I have focused on modelling and texturing the scenes, cars and objectives for my game while doing this I also continued to update my folio on my ongoing work showing my research and updating my record of production.

## TERM 1 2022 EVALUATION

---

Term 1 was considerably less productive than the previous holidays as I had a lot of other schoolwork that pulled my attention away from working on my major work. I still managed to do some modelling each week as well as research and coding. The main thing I did this term was work on my folio so I could focus on my major project throughout the holidays.

## TERM 1 HOLIDAY EVALUATION

---

During the term 1 holidays I got the bulk of my coding for my game done and continued to work on the model of my car. I used this holiday to work within unity more than I had previously developing the user interface and importing the first track into unity.

## TERM 2 2022 EVALUATION

---

This term was by far my most productive as far as advancing my major project goes, I managed to build up the code and UI of my game a lot getting me to a point where I am just left with menus and cinematics to be completed for the holidays.

## TERM 2 HOLIDAY EVALUATION

---

The term 2 holidays have allowed me to finish off my project. These holidays have consisted of pretty much 100% coding and UI/UX design as well as importing the vehicles and maps. Once finished my game I worked on finalising my folio, so it was ready for submission.

## FINANCE PLAN

For my multimedia Major Work, there will be products and software necessary for me to purchase so I am able to complete my game to the best of my ability. I have a budget of \$1,000 for my project but thankfully I already own a lot of the necessary equipment, so I won't exceed my budget.

### THE FINANCE PLAN

Item	Proposed Cost	Actual Cost
<b>Software</b>	<b>\$239.88 + 336.88/annually</b>	<b>\$0</b>
Unity	\$0	\$0
Blender	\$0	\$0
Adobe Creative Cloud	\$19.99/monthly	\$0
Office 365	\$97/annually	\$0
<b>Hardware</b>	<b>\$6421.25</b>	<b>\$0</b>
MacBook Pro (13-inch, M1, 2020)	\$1,899.00	\$0
RTX 2070 Super	\$3004.55	\$0
Ryzen 5 3600x	\$719.70	\$0
Corsair Vengeance RGB PRO SL 32GB (2x16GB) DDR4 3600Mhz	\$239	\$0
Samsung 970 Evo 1TB M.2	\$205	\$0
Motherboard Prime X570-PRO	\$319	\$0
128GB USB	\$35	\$0
<b>Other Processes</b>	<b>\$80</b>	<b>\$80</b>
Printing	\$80	\$80
<b>Total</b>	<b>6741.13 + 336.88/annually</b>	<b>\$80</b>

### FINANCE PLAN EVALUATION

After looking into the potential costs of my multimedia project I am extremely glad that I was able to purchase the necessary computer components last year before the prices were driven up due to bitcoin mining. If I hadn't bought the parts this project may have been impossible for me to make due to the lack of meeting requirements.

# RECORD OF PRODUCTION

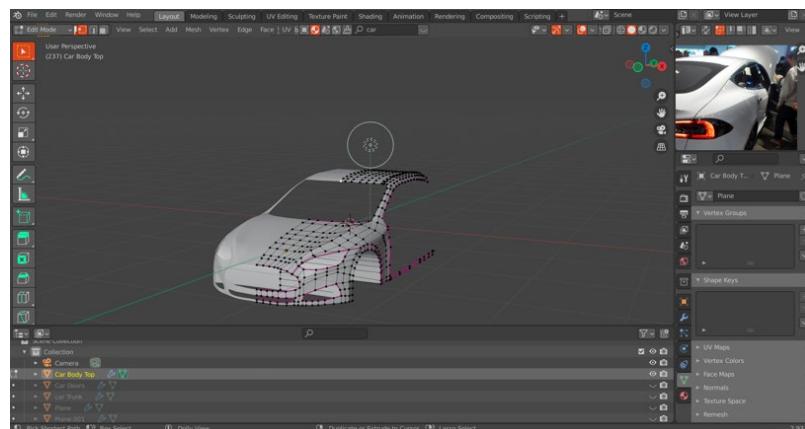
## BLENDER

### EVALUATION

Modelling with blender has been a very slow and tedious process but its flexibility and tools has allowed me to create all the necessary models and animations for my game. Without blender this project would have been impossible for me to do for without spending hundreds of dollars on a subscription software. The 3D modelling process was extremely difficult for me particularly because there were lots of skills which this major work forced me to learn but thankfully a software like this has a massive community which I have been able to use to discover new ways of creating and designing my models for all aspects of my game.

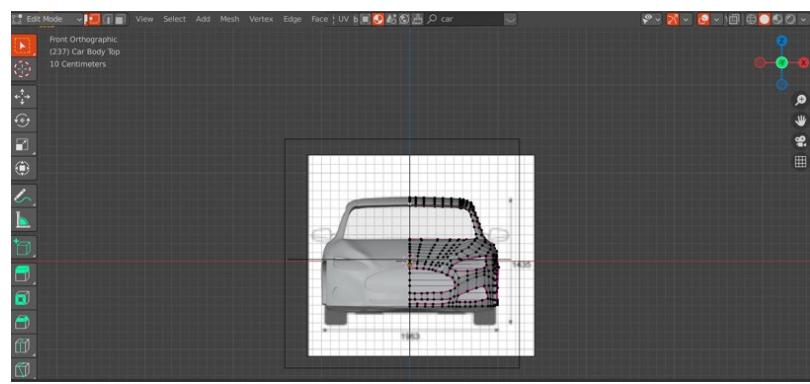
### CAR MODELLING

#### FRONT OF CAR



done this I went to the front view and created a plane which I positioned over the hood of the car and then cut in half along the x axis. I then applied subdivision, solidify and mirror modifiers on to the plane. Then going into edit mode I started to move each vertex to follow along with the blueprints outlines of the hood of the car. After moving the vertices around for a

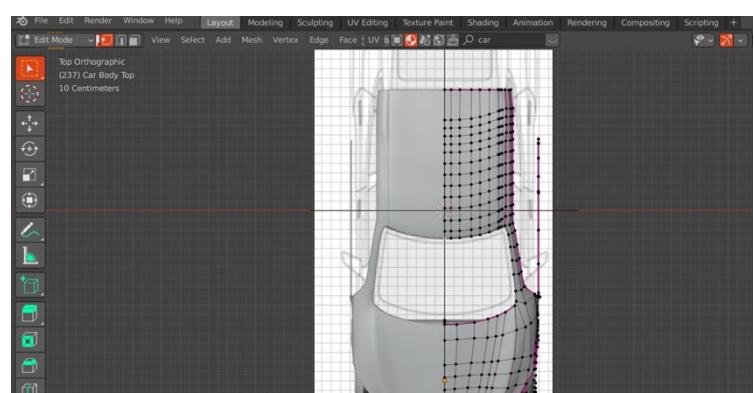
This is my second time around attempting to model this car, this time I knew what I was doing already so the process went by much faster this time around. The first step was to create and invisible box that I could use to make sure the reference images lines up for the front, back and sides. Once I had



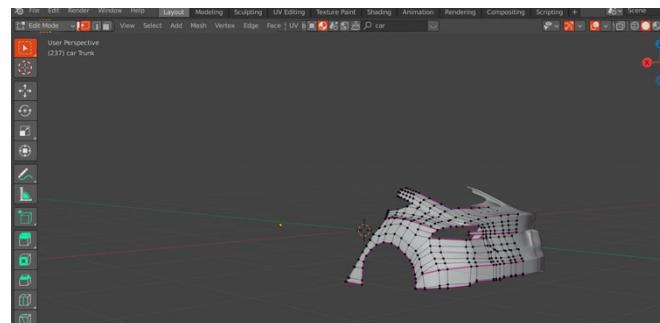
while and creating a shape I was happy with I switched to side view to reshape the hood and make sure it wasn't sitting too high up. The main things I had to do was pitch the hood downwards at the front. Then I started to raise the defining features of the hood. Once the hood was finished, I started extruding the edges down following along the blueprints continuously tweaking and adjusting vertices to create the proper curves making the car look as accurate as possible.

## ROOF OF CAR

Once the front had been modelled, I started working up the sides of the windscreen extruding and scaling the edge to fit the blueprint. Once I had reached the top, I selected the edge and started extruding it inwards along the x axis, once it met in the centre of the mirror, I duplicated the line of edges and slide them up along the y axis. Then I created faces between them and worked to extrude backwards towards the boot adjusting the roof to follow along the curve in the roof of the car.

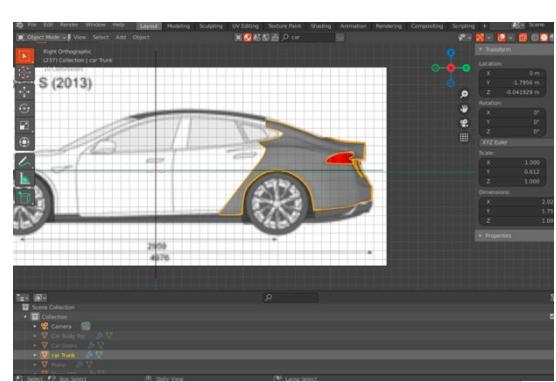


## BACK OF CAR

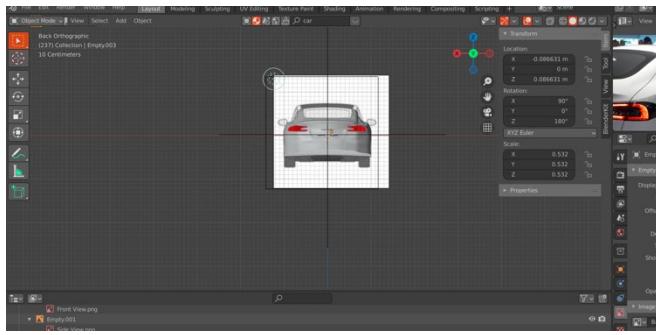


and extruded them down until I hit the top of the boot. Once I had reached the top of the boot, I started extruding the left most vertex all the way to the centre of the mirrored object, then duplicated the whole edge and connected them with faces. After that I started extruding them across the top of the boot on a relatively flat path until I have reached where the car ends according to the blueprint then extruding down along the

Once the front and roof had been modelled, I started to work on the back section of the car and separated the selection to indicate where the front and back is, this also allowed me to hide the front section so I could use the back side of the blueprint. I started by selecting the last 2 edges on the back side of the roof



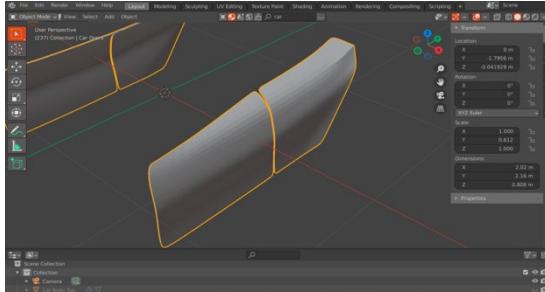
back of the car creating indents to indicate the area the numberplate sits as well as working around leaving holes for where the lights for the car will go. Once I had extruded the boot down to where indicated by the blueprint, I started to extrude the plane around the side of the car making my way towards the back of the rear door and down to the wheel creating the hole around where the wheel will go. Doing this made was quite difficult because I



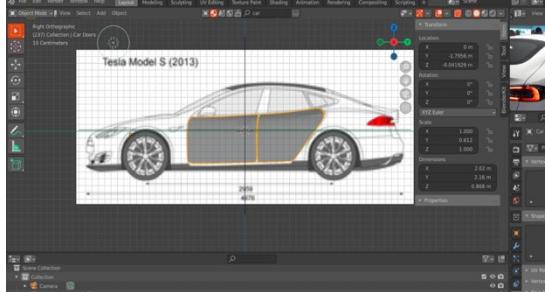
had to do the majority of it while in side view which meant I lost my perception of the depth of the plane when working on those features which meant I had to change the angle constantly and adjust everything. This was made more difficult by the fact that the quality of the blueprints was extremely low as well as didn't line up perfectly which meant some of the adjustments, I made would put it further out of place meaning I had to try to get the right position myself without any proper indication.

## DOOR OF CAR

Modelling the doors was a very challenging process. This process was made challenging due to the fact I was required to adjust the vertices on either side of where the doors fit in so they

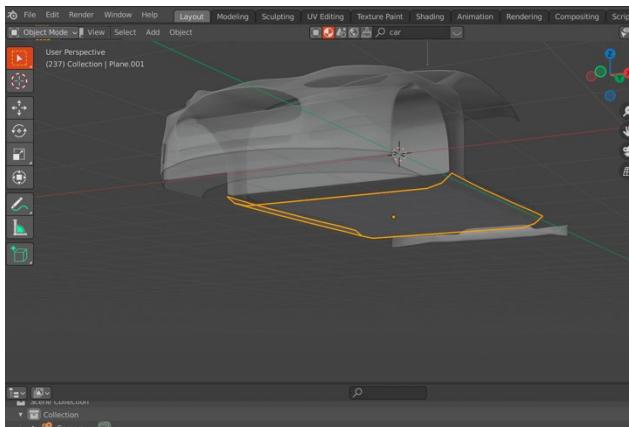


had the same number of vertices that lined up as much as possible without massively effecting the mesh and shaping I had already worked on. Then I copied either side of the now adjusted vertices and pasted them slightly moved inwards towards on the y axis then selecting them and creating a face in between using the loop tools add-on that automatically generated the doors on the correct



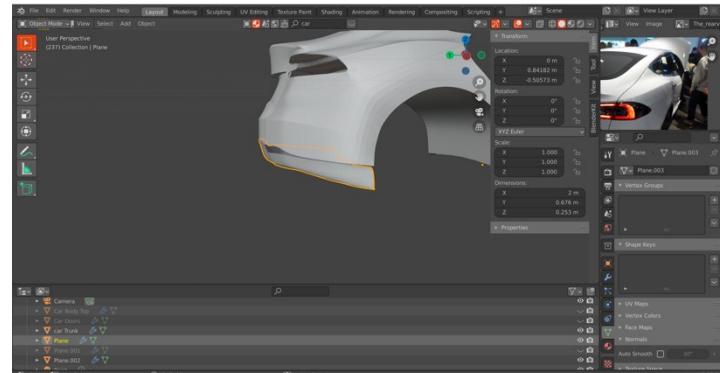
gradient. I then separated the plane to represent that they were two different doors. Once this was done, I begun to adjust the door pulling it along the x axis to change the depth according to the blueprints from all sides making sure to line it up with both the front and back sides of the car that would allow the doors to fit in perfectly with the side of the car as the blueprints show.

## BOTTOM OF THE CAR



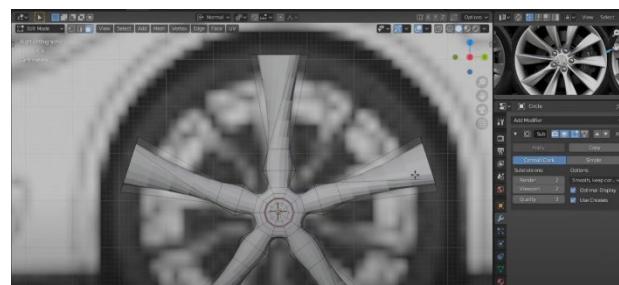
that. Because this was a new plane, I was required to spend time making sure that the plane was placed at the correct level along the z axis before editing from there I extruded the plane along the whole body of the car and then began adjusting the shape of it to fit in with the bottom of the car. Once I had done this, I began to separate the plane into 3 separate pieces as to leave the spaces for the wheels. Now it was in the correct position I started creating a curve up along the edges of the planes so that they met the bottoms of each part of the body. Doing this was an extremely difficult task due to the fact that I was modelling in 3d a curve on what was already a curve

modelling the bottom of this car was my first time doing something like this which made it extremely challenging. The first thing I tried was to extrude the back of the car downwards and work from there but once I had begun that I was presented with challenges of trying to get the mesh in the correct shape and once I had finished modelling it I decided I wasn't happy with the end product so I decided to redo it by creating a new plane and working off of



## WHEELS OF CAR

Modelling the wheels was a difficult task as it required me to include twists and turns that didn't ruin the mesh. Creating the wheels was achieved by following a similar method of going to side view looking at the blueprint and extruding the sides of a circle outwards at the corresponding parts of the blueprint then half way twisting

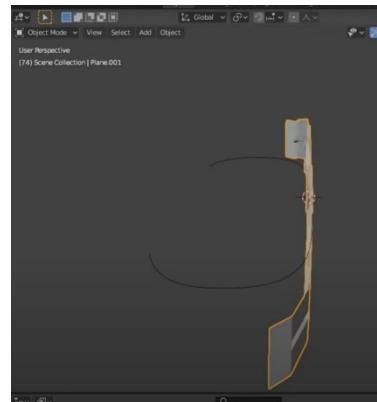


downwards slightly and continuing to extrude. Once this was done for all spokes I selected the centre circle, duplicated it and then scaled it so it created an outer border for the spokes, I then extruded it to create a thicker border.

## TIRES

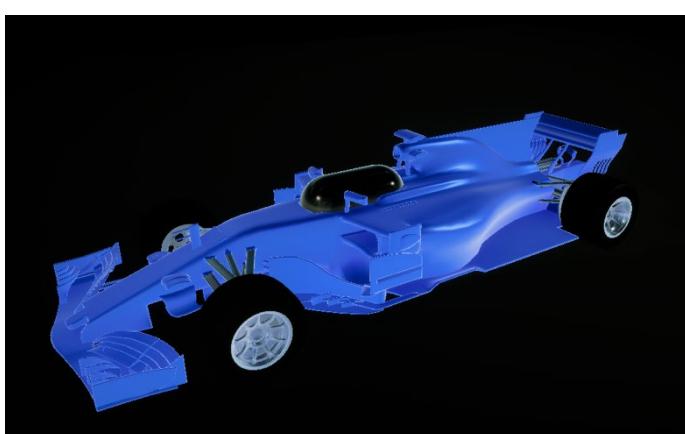
To create the tires, I had to get a reference image so I could map out the correct pattern. Once I had the reference image, I used a plane to recreate the repeating design I then duplicated it and shifted it along bridging the gap between the pattern and the extruding inwards at the corresponding place that are

indented in the design. I then used the curve modifier to make the pattern follow along a Bezier curve that made the pattern curl inwards. I then added a circle and got the new pattern to follow the circle using an array modifier to increase the amount of time the pattern repeated around the circle forming a full tire which I could place over my wheel.



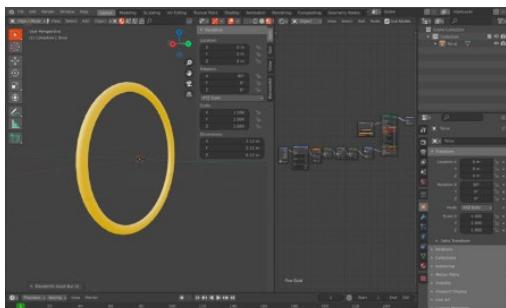
## EVALUATION

The car modelling process was long and would take up all of my folio, so I only have included the challenging parts of the process of modelling my first car to demonstrate my knowledge and proof of work, but I needed to repeat all of these processes multiple times to model each car for the final product of my game.



## ACCESSORY MODELLING

### CHECKPOINT MODELLING FIRST ATTEMPT



Unleashed. I created this by adding in a torus and increasing the size of the centre circle so that the ring was thinner.

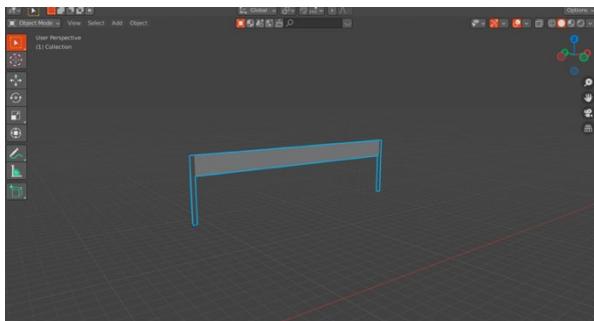
This is my first idea for checkpoints within my game they will be floating golden ring with the opacity turned down to make the checkpoint more transparent. This design was inspired by the checkpoint design in Grand Theft Auto V and the coins in

Sonic

I then subdivided it to increase the level of detail that the ring is, then proceeded to create a material for it that would make it glow, so it was easy for the player to identify. I then decided I did not like this design and began to come up with a new idea



### CHECKPOINT MODELLING FINAL RESULT



race game. This was achieved quite easily by creating a cube stretching it out into a rectangle then using the knife tool to create two small squares on each end which I then extruded downwards.

With my checkpoint model I ended up deciding to go with a more realistic flag type of checkpoint closer to what you would see on an actual racetrack as well as in an old arcade



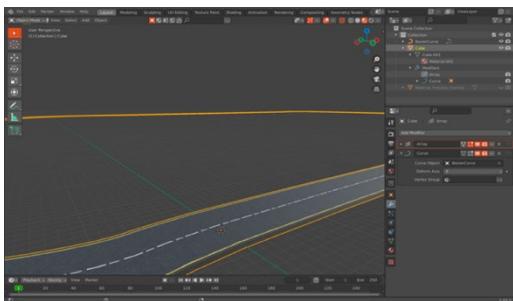
## STARTING LIGHTS



When planning my project, I had trouble trying to decide how I wanted to do my start alert. The main ideas I had for the alert was to either create an overlay on the player screen which would count down 3,2,1 or a traffic light overlay with lights changing between red, amber, and green. The next key idea was to create a physical traffic light which hung above the player's head. When it came down to it, I decided to make the physical light however instead of it being a traffic light I turned it in to more of a board of lights. I did this by modelling a general post by creating a rectangle then cutting a section and extruding it followed by creating a flat rectangle pulling out the sides around it creating a frame and attaching them together, as well as creating a small dome which could represent the light, which later in unity I copied several times and applied materials to give them the proper colours.

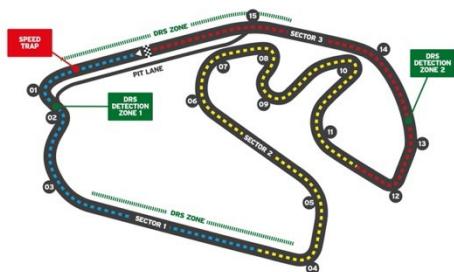
## TRACK MODELLING

### LEARNING HOW TO USE THE MODIFIERS



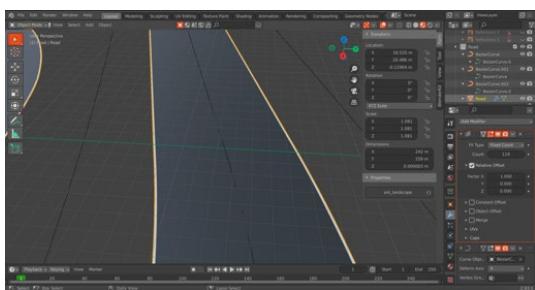
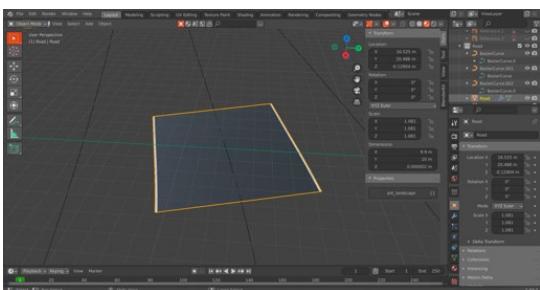
Throughout this process I was trying to learn how to use modifiers so I could design my road with ease. The specific modifiers I ended up using was the subdivider modifier and the array modifier. The process started with creating a Bezier curve that mapped out the path that the road would follow then designing a small portion of road by using a plane then applying a subdivision to the road so that once the array modifier was applied the track would follow the twists and turns without creating jagged angles on turns.

## USING A REFERENCE



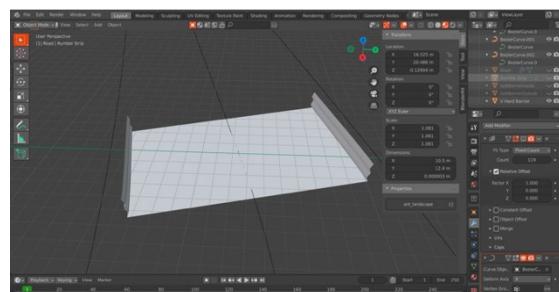
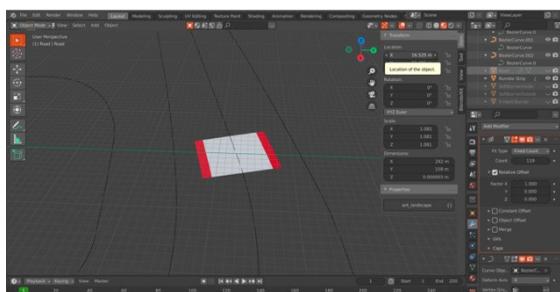
When creating my racetrack, it was important I had an accurate and simple reference to base my track off. Here I have used a 2D image that shows the path that the interlagos track follows, using this I created a Bezier curve that followed the path going down the centre until it connected to the start.

## CREATING THE ROAD

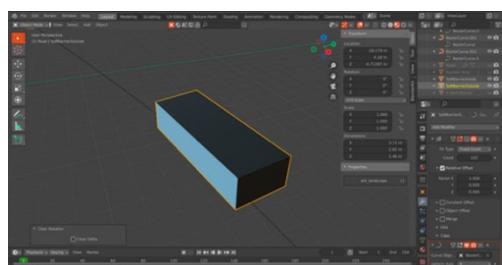


The next step I had to do was create a road segment that I would be able to copy along that would follow the Bezier curve to eventually create the full road for the car to drive on. The next step was to apply the necessary modifiers to the road segment. The first one I added was the curve and array modifier which guided the road to copy itself along the path of the Bezier curve after I had done this, I noticed clipping on some of the sharper corners so to fix this I by using the knife tool to create multiple smaller segments within the segment of road that allowed the road to bend around the corners my nicer without any clipping. Now that I had done this the next step was to adjust the array modifier, so the road had enough segments to fully wrap around the whole track.

I then proceeded to do the same thing with the other segments of the road that surrounded it to make up a full track. These segments included The Rumble Strip and the Hard Barrier.

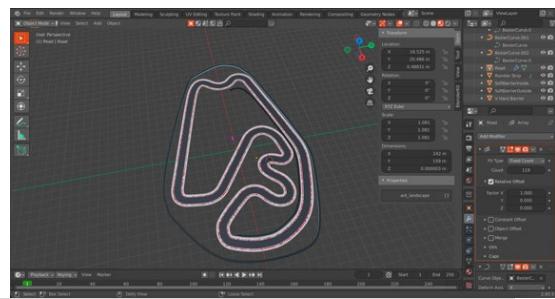


## CREATING THE BARRIERS



Now that my road had been made the next feature, I worked on was creating the soft barriers that surrounded the track this followed a similar process as the road where I used the Bezier curve tool to create two paths, modelled the inner and outer barriers and then applied the required modifiers. The important part however was identifying the correct locations for the

barriers to go. This was done by looking at the track I was referencing in real life and making adjustments in areas where I felt needed to be more covered. The end result was having the outer soft barriers surround the track and the inner soft barriers on the straight section to avoid the cars from being able to jump across the track.



## EDITING THE TRACK



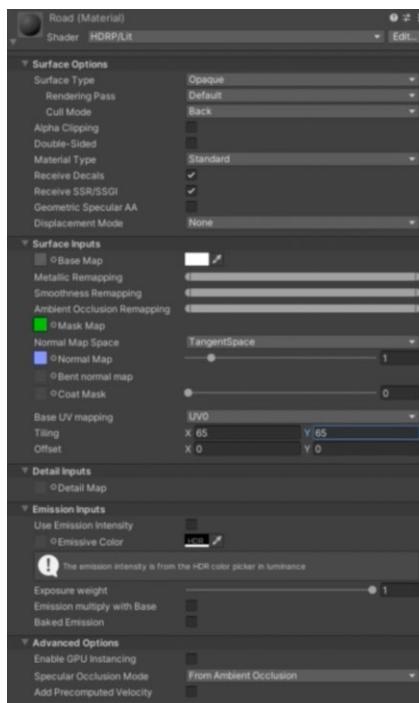
Now that all features of the track had been created it was time for me to apply the modifiers and edit the track to make sure the road didn't get caught in the other segments of track. The first step was to remove the centre part of the rumble strip and hard barrier so that it didn't create a problem with overlapping the road. The next step was to go over the rumble strip segments and cut them out, so they only came up in sections where a rumble strip was necessary such as on corners. I then proceeded to do the same thing with the hard barriers adding them in places where the road could be cut across like in straight sections as shown in the image.

## UNITY

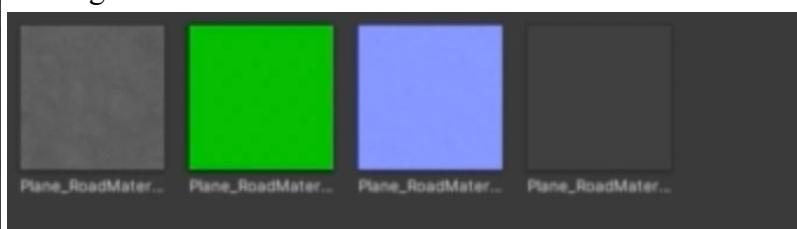
### EVALUATION

Unity has been an extremely useful software throughout this process of creating my major project. Thanks to my familiarity with the software and it having all the features I need, I have been able to put together my major work efficiently and effectively. Becoming familiar with this software has been easy thanks to its simplicity and fantastic UI and UX.

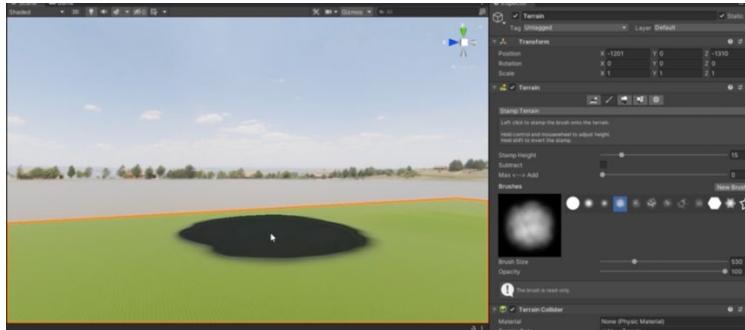
## IMPORTING TRACK



After modelling the track, I started to create materials within unity that I could apply to the different sections of the track to give the impression of driving on a real racetrack. I did this by creating a material and applying each different material to each part of the track that required it, as I did this I went and applied the normal map, mask map and base map then adjusting the tiling and offset to create a look that matched the race tracks I have seen before such as the interlagos track. This is how I created the red and white pattern on my rumble strips that you commonly see on racing circuits.



## ADDING THE LANDSCAPE



Once I finished importing and adding materials, I worked on adding a landscape around the track filled with hills to give the environment. To create this landscape, I used the terrain tool which is fitted with all sorts of features that allow sculpting of the plane creating hills and valleys that fill up the world around the racetrack. I also decided to use the texture brush to add areas within the racetrack that is surrounded by sand that will have different effects on the steering of the car due to the traction it gets.

## SCALING THE MAP

When I originally imported the interlagos map to my game I decided that the feel of the map was too small so I scaled the map and terrain by 100 on the x, y and z coordinates, this left me with much more room to work with and gave the car a greater distance to cover whilst travelling at high speeds increasing the average lap time by multiple seconds.

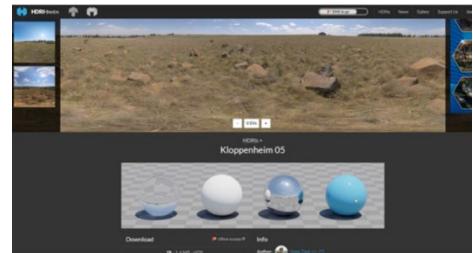


## ADDING TREES AND SKYBOX



The environment within my game was still quite empty so I wanted to add more and make it feel like more of a world. So, I began to add some trees I did this by getting a bunch of trees into a pack and randomly painting them around the empty areas of the map where there was only grass. At first this seemed fine and looked good but over time I realised that

there were too many trees with meaning there were too many vertices creating significant lag and drops in fps. Because of this I opted to have fewer trees but much closer together creating a wall around the track that gave the impression that the track was surrounded by trees in a forest. The next thing I did to increase the quality of the world around me was download a realistic skybox making the world have a much nicer view.



## IMPORTING CAR MODEL

Importing the car model was a very simple process of exporting as .fbx from blender then dragging and dropping the car into the scene which was saved as a prefab so all the material edits, I made stay consistent throughout the whole game. Once it was in, I just assigned each wheel to the corresponding wheel collider in the script and then proceeded to line up each wheel with its corresponding wheel collider, so the driving experience felt more accurate.



## EVALUATION

Once the first track had been created to my satisfaction, I repeated the process for my other maps adjusting the scene and scaling till I saw fit.

## PROGRAMMING/USER INTERFACE

### UI SCRIPT

The UI script is used to tell the game when to display what on the users screen a key example of this is the Lap banner that appears when you complete a lap and the lap time that is displayed in the corners along with position, lap number and speed.



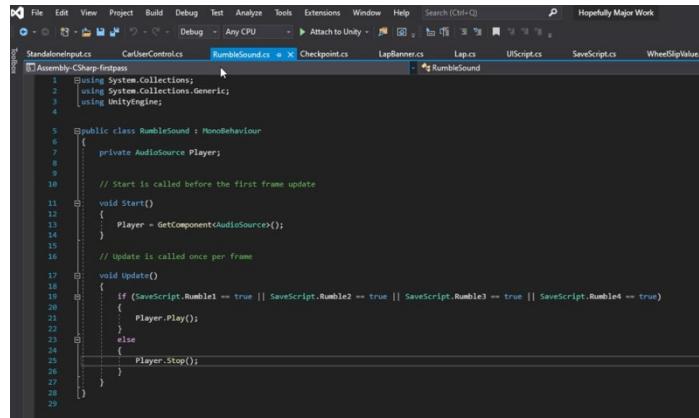
### SAVE SCRIPT

The Save Script I have in my game is used to store variables this is why throughout my code you can see SaveScript."Variable Name" this is the code I used to tell the script I am currently in to search through the folder and find the script and within that script to find that variable.



## RUMBLE SOUND SCRIPT

This script's purpose was to call the audio file attached to the wheels when it detected that the wheels were colliding with an object that had the tag "rumble" which was applied to the rumble strips along the sides of the track. This script would detect each individual wheel and play the audio from the wheel accordingly giving the environment and racing experience a more realistic feel.



```
File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+F) P Hopefully Major Work

StandaloneInput.cs CarUserControls.cs RumbleSound.cs Checkpoint.cs LapBanner.cs Lap.cs UIScript.cs SaveScript.cs WheelSlipValue.cs

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

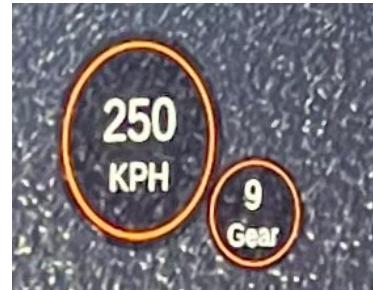
public class RumbleSound : MonoBehaviour
{
    private AudioSource Player;

    // Start is called before the first frame update
    void Start()
    {
        Player = GetComponent<AudioSource>();
    }

    // Update is called once per frame
    void Update()
    {
        if (SaveScript.Rumble1 == true || SaveScript.Rumble2 == true || SaveScript.Rumble3 == true || SaveScript.Rumble4 == true)
        {
            Player.Play();
        }
        else
        {
            Player.Stop();
        }
    }
}
```

## SPEED AND GEAR DISPLAY

Through a mix of scripting UI design, I have created a display in the bottom right corner which shows the speed as well as a border which increases around the speedometer, on top of this I have also used scripting and UI design to display the gear the car is set in.



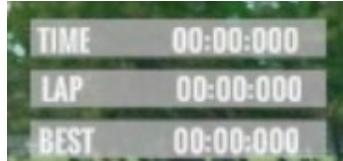
## MINIMAP DISPLAY

Setting up the minimap was an easy feat to accomplish I achieved this by creating a new camera that had a bird eye view of the track, after this I duplicated the road and created a new layer for the camera to see. I then created an image which updated to display the view of the camera and positioned it in the bottom left corner of the screen. This created a white road that appeared in the bottom corner. The next step was to add an object set to the same layer which was parented to the player allowing for the camera to see the object whilst it follows the player creating a blip on the minimap that showed the players position on the track. Once that was setup, I adjusted the zoom and transparency as well as adding a border to give the map a more polished finish. I then used a script to change it the minimaps active camera and increase the box size whenever I would press tab so you could see the whole map.



## TIME, POSITION AND LAP DISPLAYS

First thing I set up was the position sign which was done with an easy creating of 3 text boxes one that displayed “Your position” another that displayed the divider and the last one that displayed the total amount of players. I then created a similar group of text for the lap text which displayed



the lap you are on the divider and the total amount of laps. The current lap and total laps were then later hooked up to the laps script which changed the displayed number. To create a space for the timers I created a grey panel and positioned it into the top left corner that displayed minutes, seconds, and milliseconds I then copied and pasted this three times shifting down slightly to create three spaces that displayed different times labelled as “Time”, “Lap” and “Best” representing the total time of the race, the total time of the current lap and the time of your best lap so far.



## LAP BANNER

Creating the lap banner was simple all I did was add a panel change the colour to black and add some text over it that said “Lap” and “Lap Number” I then connected the lap number text to a script that would adjust the value to the correct number. I then attached the banner to a script which would activate it whenever you passed the lap checkpoint at the starting lights.



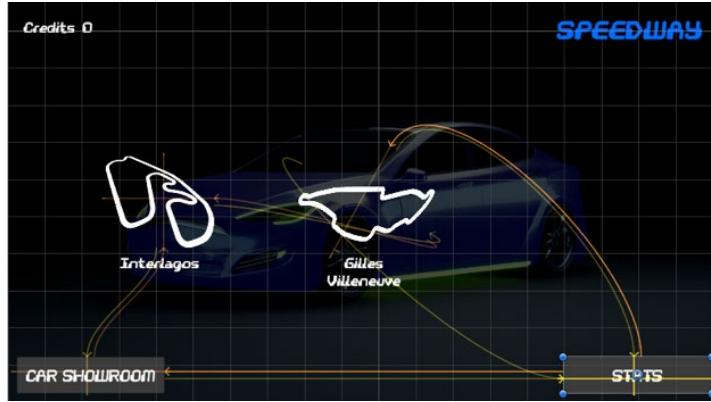
## CHECKPOINT BANNER

I repeated the same process to create the checkpoint banner as I did for the lap banner except this time, I added a feature that used some code to measure the time since the last time between checkpoint against your current checkpoint time and telling you whether you were faster or if you were slower.



## BUTTONS

Through a mix of use of the unity events system and scene management system I added into a few scripts that I could apply to buttons so I could switch between scenes allowing me to access menus and the different game modes. Once I had set it up so the buttons would move you between scenes, I needed to make it so the events system would have a button to highlight on the next screen so the controller could then navigate between buttons allowing for users to operate the game with just a controller.



## CAR USER CONTROLLER

My car user controller script is in charge of controlling many different functions that work to translate how the car can and will move from the players inputs. The movement of the car is done by taking the input of the player and applying speed on that axis. This script is paired with another script called car controller that works to set the cars top speed, translate the speed into kilometres per hour, make the steering feel accurate to real driving using things such as max steer angle as well as make the car complete all aesthetic tasks such as playing skid sounds, setting a gear value based on the top speed divide by the amount of gears with a curve factor so each gear pushes further making the game between 2<sup>nd</sup> and 3<sup>rd</sup> greater than 1<sup>st</sup> and 2<sup>nd</sup>.

```
@Unity Message | 0 references
private void FixedUpdate()
{
    if (SaveScript.RaceStart == true)
    {

        // pass the input to the car!
        float h = CrossPlatformInputManager.GetAxis("Horizontal");
        float v = CrossPlatformInputManager.GetAxis("Vertical");

        if (CrossPlatformInputManager.GetButton("Fire1"))
        {
            v = 2.0f;
        }
        if (CrossPlatformInputManager.GetButton("Fire2"))
        {
            v = -0.5f;
        }
        if (!CrossPlatformInputManager.GetButton("Fire1") && !CrossPlatformInputManager.GetButton("Fire2") && !CrossPlatformInputManager.GetButton("Vertical"))
        {
            v = 0.0f;
        }
    }
}
```

## STARTING LIGHTS

The starting lights script is used to let the player know how long they have until the race or time trial starts this is done by switching a series of objects on and off with yields in between the process all starts after 20 seconds which is the length of the opening cut scene and then itself take 3.5 seconds to start the race and trigger the race start bool which allows the player access to the car user controller script allowing them to start driving, then an additional 1.5 seconds to turn off the final “go” message. As an added feature to increase the quality of the game the lights also make use of beep sounds with a change in pitch for the final one that signifies the player can go as seen commonly in racing games.

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

public class StartingLights : MonoBehaviour
{
    void Start()
    {
        StartCoroutine(StartingLights());
    }

    IEnumerator StartingLights()
    {
        //Red Lights
        yield return new WaitForSeconds(20);
        RLIGHTOFF.SetActive(false);
        RLIGHTON.SetActive(true);
        RNUMBER.SetActive(true);
        SOUND1.Play();
        yield return new WaitForSeconds(1);
        RLIGHTOFF.SetActive(true);
        RLIGHTON.SetActive(false);
        RNUMBER.SetActive(false);
        SOUND1.Play();
        //Amber Lights
        ALIGHTOFF.SetActive(false);
        ALIGHTON.SetActive(true);
        ANUMBER.SetActive(true);
        yield return new WaitForSeconds(1);
        ALIGHTOFF.SetActive(true);
        ALIGHTON.SetActive(false);
        ANUMBER.SetActive(false);
        SOUND1.Play();
        //Green Lights
        GLIGHTOFF.SetActive(false);
        GLIGHTON.SetActive(true);
        GNUMBER.SetActive(true);
        yield return new WaitForSeconds(1);
        GLIGHTOFF.SetActive(true);
        GLIGHTON.SetActive(false);
        GNUMBER.SetActive(false);
        GO.SetActive(true);
        SOUND2.Play();
        yield return new WaitForSeconds(0.5f);
        SAVESCRIPT.RaceStart = true;
        yield return new WaitForSeconds(1.5f);
        GO.SetActive(false);
    }
}
```

## TIME TRIAL SCRIPT

```
if(SaveScript.RaceOver == true)
{
    if(Winner == false)
    {
        Winner = true;
        StartCoroutine(WinDisplay());
    }
}

if (CrossPlatformInputManager.GetButton("Cancel") || Input.GetKey("escape"))
{
    QuitPanel.SetActive(true);
    Time.timeScale = 0;
    //clear selected object
    EventSystem.current.SetSelectedGameObject(null);
    //Set New Selected Object
    EventSystem.current.SetSelectedGameObject(QuitPanelIN);
    Debug.Log("Quit");
}

}

IEnumerator WinDisplay()
{
    yield return new WaitForSeconds(0.15f);
    ResultsMenu.SetActive(true);
    Time.timeScale = 0.2f;
    //clear selected object
    EventSystem.current.SetSelectedGameObject(null);
    //Set New Selected Object
    EventSystem.current.SetSelectedGameObject(ContinueButtonResultsTimeTrial);
    if (SaveScript.Gold == true)
    {
        if (SaveScript.PenaltySeconds == 5)
        {
            PenaltyText.SetActive(true);
        }
        ResultsText.text = "You Won Gold";
        GoldStar.SetActive(true);
        ResultsMenu.SetActive(true);
        Credits.text = GoldCredits.ToString();
        UniversalSave.CreditAmount = UniversalSave.CreditAmount + GoldCredits;
        UniversalSave.GoldWon++;
        //clear selected object
        EventSystem.current.SetSelectedGameObject(null);
        //Set New Selected Object
        EventSystem.current.SetSelectedGameObject(ContinueButtonResultsTimeTrial);
    }
}
```

The time trial script is used to calculate the award given to the player according to the time the player got vs the times set for the award for instance, if

```
void Update()
{
    //Gold Minutes
    if (SaveScript.TimeTrialMinG <= 9)
    {
        TimeTrialMinutesG.text = "0" + SaveScript.TimeTrialMinG.ToString() + ":";
    }
    if (SaveScript.TimeTrialMinG >= 10)
    {
        TimeTrialMinutesG.text = SaveScript.TimeTrialMinG.ToString() + ":";
    }

    //Gold Seconds
    if (SaveScript.TimeTrialSecondG <= 9)
    {
        TimeTrialSecondsG.text = "0" + SaveScript.TimeTrialSecondG.ToString() + ":";
    }
    if (SaveScript.TimeTrialSecondG >= 10)
    {
        TimeTrialSecondsG.text = SaveScript.TimeTrialSecondG.ToString() + ":";
    }

    //Gold Milliseconds
    if (SaveScript.TimeTrialMillisecondsG <= 9)
    {
        TimeTrialMillisecondsG.text = "00" + SaveScript.TimeTrialMillisecondsG.ToString();
    }
    if (SaveScript.TimeTrialMillisecondsG >= 10)
    {
        TimeTrialMillisecondsG.text = "0" + SaveScript.TimeTrialMillisecondsG.ToString();
    }
    if (SaveScript.TimeTrialMillisecondsG >= 100)
    {
        TimeTrialMillisecondsG.text = SaveScript.TimeTrialMillisecondsG.ToString();
    }
}
```

gold was set at 40 seconds the player must achieve a lap time of 40 seconds or less to qualify for the gold award however if they trigger the penalty bool which triggers if cheating is detected 5 additional seconds is added to your time so if you may have reached the point in 39 seconds but you skipped a turn the game will trigger the penalty bool and add 5 seconds so your new time will be 44 seconds.

## PRACTICE SCRIPT

The practice script is very simple and all it does is set the bool race start to true by default, so the player is able to just go and practice their driving on the chosen map as soon as the animation is done. I included this so the player had the opportunity to perfect their runs and become the best racer they can be by finding the optimal path to take when driving each map.

```
using UnityEngine;
using System.Collections;

public class True : MonoBehaviour
{
    public bool Practice = true;

    // Start is called before the first frame update
    @Unity Message | 0 references
    void Start()
    {
        if(Practice == false)
        {
            SaveScript.RaceStart = true;
        }
    }

    // Update is called once per frame
    @Unity Message | 0 references
    void Update()
    {
        if (Practice == true)
        {
            StartCoroutine(PracticeStart());
        }
    }

    IEnumerator PracticeStart()
    {
        yield return new WaitForSeconds(20);
        SaveScript.RaceStart = true;
    }
}
```

## RACE SCRIPT

```
if (FinishLine.PlayerFinishPosition > 3)
{
    WinMessage.text = FinishLine.PlayerFinishPosition + "th Place";
    Credits.text = "0";
    UniversalSave.RacesLost++;
    UniversalSave.Races++;
}
UniversalSave.Saving = true;

}

0 references
public void DisplayLeaderboard()
{
    this.gameObject.SetActive(false);
    Leaderboard.SetActive(true);
    Time.timeScale = 0;
    //Clear selected object
    EventSystem.current.SetSelectedGameObject(null);
    //Set New Selected Object
    EventSystem.current.SetSelectedGameObject(ContinueButton);
}

0 references
public void BackToMenus()
{
    SceneManager.LoadScene(1);
}
```

```
void Start()
{
    Leaderboard.SetActive(false);

    if(UniversalSave.OpponentsCount > 0)
    {
        FirstPlaceCredits *= UniversalSave.OpponentsCount;
        SecondPlaceCredits *= UniversalSave.OpponentsCount;
        ThirdPlaceCredits *= UniversalSave.OpponentsCount;
    }

    if(FinishLine.PlayerFinishPosition == 1)
    {
        WinMessage.text = "1st Place";
        Credits.text = FirstPlaceCredits.ToString();
        UniversalSave.CreditAmount = UniversalSave.CreditAmount + FirstPlaceCredits;
        UniversalSave.RacesWon++;
        UniversalSave.Firsts++;
        UniversalSave.Races++;
    }
}
```

This script is used to measure the position the player places in and assign the credits accordingly to adjust the credits you gain I have used some maths to adjust the amount of credit paid out according to the amount of opponents you faced so that if it is just a 1v1 you get significantly less credits than if you vs 7 opponents. This script is also used to display the leaderboard once the player has crossed the finish line and set the continue button to the selected button in the events system.

## LEADERBOARD

```
public TextMeshProUGUI Position;
public TextMeshProUGUI Name;
public TextMeshProUGUI RTMin;
public TextMeshProUGUI RTSec;
public TextMeshProUGUI Milliseconds;
public TextMeshProUGUI Player;

// Update is called once per frame
@Unity Message | 0 references
void Update()
{
    if(true == true)
    {
        Position.text = FinishLine.PlayerFinishPosition.ToString();
        Name.text = FinishLine.PlayerName;
        RTMin.text = FinishLine.PlayerRTMin.ToString();
        RTSec.text = FinishLine.PlayerRTSec.ToString();
        Milliseconds.text = FinishLine.PlayerMilliseconds.ToString();
    }
}

if(FinishLine.PlayerRTMin >= 0)
{
    Minutes.text = "0" + Mathf.Round(FinishLine.PlayerRTMin).ToString() + ":";

    if(FinishLine.PlayerRTMin >= 10)
    {
        Minutes.text = Mathf.Round(FinishLine.PlayerRTMin).ToString() + ":";

        if(FinishLine.PlayerRTMin < 90)
        {
            Seconds.text = "0" + Mathf.Round(FinishLine.PlayerRTSec).ToString() + ":";

            if(FinishLine.PlayerRTSec < 90)
            {
                Seconds.text = "0" + Mathf.Round(FinishLine.PlayerRTSec).ToString() + ":";

                if(FinishLine.PlayerRTSec >= 10)
                {
                    Seconds.text = Mathf.Round(FinishLine.PlayerRTSec).ToString();
                }
            }
        }
    }
}

if(FinishLine.PlayerRTSec >= 0)
{
    Milliseconds.text = "00" + Mathf.Round(FinishLine.PlayerMilliseconds).ToString();

    if(FinishLine.PlayerMilliseconds < 99)
    {
        Milliseconds.text = "0" + Mathf.Round(FinishLine.PlayerMilliseconds).ToString();
    }
}

if(FinishLine.PlayerRTSec >= 100)
{
    Milliseconds.text = Mathf.Round(FinishLine.PlayerMilliseconds).ToString();
}
```

The leaderboard was a simple process that took the saved value of the cars position and race time then organised it into the categories then placed the information in the outlined area made for the position you placed. This was a quick and easy feature to add to the game that improved the overall experience.

```
if (Player == true)
{
    Position.text = FinishLine.PlayerFinishPosition.ToString();
    Name.text = FinishLine.PlayerName;
    //Minutes
    if (SaveScript.RTMin < 9)
    {
        Minutes.text = "0" + Mathf.Round(SaveScript.RTMin).ToString() + ":";

        if (SaveScript.RTMin >= 10)
        {
            Minutes.text = Mathf.Round(SaveScript.RTMin).ToString() + ":";

            if (SaveScript.RTMin < 90)
            {
                Seconds.text = "0" + Mathf.Round(SaveScript.RTSec).ToString() + ":";

                if (SaveScript.RTSec >= 10)
                {
                    Seconds.text = Mathf.Round(SaveScript.RTSec).ToString() + ":";

                    if (SaveScript.RTSec < 90)
                    {
                        Milliseconds.text = "00" + Mathf.Round(SaveScript.RTMil).ToString();

                        if (SaveScript.RTMil < 99)
                        {
                            Milliseconds.text = "0" + Mathf.Round(SaveScript.RTMil).ToString();
                        }
                    }
                }
            }
        }
    }
}

if (SaveScript.RTMil < 99)
{
    Milliseconds.text = "00" + Mathf.Round(SaveScript.RTMil).ToString();

    if (SaveScript.RTMil >= 100)
    {
        Milliseconds.text = Mathf.Round(SaveScript.RTMil).ToString();
    }
}
```

## CAR AI CONTROL

The car AI control works by having AI target waypoints I have set out that follows the track and measure the angle and distance between waypoints to see if it needs to slow down and turn harder or speed up and stay on a straight path.

```
private void FixedUpdate()
{
    if (_Target == null || !_m_Driving)
    {
        // Car should not be moving,
        // use handbrake to stop
        m_CarController.Move(0, 0, -1f, 1f);
    }
    else
    {
        if (SaveScript.RaceStart == true && RaceCompleted == false)
        {
            Vector3 fwd = transform.forward;
            if (_Rigidbody.velocity.magnitude > m_CarController.MaxSpeed * 0.1f)
            {
                fwd = _Rigidbody.velocity;
            }
        }
    }
}
```

## WRONG WAY

An important part of a race game is the crashing and spinning out when this happens the player needs a way to make sure they are racing in the right direction so I set up a system of cubes with trigger colliders that follow along the track with a script on the them that assign them values which increase as you progress that communicates to the game that the player is moving in the right direction so when the game sees that the player goes from a higher number to a lower number by a set amount and will notify the player with a banner that says “wrong way” until it detects the player moving in the correct direction again.

```
if(LastWPNumber > ThisWPNumber)
{
    SaveScript.WrongWay = false;
}

if (LastWPNumber < ThisWPNumber)
{
    SaveScript.WrongWay = true;
}
```

## MULTIPLE CARS

Setting up multiple cars was a feature I hadn't really considered for the first couple months of my project but decided it would be a cool addition to include to my game, where depending on your car of choice the cars will handle differently, accelerate differently, and have different top speeds. By changing the car, you also get to race against other types of cars providing new challenges when you play. I did this by setting up an integer that translated to Tesla = 1, F1 = 2 and Ferrari = 3 I used these variables and if statements to trigger the different ai cars and player cars into spawning when you first load into the race.

```
public int RaceCarType = 1;

//Tesla
public GameObject tAIOpponent1;
public GameObject tAIOpponent2;
public GameObject tAIOpponent3;
public GameObject tAIOpponent4;
public GameObject tAIOpponent5;
public GameObject tAIOpponent6;
public GameObject tAIOpponent7;

//F1
public GameObject f1AIOpponent1;
public GameObject f1AIOpponent2;
public GameObject f1AIOpponent3;
public GameObject f1AIOpponent4;
public GameObject f1AIOpponent5;
public GameObject f1AIOpponent6;
public GameObject f1AIOpponent7;

//Ferrari
public GameObject fAIOpponent1;
public GameObject fAIOpponent2;
public GameObject fAIOpponent3;
public GameObject fAIOpponent4;
public GameObject fAIOpponent5;
public GameObject fAIOpponent6;
public GameObject fAIOpponent7;
```

## WORK, HEALTH & SAFETY

WH&S is an extremely important thing I need to consider while I am working on my Major Work so I can minimize the risk I am at while working. Through-out the process of making my project I will never be in any places that I am at any more risk than usual. I will be using a matrix table to rate the possible risk.

	Insignificant	Minor	Moderate	Major	Severe
Very Unlikely	Very Low	Very Low	Low	Medium	Medium
Unlikely	Very Low	Low	Medium	Medium	Medium
Possible	Low	Medium	Medium	Medium	High
Likely	Medium	Medium	Medium	High	Extreme
Very Likely	Medium	Medium	High	Extreme	Extreme

## ERGONOMICS

Ergonomics is the process of designing and arranging a workspace to fit and suit the required process and make sure it is as comfortable as possible. In the following images there are two chairs the good chair and the bad chair as you can see the good chair has a back rest which Supports your back and prevents stress on your back that may possibly lead to back injuries.



Good Chair



Bad Chair

## COVID-19

COVID-19 has been around causing several lockdowns since 2020 this has reduced the productivity that students and teachers have been able to make at school. To prevent the spread of COVID-19 students and teachers must wear face masks when inside at all times, as well as practice social distancing to make sure that COVID-19 has the smallest possible chance to spread. It has now become common to see signs wherever you go that have been put up by the government indicating how to protect yourself from the virus. Since the start of the COVID-19 pandemic many students at Killara high school have had to self-isolate due to contracting one of the many different strains of the COVID-19 virus. Since the beginning of the pandemic governments have been trying to find ways to contain and weaken the virus the most commonly seen one now in 2022 is the use of the COVID-19 vaccinations the most prominent ones being Pfizer, AstraZeneca and Moderna as of the 26<sup>th</sup> of May the Australian Government have been pushing people within their borders to get their third booster shot. However, there are people all around the world who have collectively refused to receive their COVID-19 vaccinations, it is also common to see people who refuse to wear masks. Both of these factors increase the likelihood of catching COVID-19 which may interrupt my major work progress.



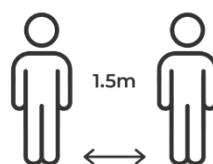
Does Not Wear Mask



Signage



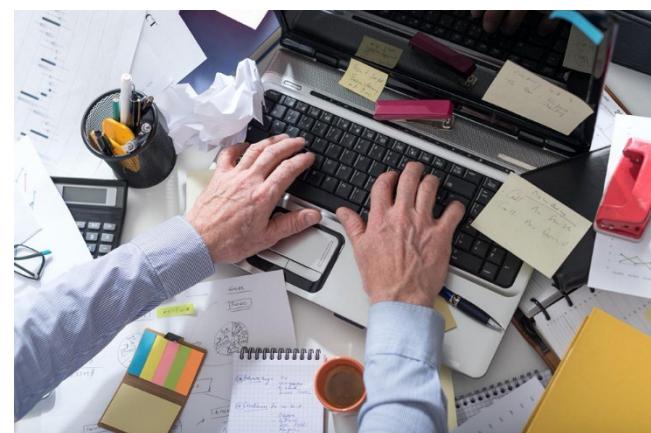
Wears mask



Social Distancing

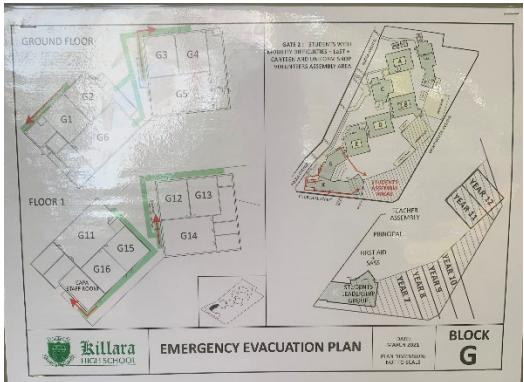
## HAZARDS

A Hazard is any potential danger or risk. The risk they pose in the workplace is great and can be anything from untidy cables that could be tripped on to fire hazards. It is important to identify hazards so that people are aware of what to look out for. This prevents them from getting hurt

Potential Hazard	Example
Untidy Cables	
Fire Hazard	
Obstructed Workplace	

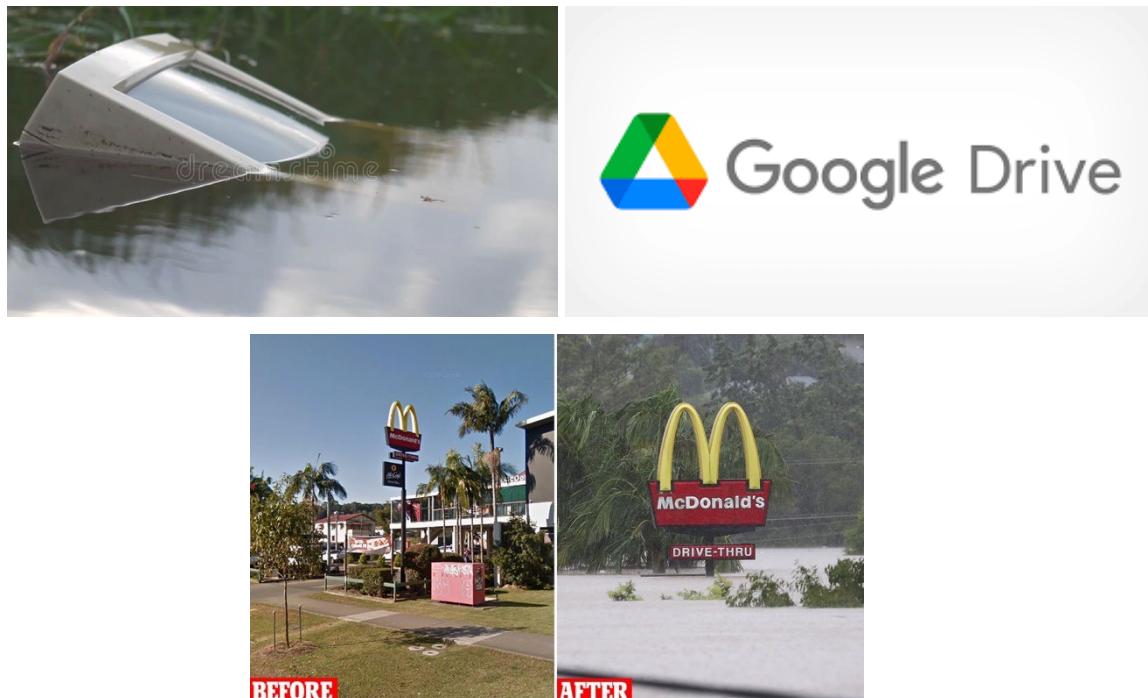
## SIGNAGE

Signs are put around workplaces to warn people nearby of any potential hazards and in some cases how to deal with the hazard.

Sign	Example
Fire Extinguisher	
COVID-19 Spread Prevention	
Evacuation Plan	

## FLOODING

The recent weather in Sydney has caused serious flooding which has extremely affected some people in the area, destroying homes, possessions and more specifically to this project it has destroyed thousands of electronics. If this were to happen in my local area there is a chance it could permanently halt the production of my multimedia major work by destroying my computer and the files on it. The reason these have the ability to stop my work completely is due to the fact that I require the power of my pc to complete this project and it would be too expensive to repair or possibly even repair. These dangers can be subverted by saving files to online servers and placing my computer in an elevated location to prevent the water from destroying the pc components.



## EVALUATION

The process of creating my major work has been long and still has a lot to go. The graphics card in my desktop has recently failed for an unknown reason because of these I have been forced to put in my old graphics card the GTX 970 into my current pc which has been significantly slower and has temporarily slowed down the process of me completing my multimedia major work. This downgrade in graphics card has also prevented me from being able to utilise my other monitors making my work significantly less productive. I am currently waiting to hear back from msi about my RTX 2070 Super's warranty, if for whatever reason I am unable to claim it I will need to purchase a new graphics card in order to complete my multimedia major work.

## FINAL EVALUATION



Overall, I am extremely pleased with how my major work has turned out. Through this project I have learned new skills which have allowed me to refine the game and create a final product of high quality. I believe this project manages to show off my skills and understanding in programs such as blender, unity, and visual studio. The actual gameplay of my major work I am extremely proud of as I know that my friends and I all enjoy playing it ourselves due to all the in-game functions working exactly as intended. The skills I have successfully demonstrated through this project are my ability to code, model, animate and create a good UI/UX.

## STATEMENT OF INTENT

*"For my Year 12 major work, I intend to code and create a video game using a game engine such as Unity, Unreal Engine or GADOT. This game will use a 3D model, which I will model, texture, and animate and will be driving around in a 3D environment which I will also design. This 3D game will also include features such as music, multiple levels, title screen, save and load functions, high-scoring system."*

The Final product of my major work is even better than I even initially set out to achieve. I successfully created and coded the video game designing all the features to make the game as challenging and fun as possible. The 3D modelling and animation of the main car and maps was done in much higher quality than I initially expected. The menus and user interface (UI) navigation is smooth and smart looking providing the best user experience (UX). I also managed to design a high-scoring system, as well as functions for saving and loading different aspects of the game. Overall, I think that my original statement of intent was limited by what I thought I could do whereas the final product was much better than originally planned.

## RESEARCH

The large amount of research I completed for topics such as game genres, modelling software's, game engines and computing languages educated me on how to perform and produce many complex aspects of my major work that previously I may have struggled with. Even with my previous knowledge on the topics my research allowed me to ensure my major work was completed to the highest standard.

## PLANNING

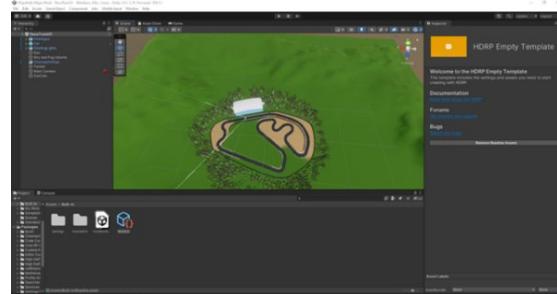
Despite my extensive planning beforehand including time plan, finance plan and concept sketches, I still ran into some difficulties when trying to meet those targets. A big factor of these delays was me not having a proper understanding of the requirements to reach each goal but thanks to early starts I was able to complete all aspects. The majority of my project always relied on me being able to do work at home on my own pc due to the computers at school not being powerful enough, however this year at the height of my productivity on the project the GPU in my computer suffered from water damage temporarily halting production and permanently slowing it down. The power of my graphics card was a key point that was allowing me to complete GPU intensive tasks with relative ease. Due to the downgrade in graphics card, I wasn't able to render the opening scenes animation at the quality I would like instead having to half the quality of the animation. Another limitation of my GPU dying was that my computers screens died and left me with only one which also reduced my efficiency when researching various factors of my project.

## CONSTRUCTION

The construction of my major work was a challenging process and required me overcoming several difficulties however thanks to my testing and my previous years' work I was able to work out how to solve all the problems I came across. Thanks to my skills I was able to finish my project at a very high standard which I am extremely pleased with, although I do wish I was able to focus on coding more instead of spending so much time modelling the assets for my game as I feel that would have allowed me to add many more features that would make the game even more fun to play.

## RACETRACKS

Throughout my project I was able to model a few racetracks each with their own challenges and aspects that made the race challenging. However, the first track I modelled, interlagos was by far the most challenging to model and most enjoyable to play with the interlagos track is inspired by the real f1 interlagos track however I customised the scenery so it was less intensive on the computer so the player would be able to have more fun when playing.



## CARS

The main car in my game was the tesla model which is seen on the front home screen when you open my game. This was because of my original idea for my major work which was going to be an animation of a tesla driving through a city but when I realised that I could make a racing game with the car I immediately decided to do that. The model itself was a challenging process of adjusting hundreds of vertices on a 3D plane but in the end, I am extremely satisfied with the end product and even happier that I was able to get a bit of my animated intro into my game on the boot up screen.



## IN-GAME USER INTERFACE

The user interface in my game is exactly how I had hoped it would turn out I have every feature of it that I wanted on display on display, and it is all perfect and easy to read the hardest part by far was figuring out how to set up the timers that tell you how long the race has been going and then further breaking it down into lap time and best time. The next challenging part which I was extremely pleased with was the results and leaderboard that come up when you would finish the race the blend of code and design was a very fun process although it times repetitive when I needed multiple vary similar variables.



## TITLE SCREEN

The design of my title screen is very clean and nice it is just an animation of my games main car spinning around while text that reads “press enter to start” flashes along with some music that suits the style of the game.



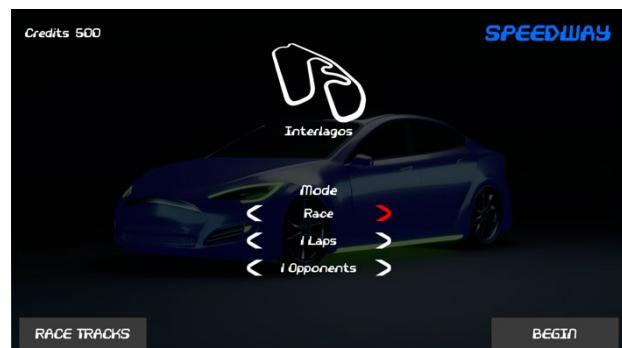
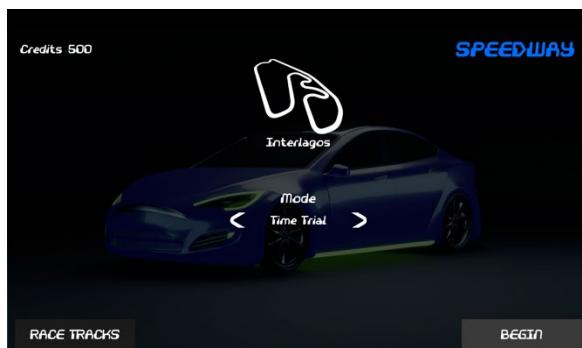
## MAIN MENU

The design of my main menu utilises a very simple design that gives the player the option of which map they would like to play as well as the option to change car and look at their stats. Whilst displaying the logo, the credits the player has earned with a background of the car modelled on blender that has a filter over it, so the image is slightly darker.



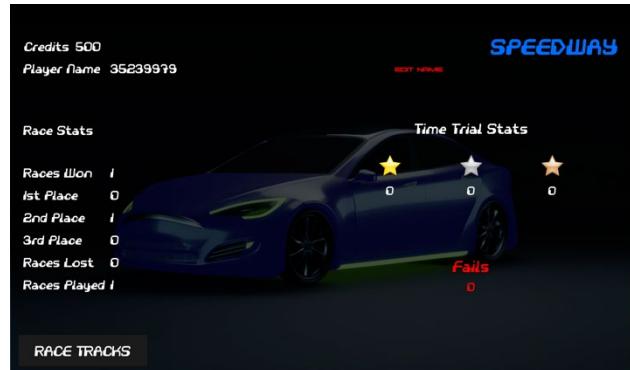
## RACETRACK MENU

The racetrack menus are used to allow you to change the race mode (time trial or race), and then once you have decided to further alter the conditions changing values such as the number of opponents and the amount of laps you want to race for.



## STATS MENU

The stats menu has a very simple purpose of being the place where you can look at how you have previously performed in races showing stats such as 1<sup>st</sup> 2<sup>nd</sup> and 3<sup>rd</sup> places, races lost (which is anytime you were 4<sup>th</sup> or lower), races won (anytime you get 3<sup>rd</sup> or higher), the races played, golds earned, silvers earned, bronzes earned and fails (anytime you took longer than the allocated time for a bronze on a time trial)



## CAR SHOWROOM MENU

I am very satisfied with the ability of my car showroom. The ability to cleanly switch between the three cars was a feature I wanted to work extremely smoothly and am really satisfied with.



## MUSIC

Across all of my menu scenes in game I have put songs to provide a better user experience when getting ready to start I am very pleased with this feature overall. The songs I used were punch deck conflicting ideas and cj beards don't have money.

## HANDLING

The handling of the cars in game is something I am very happy with as it took me many days to set up the game was always meant to be played with a controller however at the start of the testing, I didn't have access to a controller to know how the handling converted however once I got access to a DualShock controller I was able to tweak the values, so the handling suited the game perfectly.

## MENU NAVIGATION

The menu navigation in my game took a bit of planning but once set up worked incredibly smoothly the difficulty I faced came from setting up the controller and the navigation between buttons on the screen however I was able to reconfigure the buttons manually so the controller could navigate between them accurately when you flick the joystick up down or sideways.

## OVERALL

Overall, I think I have done a really good job with my major work and have managed to create a high-quality game that myself and others are able to enjoy. There are very limited areas which could have been improved this is why I believe my major work is overall very high-quality.