|  |
| --- |
| Final Year Project |
| Game Design Document |
| P2672054 |

|  |
| --- |
| Eimantas Stoskus |

Contents

[Introduction 2](#_Toc801628122)

[Literature Review 2](#_Toc727032270)

[Overview 2](#_Toc1962308610)

[Motivation 2](#_Toc1226129244)

[Research 2](#_Toc1926862673)

[The physics behind cars 3](#_Toc727164662)

[Similar games to the project 3](#_Toc1311841845)

[Implementation of Drifting 3](#_Toc1160511330)

[Opponent AI 4](#_Toc1732066413)

[Game Story Summary 4](#_Toc801984237)

[Main Gameplay 4](#_Toc1210219187)

[Versus Mode 4](#_Toc1973440773)

[Time Attack 4](#_Toc1970324847)

[Game World 4](#_Toc1629491330)

[Game Experience 5](#_Toc2014811675)

[Gameplay mechanics 5](#_Toc475476308)

[Drifting 5](#_Toc861495233)

[Timer 5](#_Toc1635245824)

[Opponent 5](#_Toc1462929336)

[List of Features 7](#_Toc1975594454)

[Controls 7](#_Toc329086035)

[AI 7](#_Toc447270204)

[Level Design 7](#_Toc1969895226)

[Physics Simulation 7](#_Toc391085426)

[Game Engine 8](#_Toc1911515735)

[Audio and Visuals 8](#_Toc1390297876)

[Visuals 8](#_Toc1991532285)

[Menus 8](#_Toc592234532)

[Audio 8](#_Toc223635089)

[Hardware/Software requirements 8](#_Toc2110530120)

[References: 8](#_Toc463510823)

# Introduction

The proposed project is a semi-realistic racing game based on a popular show called Initial D, with a focus on simulating accurate car physics. This project was proposed after the recent release of the Forza game and the reception it received. After some research, there is a clear demand for games in the racing genre. Due to the recent release of the show MF Ghost, which has received adequate reviews and is a successor to the show Initial D, the demand for a game inspired by the show will increase as the popularity in the original show resurges. This will increase the number of users to search for games that recreate their favourite show and allow them to experience it themselves.

The environment was chosen to be based around mountains, due to a demand for racing games to be based around mountain racing, but a lack of games to fill the demand. Initial research showed that the last game with good reception was Need for Speed: Carbon, which was based in canyons and was released in 2006. There are also the Initial D arcade games released, which were exclusive to Japan, and the fan-made games, which have poor reviews but with most reviews stating they like the idea and concept.

The game will be a single track around a mountain. The player will have two options to race the mountain track. The first option is that the player will be able to race an opponent on and compete in a 1 vs 1 showdown scenario. The second option is for the player to practice the mountain track on their own and focus on achieving and improving their best personal time.

The level layout will be themed around a regular mountain road, filled with steep slopes and cliffs around the road to give the player a feeling of being on a steep mountain filled with various dangers that only require a single mistake to cause critical failure. The road itself will also be filled with road signs and regular road markings to remind the player that they are on a normal road and not a racetrack, creating the sense of an illegal street race, and immersing the player into the surroundings.

The vehicles will be modelled at a lower polygon count and will use lower resolution textures. This will lower the system requirements needed to run the game and allow a broader audience to enjoy the game to its fullest. The use of lower resolution models and textures will also recreate the feeling of the original Initial D arcade games that are used as an inspiration for the project. The vehicles will also be modelled after cars in the street racing car scene that were, and still are, popular at the time. This will allow players to drive some of their favourite vehicles and further enhance their experience by immersing them into the environment.

# Literature Review

## Overview

The aim of the project is to create a realistic car racing game that simulates the engine and gears of a car. The literature review will investigate similar games of the genre, the physics required to implement a car and other features required to fit the theme of the project.

## Motivation

The motivation for this project is to gain experience in working with physics-based racing games, due to the recent popularity of such games, with games such as Forza Motorsport and Need for Speed. The theme for the game was chosen due to the recent release of the show MF Ghost, which has revived popularity for the predecessor of the show, Initial D, and increased the demand for games based in a similar environment.

## Research

### The physics behind cars

Cars use the chemical reaction that occurs inside and engine to move the vehicle. This is achieved by converting the reciprocating movement from the pistons, into rotational movement at the wheels. Due to the high speeds that the engine rotates, a gearbox is used to reduce the engine speed and transmit power to the drive wheels more efficiently. The clutch is used to switch between gears by separating the engine from the drivetrain. As long as the torque transmitted to the wheels does not exceed the maximum friction limit of the wheels, the vehicle will accelerate forward or backwards in the intended direction. If the torque transmitted exceeds the friction limit of the wheels, the wheels will begin to slip, which will result in the transmitted power to be wasted, with the car accelerating slowly until the wheels are able to grip the road surface properly.

Games usually use calculations to simulate the engine. Using the equation below, we can simulate the torque produced by an engine to move the vehicle:

Gear Reducer Torque = 9550 \* Motor power / Rated Power motor’s input revolution\*Ratio\* gear reducer efficiency

It also allows the simulation of a gearbox by changing the gear ratio as different gears are selected.

### Similar games to the project

As the project is inspired by the show Initial D, the first step was to look Initial D game releases. The games were well reviewed, but their success was limited to Japan, as players in Europe did not like the idea of being last place and instead prefer more opponents and more open roads.

Another game to research was Beam.NG, as the game is known for the high realism in its physics and car simulation. Beam.NG uses physics equations in real-time to calculate the car speed and other physics-based values. The vehicle bodies are simulated using soft-body node-beams and invisible skeletons to simulate the flex and deformation in the vehicles.

Another game to research was Need for Speed Carbon, which had a map based between canyons and had some focus on drifting. Overall, the game was received well and brought a nice change of pace as it replaced the drag racing from previous need for speed games at the time and focused on a unique style of racing.

### Implementation of Drifting

There are multiple ways to implement drifting in games. A simple way of implementing drifting is to have a dedicated drift button and having two separate vehicle controller scripts to handle the car. One script will be used to drive the car normally, while the other script will be used to control the car whilst drift is initiated. This is usually used in arcade styled games such as Mario Kart and most racing game arcade machines.

Another implementation of drifting is by having controlled slip in the wheels. This is achieved by having the rear of the vehicle have a controlled loss of grip which allows the car to initiate drifting by either using the hand brake or by turning sharply in either direction. This can be seen in most games such as the Need for Speed series and the Forza Horizon series.

A third implementation of drifting is achieved by using the physics of the vehicle. This is achieved by having all the required components to drift a car simulated and by having the vehicle interact with the road surface realistically. This allows the player to drift the vehicle by using real life drift theory and gives the game a more immersive experience. An example of this implementation can be seen in games such as Assetto Corsa and Beam.NG where the vehicles are simulated to a high degree of realism.

### Opponent AI

There are different methods to consider when creating an AI for a racing game. One of the simplest ways is to have the AI drive along a predefined racing line, or a spline. Other components, such as cornering speed, acceleration, and braking, can be emulated using formulas and basic collision detection. This was used in most early racing games due to the simple implementation and is still used in some track-based games, with the implementation of more advanced AI, allowing the opponents to dynamically adjust their skill level and adapt to different elements of the race, such as track conditions and player driving skills. This can be seen in games such as Forza Motorsport and Horizon series.

Another example of ai in racing games is using a mix of pathfinding and a tactical AI. This is mostly used in open world games, where the AI needs to have the ability to make decisions. Pathfinding is used to determine what path the ai needs to take based on the destination, such as if the destination is stationary or if its moving. The tactical AI is involved in the immediate decision-making process. This includes actions such as creating roadblocks in an attempt to stop the player, or if the AI is close enough to the player, attempting to ram the plyer off course. This can be seen in open world games such as, Grand Theft Auto and Watch Dogs, where the police actively attempt to stop the player from escaping.

# Game Story Summary

The main story begins on top of an undisclosed mountain on a full moon night. The main character and opponent are lined up at the top of the mountain, preparing for the big race. The opponent showed up in town a couple of days ago claiming to be the best racer there is, with an undefeated win record. The main character took it up to himself to put this outsider in his place and finally beat his undefeated win streak. Will he be able to win the race and show his arrogant opponent who is truly the best, or will he lose and become a statistic and a part of the countless other drivers who challenged the outsider’s undefeated win streak. Only this race will tell.

# Main Gameplay

The game will consist of three different modes for the player to choose from based on the experience they want.

## Versus Mode

Versus Mode will be the main mode of the game. This will consist of the player in a 1 vs 1 showdown against an opponent and will be the main mode of the game. The player will be facing against the arrogant outsider who showed up to their town a couple of days ago and claimed to be the best of the best, showing off their undefeated win streak.

## Time Attack

Time Attack will be a time trial type of game mode to allow the player to have practice laps without having to worry about the opponent. This allows the player to practice the mountain track on their own and focus on achieving and improving their best personal time.

# Game World

The game world will be a single level based in a mountainous environment. The track will consist of multiple curves and turns to create a feel of driving around a mountain side. The side of the track will also be surrounded by steep cliffs and trees to enhance the feel of being in a mountainous area.

# Game Experience

The game is inspired by Initial D and is themed around racing around mountains. This will have the player partake in a 1 vs 1 showdown whilst racing either up or down the curvy roads and turns of the mountain, all whilst needing to be aware of the looming danger of the cliffs of the mountain. The models and textures in the game will also be a lower resolution, to recreate the classic feel of the arcade games that the original Initial D games originate from.

# Gameplay mechanics

## Drifting

Since the components of the vehicle will be simulated using real life statistics and will have a rear-wheel drivetrain, the player should be able to initiate a drift using realistic theories and be able to predict how the car will react.

## Timer

The game will include a timer, which will allow the player to keep track of their fastest time and compete against it to improve it. It will also allow players to compare their fastest times with each other.

# Opponent

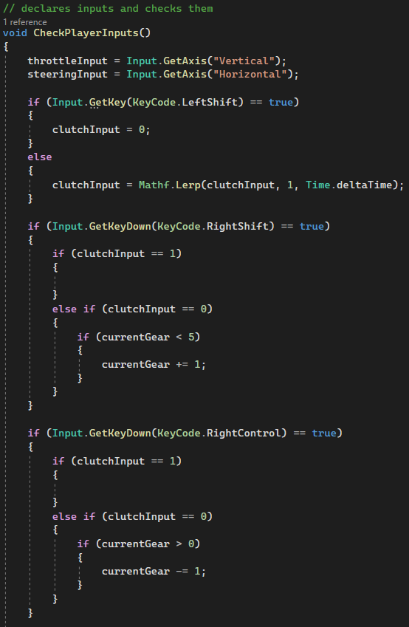
The opponent will consist of a separate vehicle that will race the player side by side.

Technical Design Document

# List of Features

## Controls

The player will be able to choose between having the car in automatic or manual mode. In automatic mode, the controller will manage changing gears and the player will just need to accelerate or use the brakes, whilst driving. In manual mode, the player must manage all aspects of the vehicle, such as changing gears, managing the clutch, acceleration, braking and driving.

The control scheme will also be available for several types of inputs, such as mouse and keyboard, and controllers. The controls will be something similar to the control scheme bellow:

* **Steering** – A and D / Left and Right on the Left Thumb stick.
* **Acceleration** – W / Right Trigger.
* **Braking** – S / Left Trigger
* **Clutch** – Left Shift / Left Bumper
* **Gear Shift Up** – Arrow Key Up / Y or Triangle
* **Gear Shift Down** – Arrow Key Down / B or X

The initial approach was to create a controller script which detects keyboard input only and focus on the implementation of vehicle features such as driving and shifting gears before implementing support for multiple control schemes.

A computer screen with white text

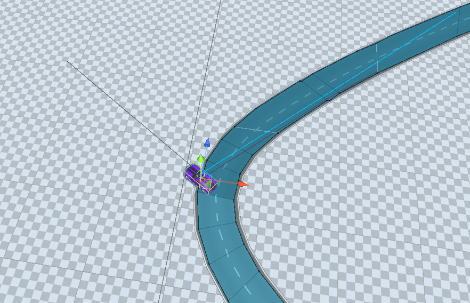
Description automatically generatedAfter implementing all of the controls, the next step was to implement multiple control schemes for controller support. I decided to approach this by using unity’s new input system.

A screen shot of a computer code

Description automatically generatedA screen shot of a computer program

Description automatically generated

## AI

The AI of the opponent will follow a set of checkpoints placed around the track. The ai will also adjust its difficulty by adjusting its speed based on the player's position. If the player’s vehicle is too far the ai will slightly slow down to allow the player to catch up without making it too obvious to the player. On the other hand, if the player’s vehicle gets too far, the ai will begin to race more aggressively to try and catch up to the player. This will only affect the speed and acceleration of the ai, and not the values of the vehicle the ai is driving, as the game focuses on realism.

The first attempt at implementing an AI was using NavMesh pathfinding. This was implemented by baking the NavMesh suface on the road pieces. This looked promising at first, but after tweaking and testing the parameters multiple times, the conclusion was reached that using a NavMesh will not work as it could not be set up to recreate a vehicle.

The next step is to use A\* Pathfinding to create an ai that will create a racing line as the ai drives through the track.

## Level Design

Since the theme of the project is based on Initial D, the level design will focus on creating a mountain-based environment. This will be achieved by having steep slopes and cliffs along the player driven track. The track will also contain other intricate details, such as road markings and signs, to make the track feel more like a normal road that people would encounter every day. This will give the level the feel of an illegal street race and create extra immersion for the player.

The roads will be designed using EasyRoads 3D. This allows me to create and change the layout with relative ease.

## Physics Simulation

The key focus area of the project is to have a car with simulated physics, meaning extensive research and effort to make the vehicle interact with the rest of the world in a realistic manner. This can be achieved by using real life statistics of vehicle engines and gearbox ratios, as values for the torque equation bellow:

Gear Reducer Torque = 9550 \* Motor Power / Rated Power Motor’s input rev \* Ratio \* gear reducer efficiency

A separate function can be used to change between the different gear ratios as the player changes gears. The torque can then be sent, as a rotational force, to the wheels which will be attached to the vehicle model, resulting in the car accelerating forward. A clutch can also be implemented to decrease gradually, which will give the player a smoother and more realistic experience and allow the gear changes to be smoother.

The first step was to synchronize the wheel meshes with the wheel colliders as the vehicle model used had them separated. This is done using a function which takes the world position and rotation of the collider and applies it to the mesh.

The next step was to simulate the engine. The first step was to calculate the vehicle’s current rpm using the wheel colliders. This is achieved by calculating the average and multiplying it by the current gear and differential ratios. The Lerp function is then used to smooth the acceleration between the current rpm and the expected rpm based on user input. Finally, the rpm is converted to torque by using an animation curve to represent the BHP curve for the vehicle. This is achieved by calculating the current percentage of the BHP at current rpm. The percentage is then multiplied by vehicle BHP and divided by current rpm to produce the IHP (indicated horsepower) of the vehicle.

A computer screen shot of a program

Description automatically generatedA computer code on a black background

Description automatically generatedThe IHP is then multiplied by the current gear and differential ratio to give the torque. The torque is then multiplied by a factor of 5252 to convert it Nm which is used by Unity’s system, and clutchInput which changes between 0 and 1.

# Game Engine

The choice of game engine for this project was Unity. This was due to the large amount of support, tutorials, and documentation available for the engine. It is also very accessible and the engine I am most experienced in using. Using Unreal Engine was also an option, but due to the lack of documentation and unreal engine’s focus on blueprints made unreal not an option.

# Audio and Visuals

## Visuals

For the overall look of the project, the game will mostly have textures and models at a lower resolution. This will allow the game to reach a broader audience as it will lower the hardware and software requirements to run the game.

## Menus

The game will only be available as a PC version, which means all the menus and UI elements will be focused and optimised to mainly work with PC only.

## Audio

The audio of the game will consist of real-life samples of the engine noise that the vehicles will be based on. Other sound effects, such as tyre screeching, and ambient noise will be taken from real life examples.

# Hardware/Software requirements

This game will be released on PC only due to the time constraints of the project. Due to the decision to have lower resolution textures and models, the game should be able to run on as many computers as possible, allowing for more players to potentially enjoy the game.

# References:

Geraltofrivia (2013) How do you drift in racing games?, Forum. Available at: https://hardforum.com/threads/how-do-you-drift-in-racing-games.1764047/ (Accessed: 30 November 2023).

How A car works (no date) How a Car Works - Guides to car mechanics and automotive engineering. Available at: https://www.howacarworks.com/ (Accessed: 24 November 2023).

Mohamed, A. (no date) Artificial Intelligence in Racing Games - University of Birmingham. Available at: https://www.cs.bham.ac.uk/~ddp/AIP/RacingGames.pdf (Accessed: 01 December 2023).

Monster, M. (2003) Car Physics for Games, Car physics. Available at: https://asawicki.info/Mirror/Car%20Physics%20for%20Games/Car%20Physics%20for%20Games.html (Accessed: 26 November 2023).

Robinson, M. (2014) Sega’s arcade redemption, Eurogamer.net. Available at: https://www.eurogamer.net/segas-arcade-redemption (Accessed: 02 November 2023).

Team, S.T. (2021) Gearbox output torque calculation formula - SDT transmission, Industrial gear reducer,gear motor manufacturer - SDT Transmission. Available at: https://www.sdtgear.com/how-to-calculate-gearbox-output-torque-gearbox-output-torque-calculation-formula/ (Accessed: 24 November 2023).

Tomlinson, S. and Melder, N. (no date) An architecture overview for AI in racing games - game AI Pro, An Architecture Overview for AI in Racing Games. Available at: https://www.gameaipro.com/GameAIPro/GameAIPro\_Chapter38\_An\_Architecture\_Overview\_for\_AI\_in\_Racing\_Games.pdf (Accessed: 07 December 2023).