National College of Ireland

BSc in Computing

2015/2016

Stephen Chapman

X12343316

X12343316@student.ncirl.ie

Drift City

Technical Report

Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name:
Student ID:
Supervisor:

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature:____ Date:____________

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Complete the sections above and attach it to the front of one of the copies of your assignment,

What constitutes plagiarism or cheating?

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

Penalties for Plagiarism

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- the requirement that a student to attend additional or other lectures or courses or undertake additional academic work.

Table of Contents

Executive Summary

This purpose of this report is to analyse the project which I have undertaken and to evaluate the final product. This report draws attention to the fact that the gaming industry is one of the world’s most thriving industries at this moment of time, and it shows no signs of slowing down. Knowing this it was an easy decision to develop a game with an eye for perspective profitability in mind.

My game is of the racing genre. Over recent years there has been a renaissance in the racing genre with the release of new titles such as DriveClub, Need for Speed and Forza. My game is aimed at people who enjoy cars, racing games and who also wish for a new kind of racing game.

The problem for many companies hoping to develop racing games is trying to find something new and exciting to differentiate it from the mainstream games. By using the traditional racing game features and adding a new aspect which has not been used as extensively my game oozes a breath of freeze air.

1 Introduction

1.1 Background

The gaming industry has undergone a substantial evolution since the 1970s and is no longer just a fringe activity but instead it has become the mainstream. The gaming industry accounts for €240million in revenue, annually in Ireland (O'Reilly, 2014). The process of creating a game no longer requires an abundance of employees. With today's designing tools which are freely available, anybody with an interest to create a game can do so with ease.

In 1974, the first racing game was created by Atari, named Gran Trak 10. Gran Trak 10 was a coin-operated, arcade game which saw the player drive around a twisting track and attempt to beat the clock. Most modern games take advantage of the traditional racing genres elements, such as racing against the clock, customising the car and unlocking more cars. Many of the modern racing games, like Need for Speed and Forza Horizon, allow users to fully customise their cars using the money and/or skill points they have earned throughout the game. This is fun for the player as it allows them to create a unique car so they can add their own personal touch to the game.

As racing games are one of the most popular genres of games around today I found my investigation quite interesting as to what it takes for a game of this genre to become successful. Seeing as racing games are one of the most popular genres in today's market I believe that customers would enjoy a traditional racing game.

I chose to make a racing game because I am a huge fan of racing games. It is possibly my favourite area of gaming and I believe that working on a racing game would be extremely enjoyable. Need for Speed, is perhaps the most well-known title in this genre and its 23rd title will be released in November of this year. What I love most about racing games is that you are able to drive the most amazing cars around the world and put them to the test. Usually these super cars are locked until the player reaches a certain level or completes a certain race, so this makes the user have to play through the game in order to unlock the cars of their dreams.

1.2 Aims

The main objective of my project is to create a fully functioning 3D racing game. The game will consist of features that most players would expect of a racing game. The game will allow players to race against the computer, and explore the more exciting aspect of the game in the Fun Zone. The player will be able to switch between first person and third person views with the push of the button, and reset the best overall race and lap times that are being saved in the game.

The most unique feature of my game is the Fun Zone, here the player can partake in a different racing scenario to what is expected from most racing games. The main objective of this Fun Zone is to offer the player an alternative racing experience, one where they can just relax and enjoy themselves. Here the players will fly over ramps, through rings of fire, take on huge loop-the-loops, and other obstacles they may face. I think game modes like this should become a fundamental part in the racing game genre. This will act as an incentive for the player to continue playing the game.

Each level of the game will have pre-set times each representing gold, silver and bronze medals. The player will receive the medal which's pre-set time they have beaten. This will act as an added incentive for the player to complete the race as fast as possible.

1.3 Technologies

The technology that I will be using is Unreal Engine 4. Unreal Engine 4 is a complete suite of game development tools made by game developers, for anybody with an interest in game development to use. These tools are very easy to use and enable me to create my own virtual world with very little hassle. Unreal Engine 4 has a number of free starter packs on its marketplace which are very helpful when you begin the development phase. Unreal Engine technology powers hundreds of games as well as real-time 3D films, training simulations, visualizations and more.

Over the past 15 years, thousands of individuals and teams and have built careers and companies around skills developed using the engine. ("UnrealEngine.com," 2016)

1.4 Challenges

There were a number of challenges that I faced while trying to develop this project. The main challenge that I came across was trying to make my game different from every other racing game. To do this I decided exploit areas of racing games which aren't as extensively explored as others. The way in which I overcame this challenge was the creation of the Fun Zone. Here the players no longer need to worry about the typical racing game features such as competing against other cars. Instead they can relax and focus on the fun side of the game.

Enabling the player to respawn the vehicle on demand, rather than solely when the player was flipped for more than 4 seconds. This led to some problems in the development stage, as when the player was respawning there was more than one car appearing onto the track. After some time, I managed to figure out what was cause this problem. The player pawn was not being destroyed when I was trying to call in another pawn. To get passed this I created a DestroyActor node followed by a SpawnVehicle node, and set the respawn location to the last activated checkpoint. This enabled me to overcome this problem and now the respawn function works perfectly on demand.

2 System

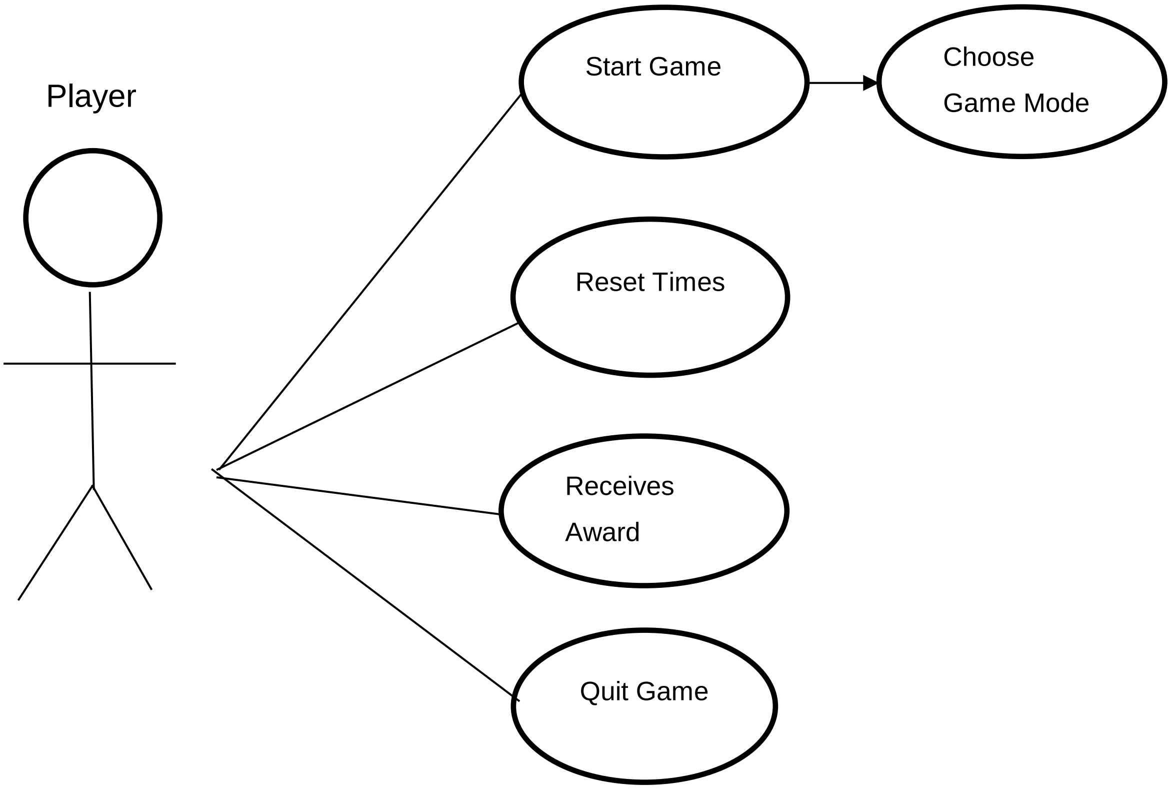
2.1 Requirements

2.1.1 Functional requirements

The functional requirements define what is needed for the system to function. This is the list of the functional requirements in my game in ranked order:

1. Start Game requirement
2. Choose Game Mode requirement
3. Reset Times requirement
4. Receive Award
5. Quit requirement

2.1.2 Use Case Diagram



2.1.3 Requirement 1: Start Game

2.1.3.1 Description & Priority

The player starts the game

2.1.3.2 Use Case

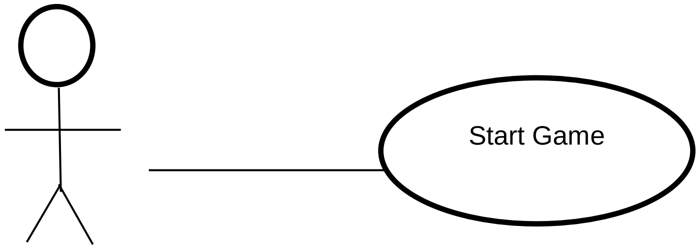
Scope

The scope of this use case is to allow the user to start the game

Description

This use case describes the process of starting the game

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode.

Activation

This use case starts when a player starts the game

Main flow

1. The system identifies the player
2. The player starts the game.
3. The system loads the main menu.

Termination

The system presents the main menu to the player

Post condition

The system goes into a wait state

2.1.4 Requirement 2: Choose Game Mode

2.1.4.1 Description & Priority

The player chooses a game mode

2.1.4.2 Use Case

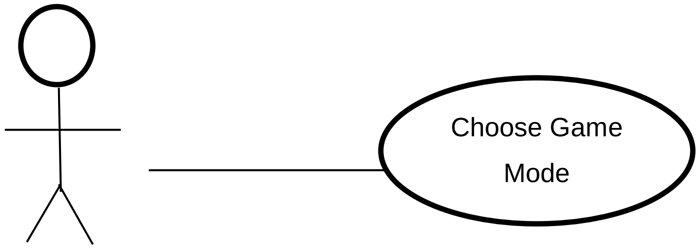
Scope

The scope of this use case is to allow the user to choose a game mode

Description

This use case describes the process of choosing a game mode

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode.

Activation

This use case starts when a player chooses a game mode

Main flow

4. The system identifies the player
5. The player chooses a game mode.
6. The system loads the game mode.

Termination

The system presents the chosen mode to the player

Post condition

The system goes into a wait state

2.1.5 Requirement 3: Reset Times

2.1.5.1 Description & Priority

The player resets the best overall race and lap times.

2.1.5.2 Use Case

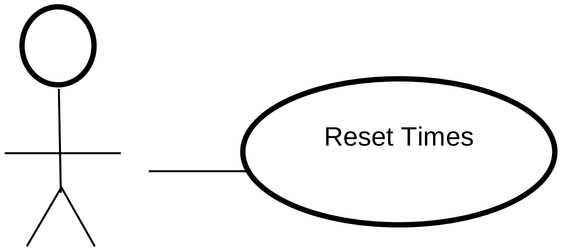
Scope

The scope of this use case is to allow the player to reset the best times.

Description

This use case describes the process of resetting the best times

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode.

Activation

This use case starts when the player resets the best times.

Main flow

7. The system identifies the player
8. The player resets the best times
9. The system the best reset times to default

Termination

The system presents the reset times to the player

Post condition

The system goes into a wait state

2.1.6 Requirement 4: Receive Award

2.1.6.1 Description & Priority

The player receives an award.

2.1.6.2 Use Case

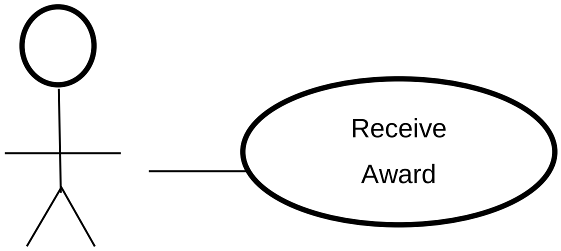
Scope

The scope of this use case is to allow the player receive an award.

Description

This use case describes the process of how the player receives an award.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode.

Activation

This use case starts when the player receives an award.

Main flow

10. The system identifies the player
11. The player receives an award
12. The system loads the award

Termination

The system presents the award to the user

Post condition

The system goes into a wait state

2.1.7 Requirement 5: Quit Game

2.1.7.1 Description & Priority

The player quits the game.

2.1.7.2 Use Case

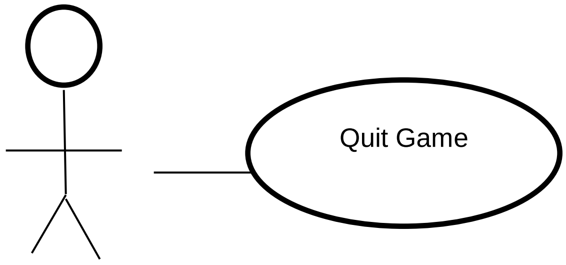
Scope

The scope of this use case is to allow the player to quit the game.

Description

This use case describes the process of how the player quits the game.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode.

Activation

This use case starts when the player quits the game.

Main flow

13. The system identifies the player

14. The player quits the game
15. The system exits the application

Termination

The system closes the application

Post condition

The system is off.

2.1.8 Data requirements

The game will require about 3.5GB of storage space from the user’s machine in order for every component used in the game to be saved and loaded correctly. Unreal Engine creates a save file using Windows registry which enables the game to set and get values related to the game. An example of this would be saving the best lap and race time and loading them again when the player returns to the game.

2.1.9 User requirements

The user will be able to play this game on any laptop or PC they wish through the use of a flash drive or another means of physical storage.

The user’s progress will be automatically saved after each race and time trial, and also after customising their vehicles. The game will also have a manual save option available to them, but this cannot be used in the middle of a race.

2.1.10 Environmental requirements

The game should run at 40 frames per second on the player’s machine. The user should manually be able to modify the graphics so that the game can run as intended. The user will also be able to disable settings such as advanced lighting and textures.

The game should be immediately responsive so that when the user initiates an action it should be performed instantly.

2.1.11 Usability requirements

The game should be available at all times to the user whenever they please. I hope to eventually add a multiplayer feature to the game by hosting it on a server. This multiplayer feature may not always be available to the user due to maintenance.

2.2 Design and Architecture

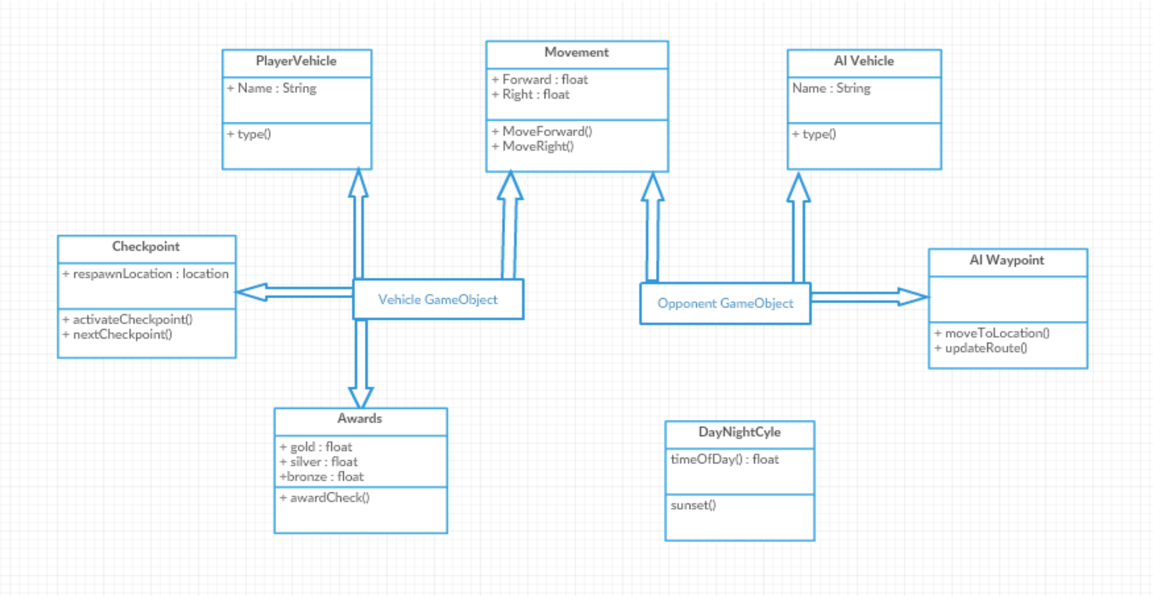


Figure 2-1

I think that this architecture is well suited to how I will create my game. The Vehicle GameObject interacts with the PlayerVehicle, Movement, Checkpoint, and Award classes. The first class the GameObject will interact with is the PlayerVehicle class. The player will then enter either a race and the next class called upon will be the movement class. This will allow the user to move in any direction. The Vehicle GameObject will also interact with the Checkpoint class. This class sets the respawn location for the player, and activates checkpoints. Finally, the Vehicle GameObject communicates with the Awards class to determine whether the player has achieved an award or not.

The Opponent GameObject interacts with the AI Vehicle, Movement and AI Waypoint classes. The first class it interacts with is the AI Vehicle class. This then communicates with the AI Waypoint class, which determines where the next checkpoint is for the AI Vehicle. Lastly, it will communicate with the movement class, which allows the car to move.

Some races will take place during the day while some will take place at night. The DayNightCycle class will determine what time of day it is and adjust the lighting accordingly.

2.3 *Implementation*

2.3.1 Checkpoint System

TriggerBoxes are placed throughout each level in my game. Once the player has passed completely through a checkpoint, the next checkpoint will spawn. The sequence node allows different code aspects to run simultaneously. There are two cars which the player can use in the game, this is why there are two pins in the sequence node.

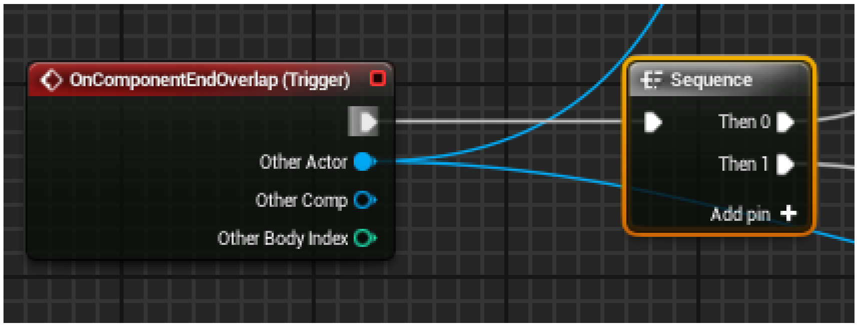


Figure 2-2

The checkpoint is cast to the vehicle’s blueprint class, so that the checkpoint will only move once it is certain that the player has passed through and not the AI car. The checkpoint determines whether the player has passed through the checkpoint in the correct direction. If they do CheckpointCleared is called from the Tracker Class and the checkpoint which the player has passed through then becomes hidden in game.

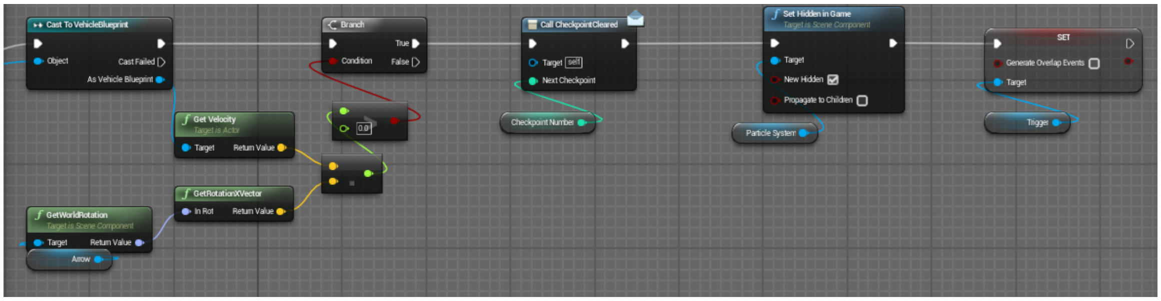


Figure 2-3

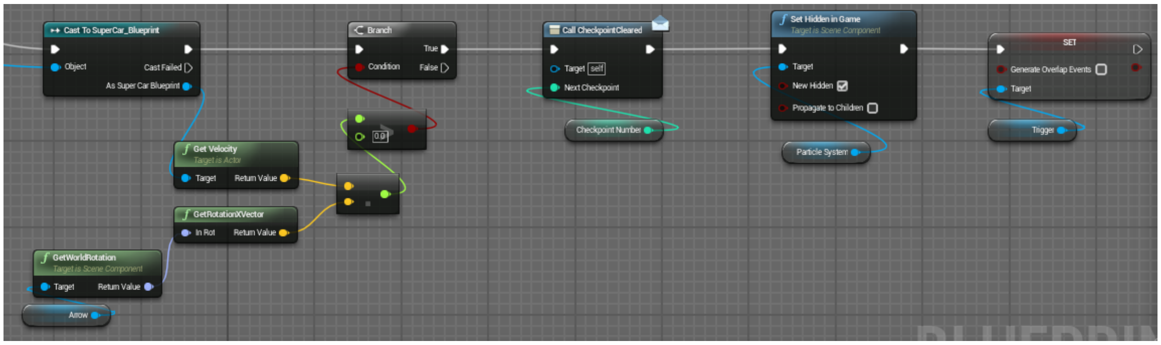


Figure 2-4

Inside the Tracker Class, the ActivateCheckpoint event makes the next checkpoint visible, as we can see in the image below, where the New Hidden checkbox on the Set Hidden in Game node is unchecked. This sets the Generate Overlap Events to true and the next checkpoint spawns, as can be seen in Figure 2-5.

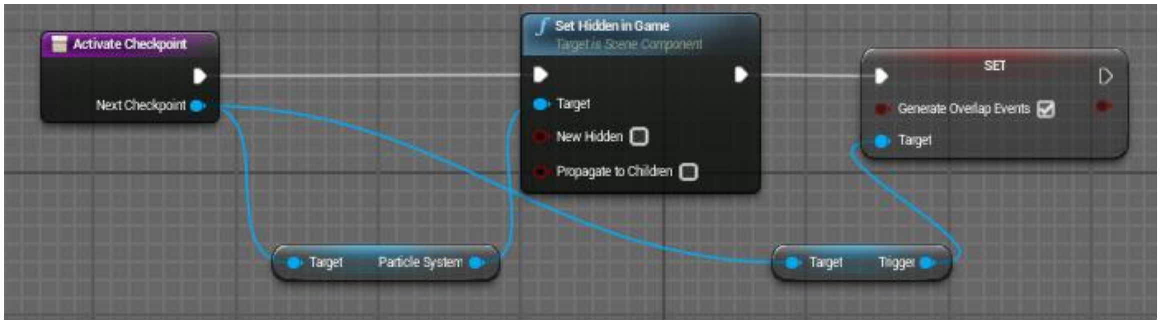


Figure 2-5

Also inside the Tracker Class is the Start Sequence custom event. This custom event gets all the checkpoints which have been set into an array, sets the array index to 1 which in turn activates the first checkpoint by calling the ActivateCheckpoint event and binds the event to the checkpoint.

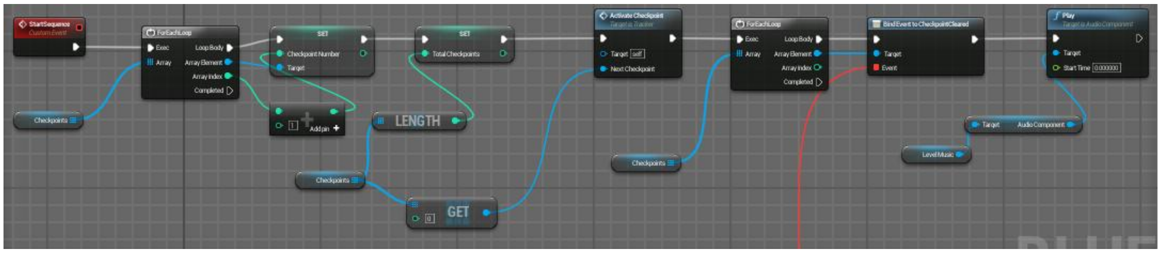


Figure 2-6

Lastly, inside the Tracker Class is the CheckpointCleared custom event. This checks to see whether the checkpoint has been cleared and if so, plays an audio snippet so the user knows they have passed through. It also checks whether the player has finished the race which can be seen in the image below, where the Race Complete variable is being called. If Race Complete is true, then the event checks to see if the player has received an award for their time and will then restart the race.

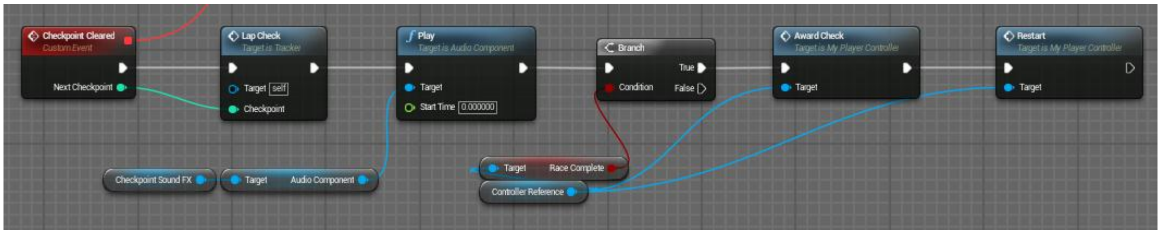


Figure 2-7

2.3.2 Lap Check

Inside the Tracker Blueprint, the Lap Check function checks to see whether the race is completed or it will update the next checkpoint. A local variable is created called LocalCheckpoint whose value is equal to the value of the current checkpoint. If this value is equal to the total number of checkpoints in the level, the function will perform a RaceCompletedCheck, if not the next checkpoint will be activated.

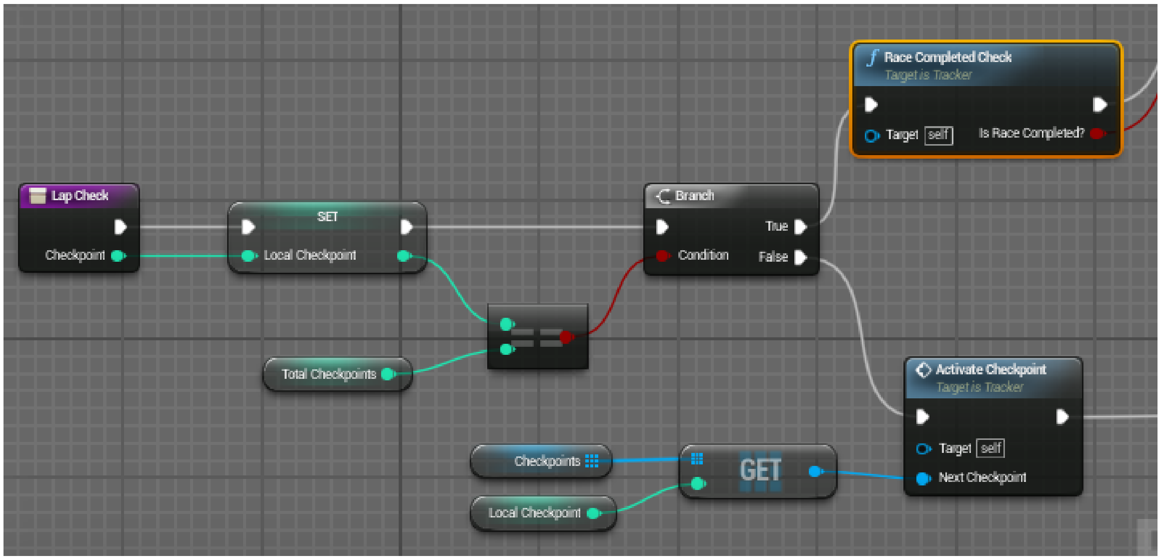


Figure 2-8

If the race has completed, it will call the UpdateLap custom event from the MyPlayerController Blueprint which checks whether the race is completed, adds to the current lap and checks for a new lap record. If the race has not completed, then the ActivateCheckpoint and the UpdateLap functions are called, and the next checkpoint is set and the current lap updates.

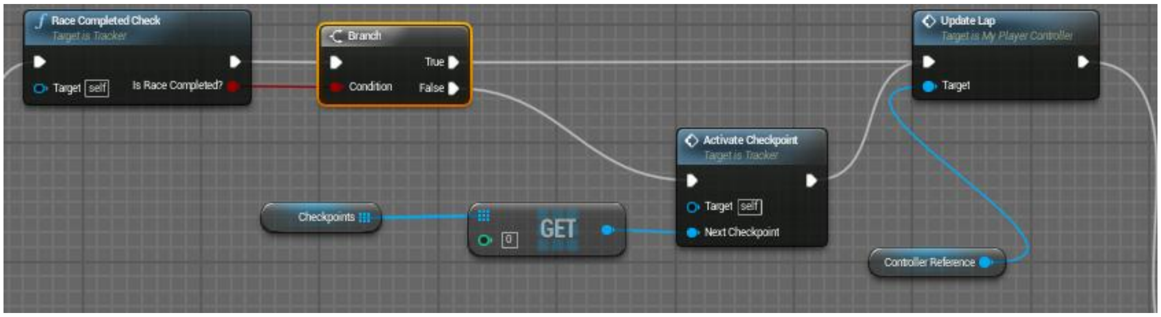


Figure 2-9

Figure 2-10 is an image of the UpdateLap custom event. This event checks whether the race is completed. If the race has not finished, the actual lap number is increased by 1 and set as the new actual lap value. It will then perform a LapTimeCheck which will determine if there is a new lap record, which will print on the screen. If the race has completed, then it no longer needs to increase the actual lap value and solely performs the LapTimeCheck.

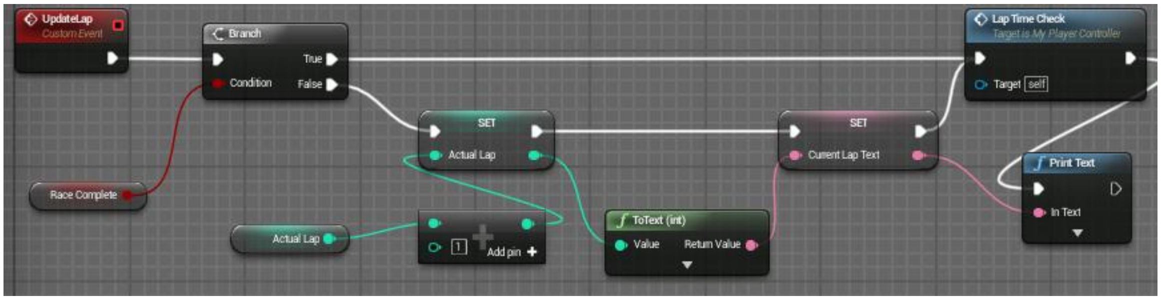


Figure 2-10

2.3.3 Lap Time Check

This is a very important function in my game. This function determines whether the player has achieved a new lap record. Firstly, the StopLapTime custom event simply stops the current lap time. After this, the current lap time is compared to the best lap time, see Figure 2-11.

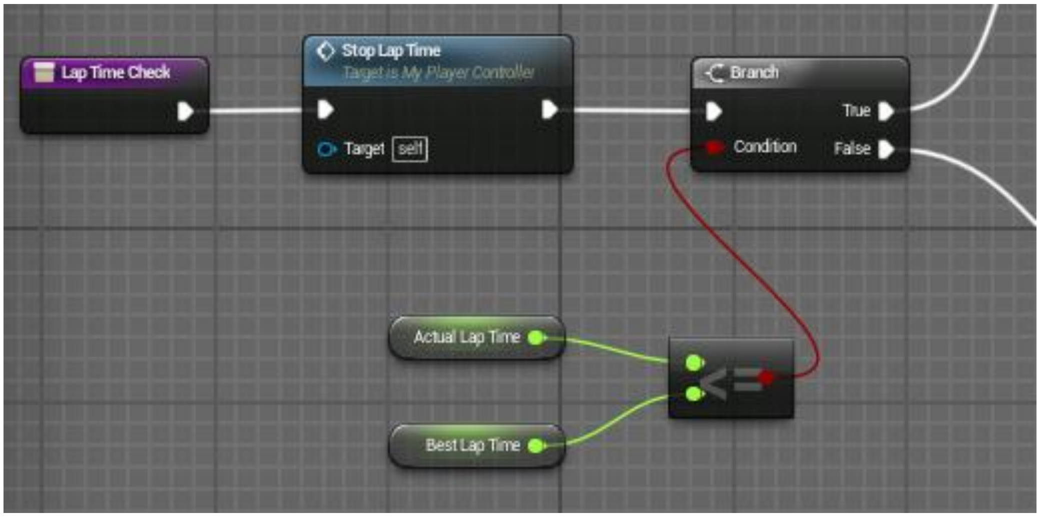


Figure 2-11

If the actual lap time is less than or equal to the best lap time, then the actual lap time is set as the best lap time. It is converted from a float variable to a text variable so that it can be displayed on the screen. The game is then saved and an animation is performed showing the user their new best lap record. The actual lap time is then reset back to 0.0.

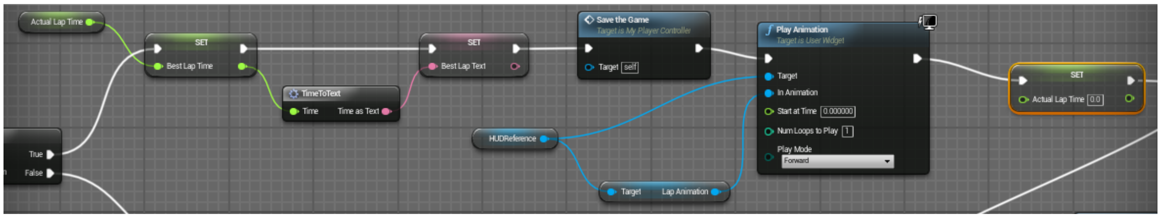


Figure 2-12

If the actual lap time is not less than or equal to the best lap time, the actual lap time is reset to 0.0.

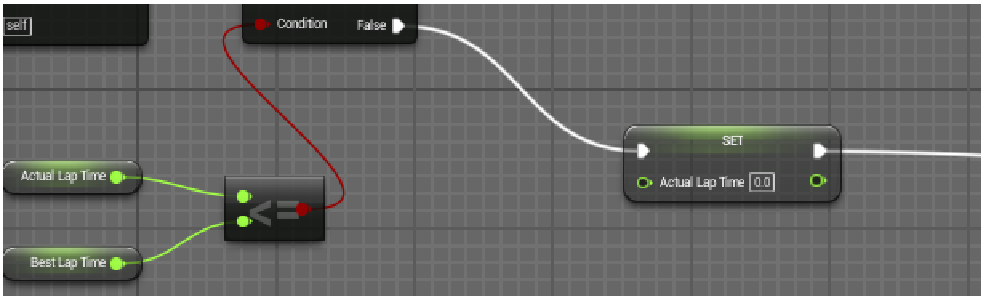


Figure 2-13

Finally, this function will check to see whether the race is completed so that it can determine whether it needs perform a RaceTimeCheck or restart the lap time.

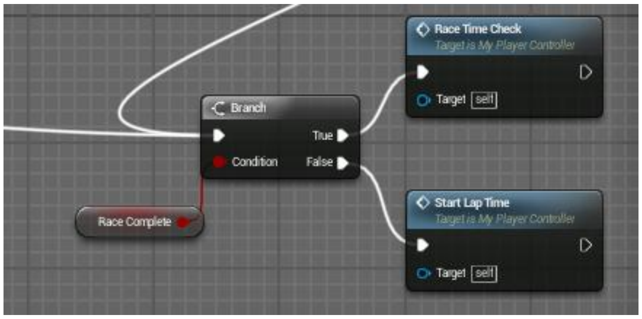


Figure 2-14

2.3.4 Race Completed Check

Inside the Tracker Blueprint, this function determines whether the race has been completed. Firstly, I created a reference the MyPlayerController so that I was able to use the variables which I defined there, such as ActualLap and MaxLaps.

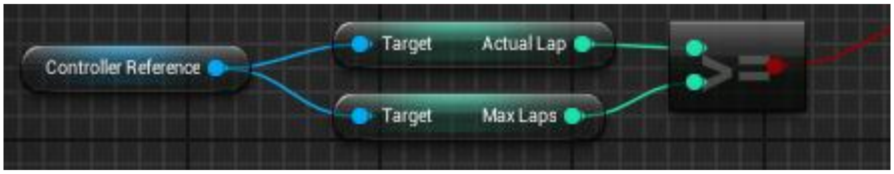


Figure 2-15

If the value of ActualLap is greater than or equal to the value of MaxLaps, then the Boolean variable RaceComplete is set to true. A local variable, LocalComplete, also of Boolean type, is then set to the same value of RaceComplete. LocalComplete sets the value on the return node to true.

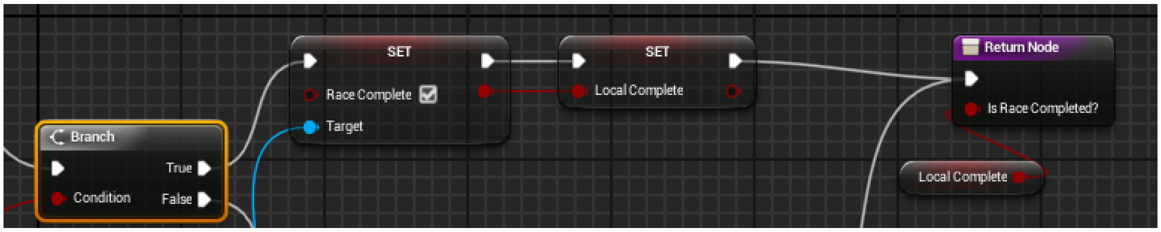


Figure 2-16

If the value of ActualLap is not greater than or equal to the value of MaxLaps, then the RaceComplete variable is set to false, which will in turn set the LocalComplete variable to false. The LocalComplete variable then sets the value on the return node to false.

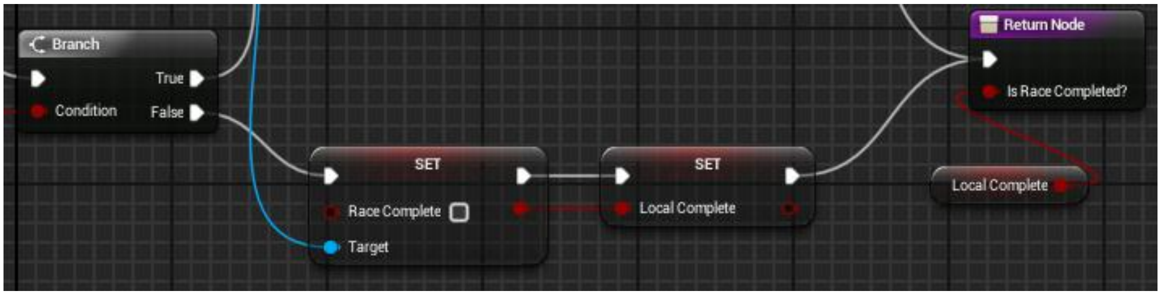


Figure 2-17

2.3.5 Race Time Check

The RaceTimeCheck function determines whether or not the player has beaten the best race time. The StopRaceTime event is called, which stops the race time. The ActualRaceTime is then compared to the BestRaceTime.

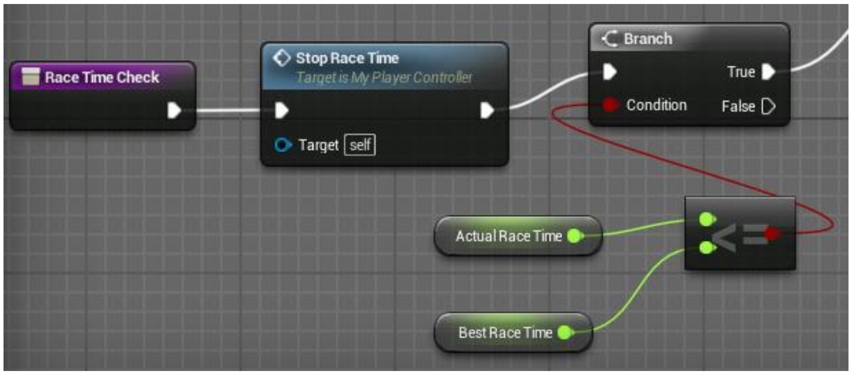


Figure 2-18

If the ActualRaceTime is not less than or equal to the BestRaceTime then it will do nothing. However, if the ActualRaceTime is less than or equal to the BestRaceTime, the ActualRaceTime is set as the new BestRaceTime. It is then converted from a float variable to a text variable so that it can be displayed on the screen. The game is then saved and an animation is performed showing the user their new best race time record. Figure 2-19 shows this.

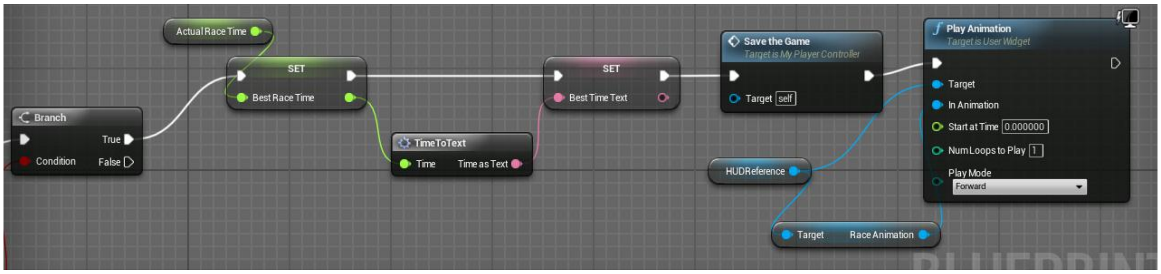


Figure 2-19

2.3.6 Respawn

The Respawn vehicle is one way in which it is possible to respawn in the game. Firstly, the player pawn is cast to the VehicleBlueprint which is the default car. The actor is then destroyed so that when the car respawns there is not multiple cars in the level. The SpawnActor VehicleBlueprint node then spawns the car at the saved respawn location, which is the last activated checkpoint. This newly spawned vehicle then possesses the default pawn, and can now be controlled by the player.

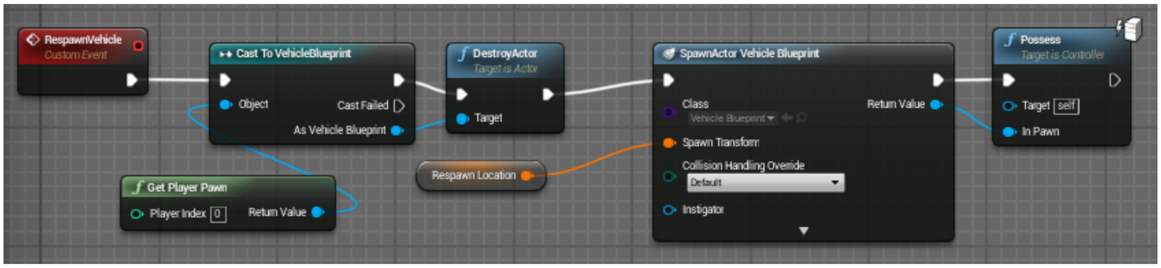


Figure 2-20

This event is called by the user when the click the P button on the keyboard, or L1 on a controller. This feature was added due to feedback that was received during testing.

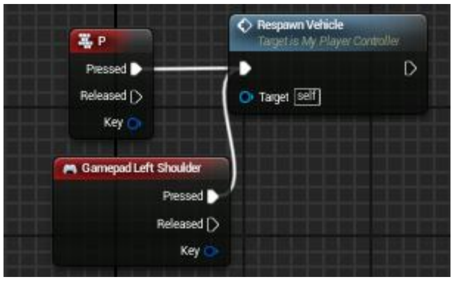


Figure 2-21

The other way in which to respawn is by flipping the car. Inside of the VehicleBlueprint class I created a function called RespawnCheck. This function checks to see if the players' car has been flipped and is no longer moving forward. If this condition is met, it determines that the car is indeed stuck and that a respawn is required. If the condition is not met, then the car is considered not to be stuck and will therefore not respawn.

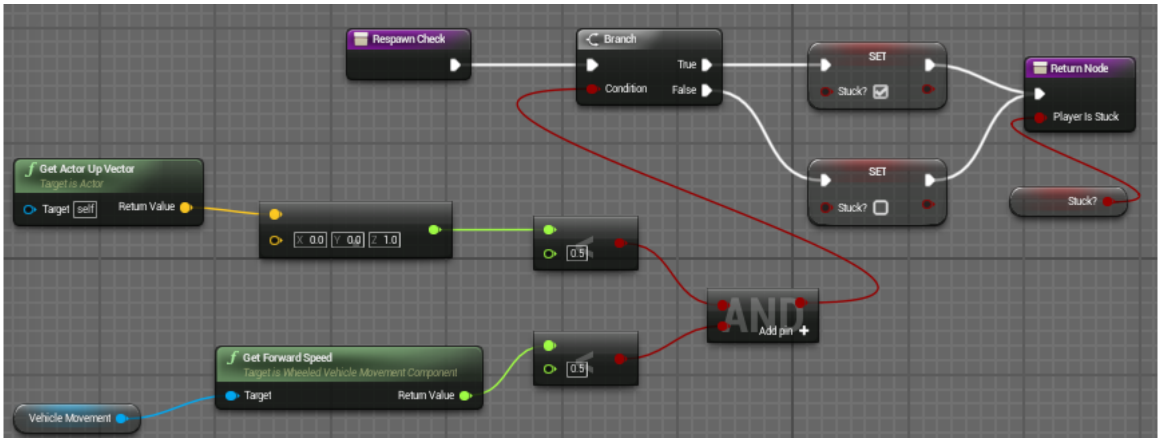


Figure 2-22

2.3.7 Reset Best Times

This feature will only appear to the user before the race has started. It allows them to reset the best race and lap times. When the R key is pressed, it will first perform a check to see if the race has not started. If this condition is true, and the race hasn’t begun, then the default best race and lap times are set to the best race and lap times. The game is then saved, and the InitText function converts the numerical values into text for the HUD. The blueprints for this can be seen in Figure 2-23

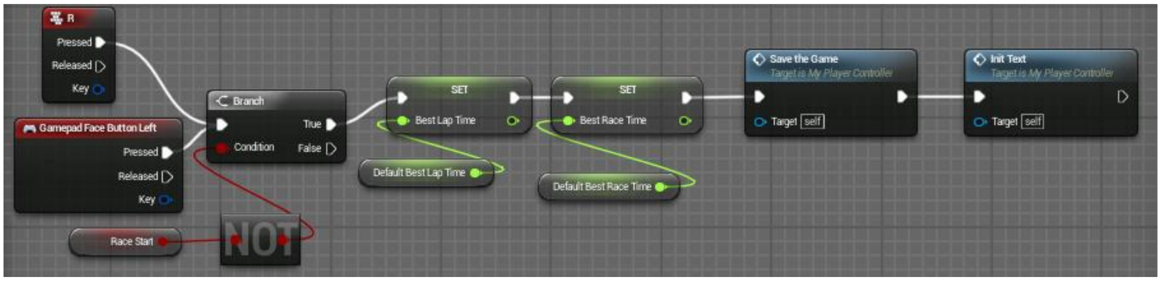


Figure 2-23

2.3.8 Save & Load System

The MySaveGame blueprint saves the best lap and race time. Any information which I need to save would be stored here. Only 2 variables needed to be created as I only plan to save the best lap and best race time.

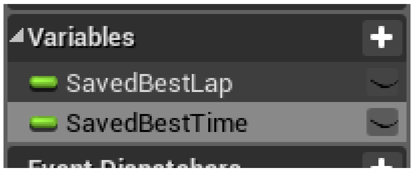


Figure 2-24

Inside the MyPlayerController blueprint there is a SaveGameCheck function. This function checks to see whether a save game exists. The Does Save Game Exist node checks for a save game. If a save game does exist, then it will load the game. If a save game does not exist, then it will set the default lap and race time as the best times and then saves the game.

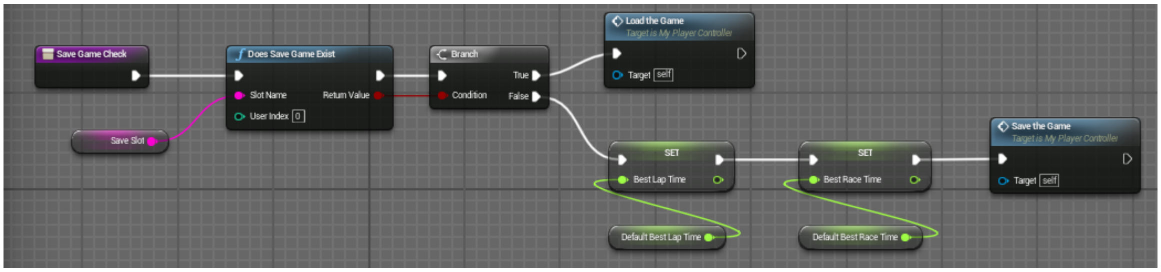


Figure 2-25

Also inside the MyPlayerController, there is a SaveTheGame function. This function saves the best lap and race time and casts them to the MySaveGame blueprint, where these values are stored. The new best lap and race times are now set as the SavedBestLap and SavedBestTime variables in the MySaveGame blueprint. The game is then saved to slot. This slot is individual to each level.

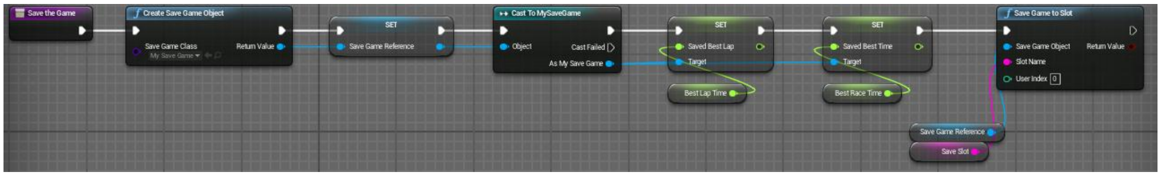


Figure 2-26

The game is then loaded from the LoadTheGame function, in the MyPlayerContoller blueprint. The game is loaded from the same slot where the game has previously been saved. This functions, gets the SavedBestLap and SavedBestTime and sets these values as the BestLapTime and BestRaceTime.

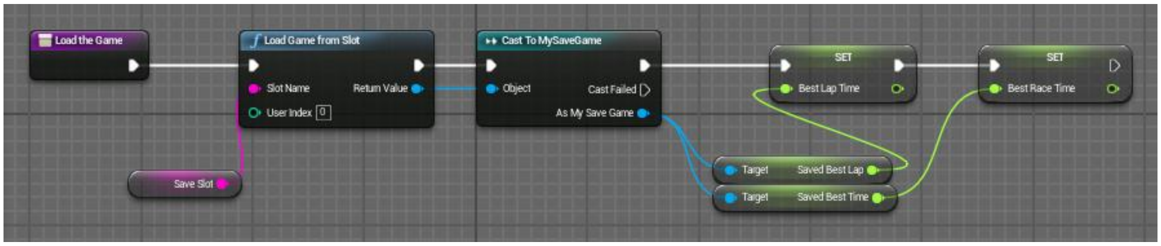


Figure 2-27

The user settings are then applied inside the ApplySettings custom event of the Tracker blueprint. When the race begins these settings are applied. Here the save slot is set to the SaveGameName which is unique for each map.

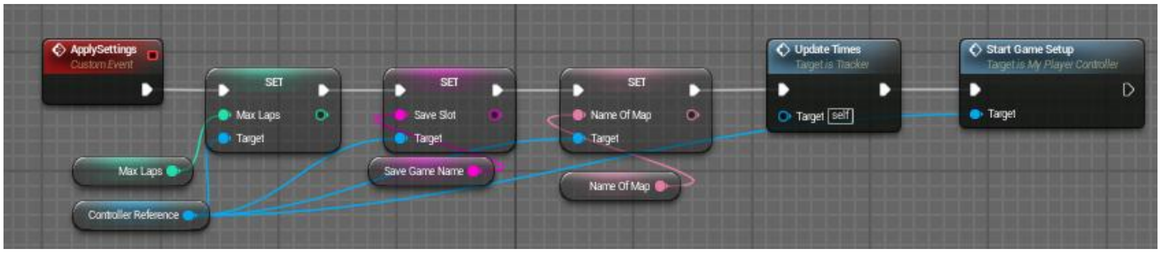


Figure 2-28

2.3.9 Award Check

The AwardCheck function compares the player's ActualRaceTime and compares it to the pre-set gold, silver and bronze times. This function begins by stopping the race time. A branch is then created, with the condition being is the ActualRaceTime less than or equal to the pre-set GoldTime.

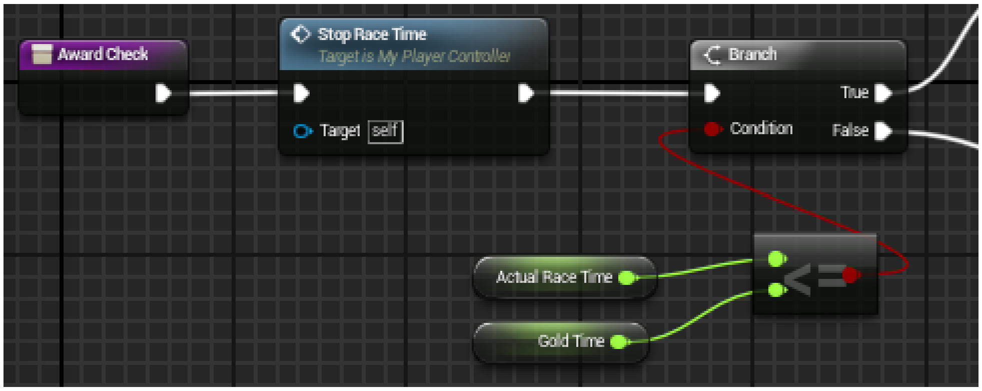


Figure 2-29

If the player's ActualRaceTime is less than or equal to the GoldTime, a reference to the HUD is created which sets the value of the GoldAward HUD to visible, so that it will appear to the player on the screen. The GameHUD is then set to hidden so that there is not too many things displaying on the screen, which is unsightly.

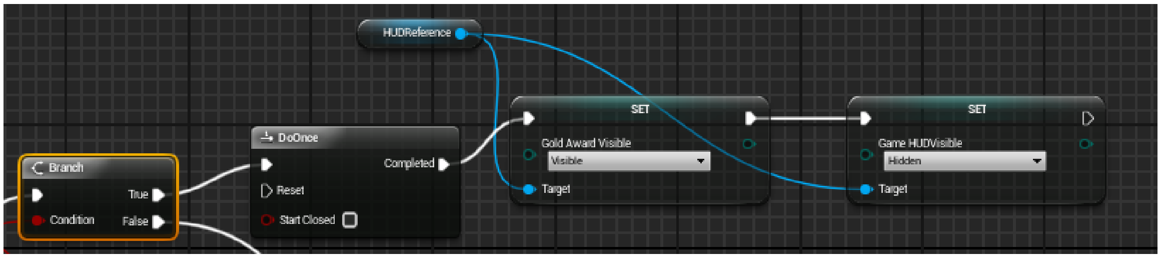


Figure 2-30

If the player's ActualRaceTime is not less than or equal to the GoldTime, the system will perform another check to see if the player's ActualRaceTime is less than or equal to the SilverTime and greater than the GoldTime. If this condition is met, the SilverAward HUD will become visible to the player and the GameHUD will be hidden.

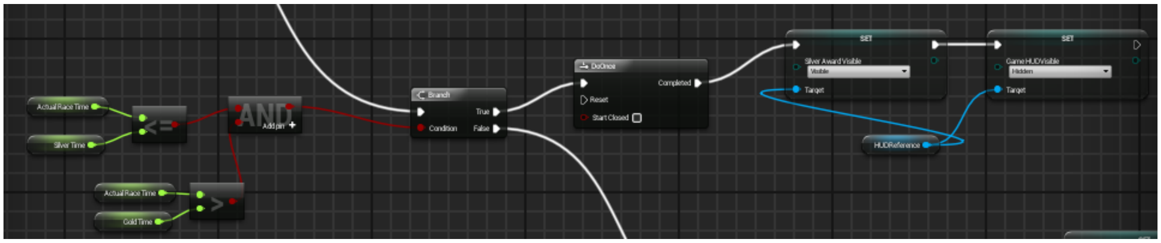
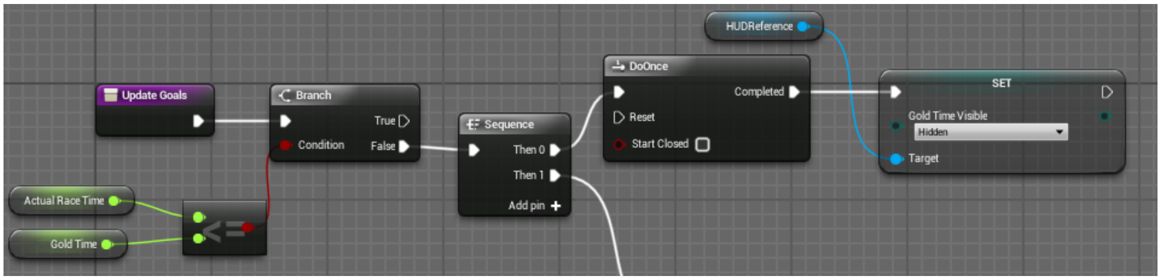


Figure 2-31

Finally, if none of the previous conditions have been met, the system will perform another check to see if the player's ActualRaceTime is less than or equal to the BronzeTime and greater than the SilverTime. If this condition is met, the BronzeAward HUD will become visible and the GameHUD will become hidden. If this condition is not met, the system will do nothing as the player did not achieve an award.

2.3.10 Update Goals

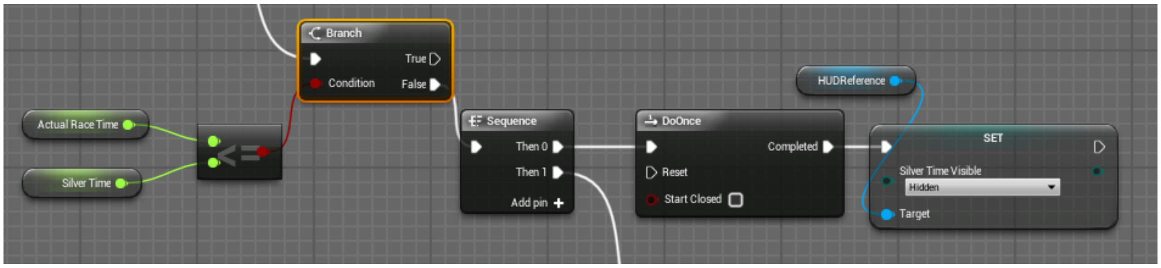
This function sets the gold, silver and bronze times to hidden on the GameHUD if they can no longer be achieved. Unlike the Award Check function it does not stop the race time, as it updates during the race. This function checks to see if the ActualRaceTime is less than or equal to the GoldTime. If this is true, it does nothing as the player may still achieve gold. If it is false, a sequence in performed which first sets the GoldTime to hidden, and then checks for another condition.



Gold Time Visible: Hidden

Figure 2-32

This next condition checks to see whether the player’s ActualRaceTime is less than or equal to the SilverTime. Again, if the condition is true, it will do nothing because the player can still achieve silver. If the condition is false, it will perform a sequence which will first set the SilverTime to hidden on the GameHUD, and will then perform another condition check.



Silver Time Visible: Hidden

Figure 2-33

Lastly, this next condition will check to see whether the player’s ActualRaceTime is less than or equal to the BronzeTime. If the condition is true, it will do nothing because the player can still achieve bronze. If it is false, it does not need to perform a sequence as there are no more awards that are possible for the player to achieve so it will simply just set the BronzeTime to hidden on the GameHUD.

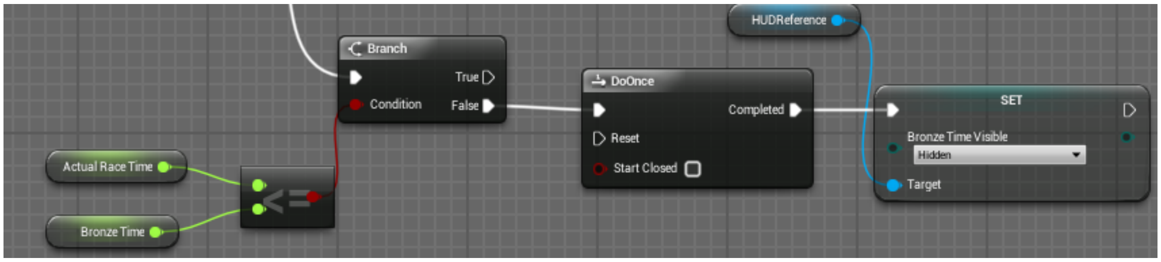


Figure 2-34

2.3.11 Movement Control

In order to be able to move the cars I had to create a number of InputAxis and InputAction events. InputAxis MoveForward sets the throttle value of the vehicle. InputAxis MoveRight sets the steering value of the vehicle. The InputAction Handbrake sets the handbrake input and sets the New Handbrake value to true when it is pressed and to false when it is released.

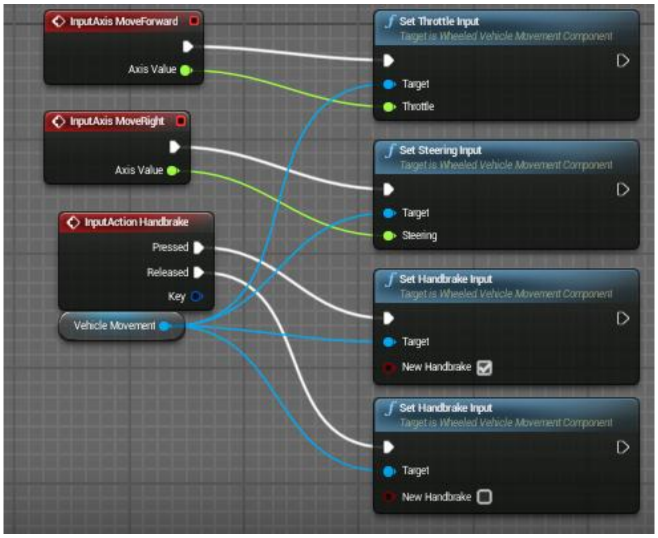
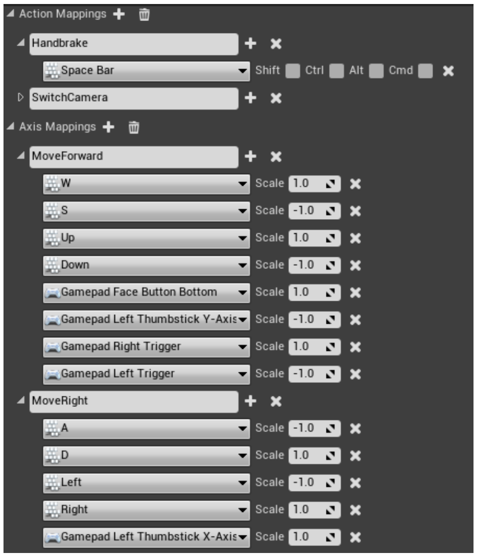


Figure 2-35

These InputAction and InputAxis events then need to be assigned input values so that they know when they are being called. This is done in the project settings section. Here we can see what button we use to control these movements. For example, in order to use the handbrake you must press the space bar. The MoveForward event uses the W key to make the car move forward, and the S key to put the car into reverse as the scale is set to -1. Similarly, the MoveRight event uses the D key to make the car move right, and the A key to make it move left as the scale is set to -1.



Action Mappings
- Handbrake
  - Space Bar — Shift □ Ctrl □ Alt □ Cmd □
- SwitchCamera

Axis Mappings
- MoveForward
  - W — Scale 1.0
  - S — Scale -1.0
  - Up — Scale 1.0
  - Down — Scale -1.0
  - Gamepad Face Button Bottom — Scale 1.0
  - Gamepad Left Thumbstick Y-Axis — Scale -1.0
  - Gamepad Right Trigger — Scale 1.0
  - Gamepad Left Trigger — Scale -1.0
- MoveRight
  - A — Scale -1.0
  - D — Scale 1.0
  - Left — Scale -1.0
  - Right — Scale 1.0
  - Gamepad Left Thumbstick X-Axis — Scale 1.0

Figure 2-36

2.3.12 Time Conversion

The time conversion blueprint converts the time from a float variable to a text variable. The value of the time variable is first divided by 60 in order to get the number of minutes. This value is then floored. It is then converted to type string. If the value is not greater than 9, then it will place a 0 in front of it so that it will be read as 01 etc. If the value is greater than 9, it does not need to place a 0 in front of it.

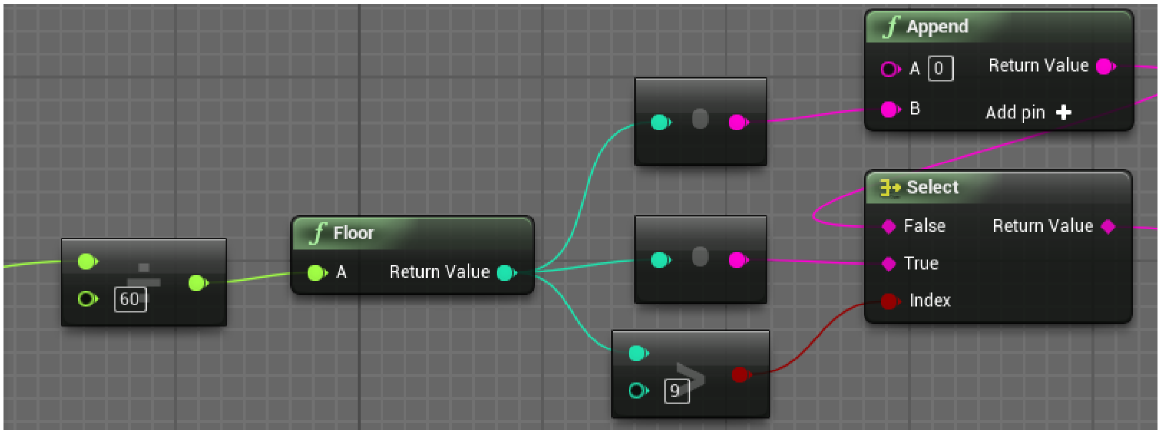


Figure 2-37

In order to convert to seconds it is the same process as the image above except that instead of dividing by 60, I used the % of 60 instead.

To convert to milliseconds the value of the time input is floored, which is then subtracted from the original value of the time input. This new value is then multiplied by 100 in order to be read as milliseconds. Then similar to the previous conversions, this value is floored and if the value is not greater than 9 a 0 is placed in front of it.

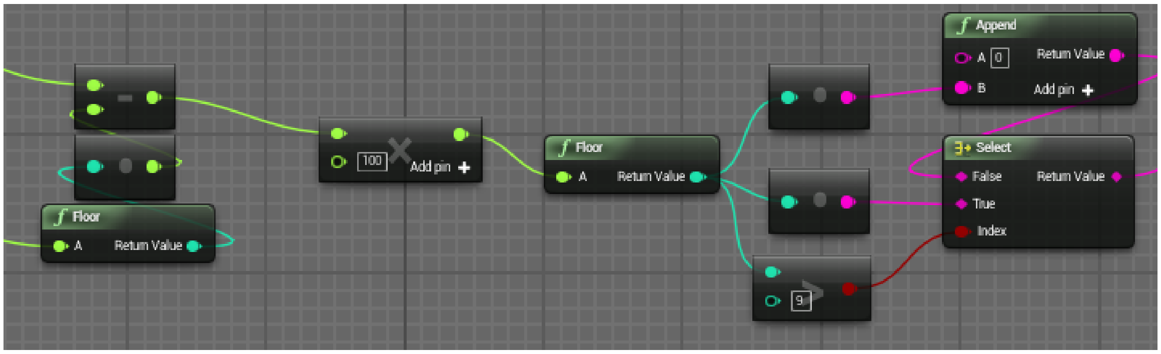


Figure 2-38

Following these conversions, the string is then built. The value for minutes is placed at the beginning of the string and is separated by a colon to the seconds. This is then placed in the first part of the string, with the next section being the seconds. The seconds are then separated from the milliseconds by a full stop. The milliseconds are then added to the string and the string is completed.

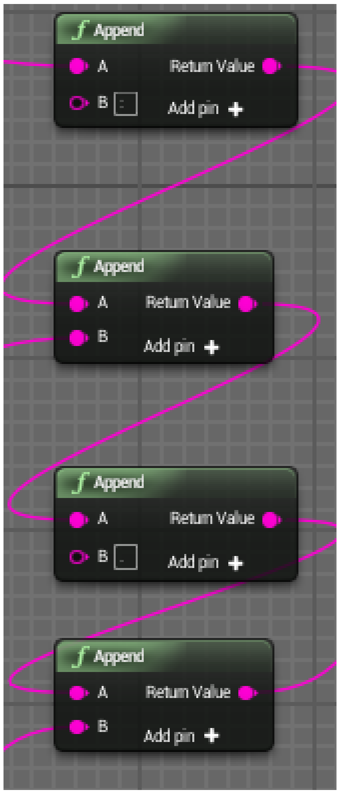


Figure 2-39

The string is then converted to text, and it is now able to be read by the game HUD.

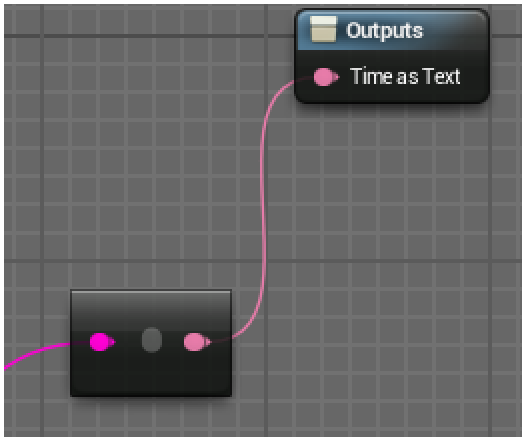


Figure 2-40

2.3.13 AI Component

Through using behaviour trees I was able to create a moving AI car. In order to do this I created two Blueprint classes called Update Route and Move to Location. The Update Route class determines where the next waypoint is and communicates with the Move to Location class, which then directs the car towards the next waypoint.

Figure 2-41 shows the behaviour tree which I use in my project.

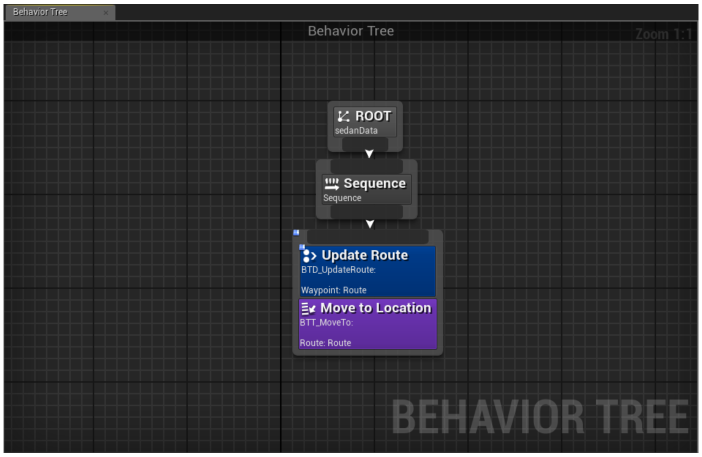
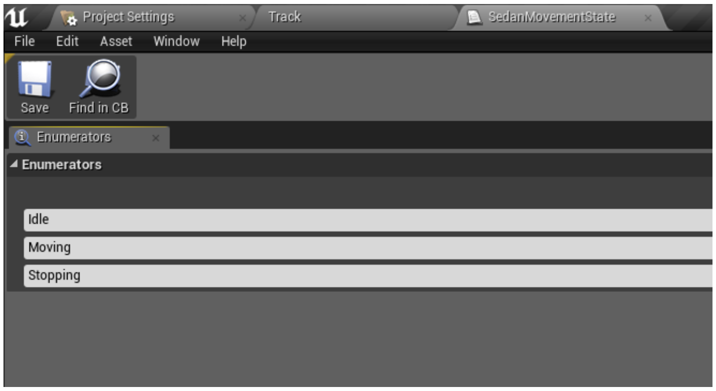


Figure 2-41

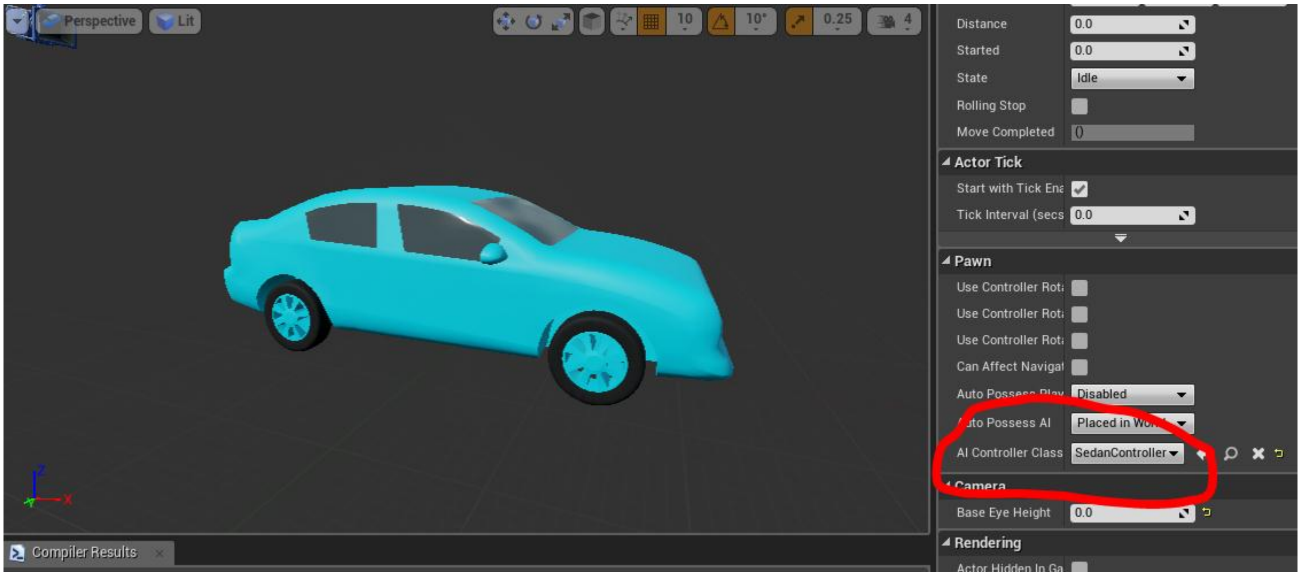
There are 3 movement states or actions for this behaviour tree which can be seen in Figure 2-42. It must be determined which state the car is in, so that the behaviour tree can decide what action to do next.



Idle
Moving
Stopping

Figure 2-42

When creating the AI car, I first had to assign an AI controller class. This controller class allowed me to communicate to the behaviour tree.



Distance 0.0
Started 0.0
State Idle
Rolling Stop □
Move Completed ()

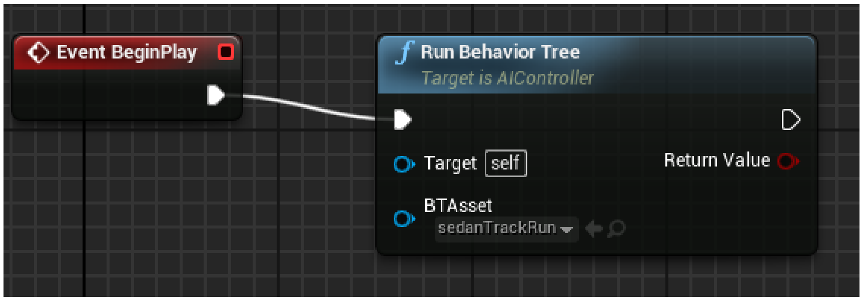
Actor Tick
Start with Tick Ena ☑
Tick Interval (secs 0.0

Pawn
Use Controller Rot: □
Use Controller Rot: □
Use Controller Rot: □
Can Affect Navigat □
Auto Possess Play Disabled
uto Possess AI Placed in Wor
AI Controller Class SedanController

Camera
Base Eye Height 0.0

Rendering
Actor Hidden In Ga □

Figure 2-43



Target self
BTAsset sedanTrackRun

Figure 2-44, Sedan Controller Class

Once I had the controller class connected to the behaviour tree I was able to use the behaviour tree to communicate further with the Update Route and Move to Location classes, which can be seen in the behaviour tree in the Technical Architecture section.

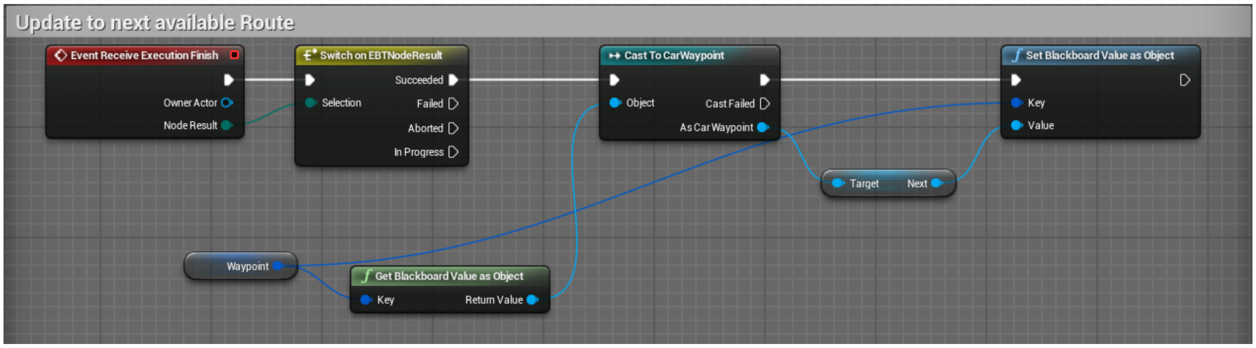


Figure 2-45, Update Route Class

This class continually updates the AI car's route. The Switch on EBTNodeResult searches for an output from the Event Receive Execution Finish that matches the input value, which is the next waypoint. This is then set in the behaviour tree blackboard, where the route key is stored as an object.

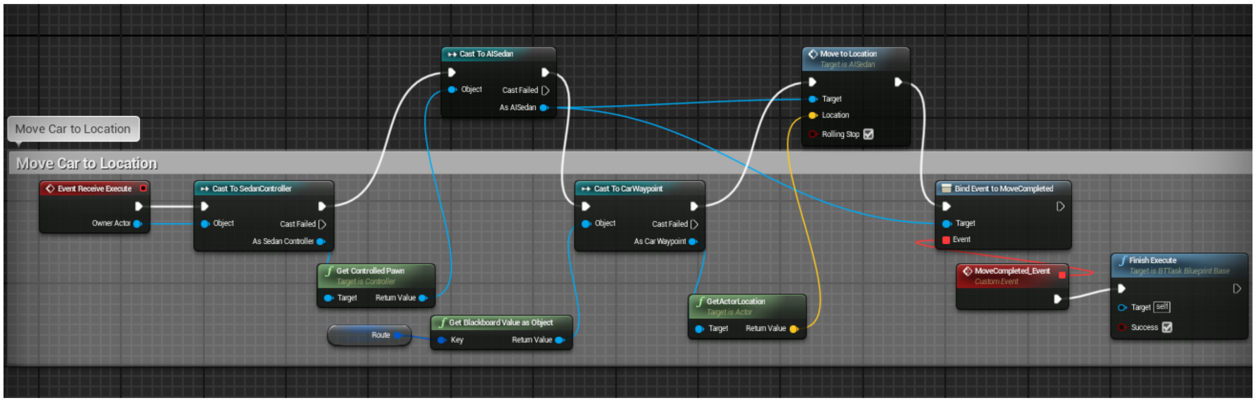


Figure 2-46, Move To Location Class

This class determines the player's current location and the location of the next waypoint. The AI actor is cast to the SedanController class which runs the behaviour tree. The behaviour tree then replays the sequence again. The car route is cast to the CarWaypoint which then directs the car to the waypoint.

Figure 2-47, Shows the AI Sedan Class. Inside this class there are constant calculations of speed and direction, updating of the route and moving to locations, along with the viewport of the Skeletal Mesh which the car uses.

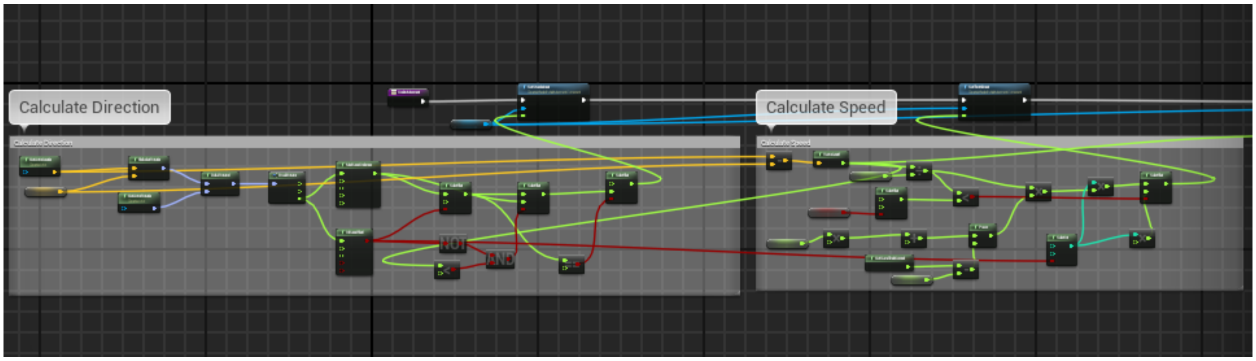


Figure 2-47, Shows the AI Sedan Class

Also inside this class, it sets the car's state which is then used in the behaviour tree. It checks to see if the conditions are met for a particular state to finish then clears the movement state. Next it stops updating the movement and resets the states.

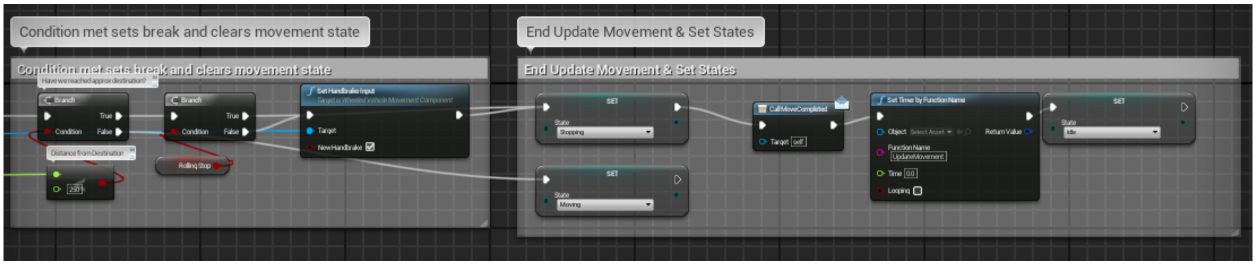

Condition met sets break and clears movement state
End Update Movement & Set States
Have we reached approx destination?
Branch
True
Condition
False
Distance from Destination
250
Rolling Stop
Set Handbrake Input
Target is Wheeled Vehicle Movement Component
Target
New Handbrake
SET
State
Stopping
Moving
Call Move Completed
Target self
Set Timer by Function Name
Object
Select Asset
Return Value
Function Name
UpdateMovement
Time 0.0
Looping
Idle

Figure 2-48

2.4 Testing

2.4.1 Unit Testing

The objective of unit testing is to isolate a unit of code and validate its correctness. The Automation System provides the ability to perform this unit testing, using the power of the Unreal Message Bus in order to increase stability. Unit testing helped to ensure that individual aspects of the code work as intended.

2.4.2 Assertion Testing

Assertion testing was also used throughout the testing process of this project. It checks the Boolean values of certain variables. These variables should return true when this test is being carried out. If the return false, then I becomes clear that there is a bug in the project. The information that is returned in this testing is specifically for the developers, and is not user-friendly. They do not provide any error-handling and are strictly used to warn the developer of an existing error so that it can be addressed and corrected.

2.4.3 Integration Testing

Integration testing was also a logical method of testing this project. This type of testing combines different units and sees how they work together. Testing each aspect of this game was very elementary, and would be a very tedious process. Accordingly, integration testing was selected to test the multiple processes inside the game and how they work together. This allowed me to identify issues and bugs in the game when multiple processes run together.

2.4.4 Usability Testing

In order to determine how the users interact with the interface of this project, usability testing was used. This was made possible with the help of 3 students. Through the use of usability testing, I was able to improve my UI based on where the users were looking when the game was at a static background, such as my main menu, and also when the game was in play. Sensing the learners' interest and capturing their behaviour in real time is a challenging task (Wei, Moldovan, and Muntean, 2009).

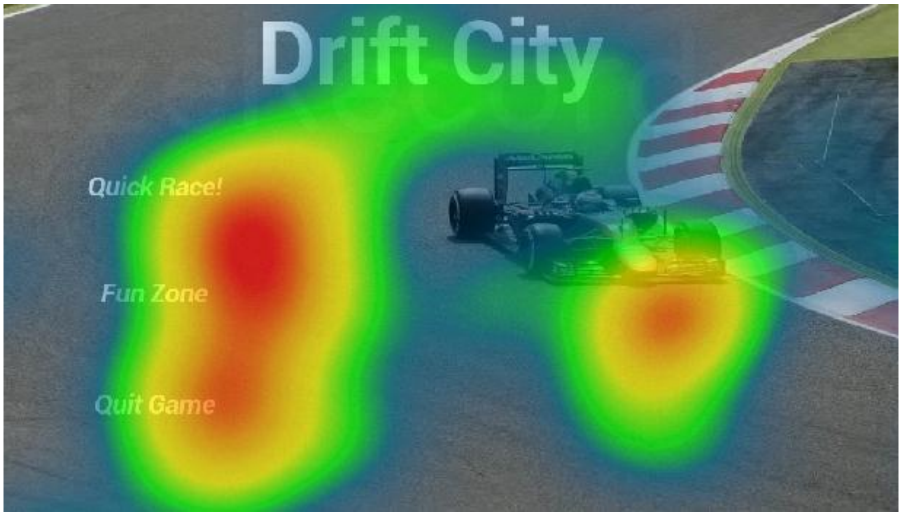


Figure 2-49

Figure 2-49 shows the Main Menu GUI which is the first place for the user to interact with the game. From this, we can see that the main point of focus for the user's attention was towards the buttons. While there was also a lot of focus on the car on the screen. These results are ideal as that is exactly where I wanted to draw the user's attention to.

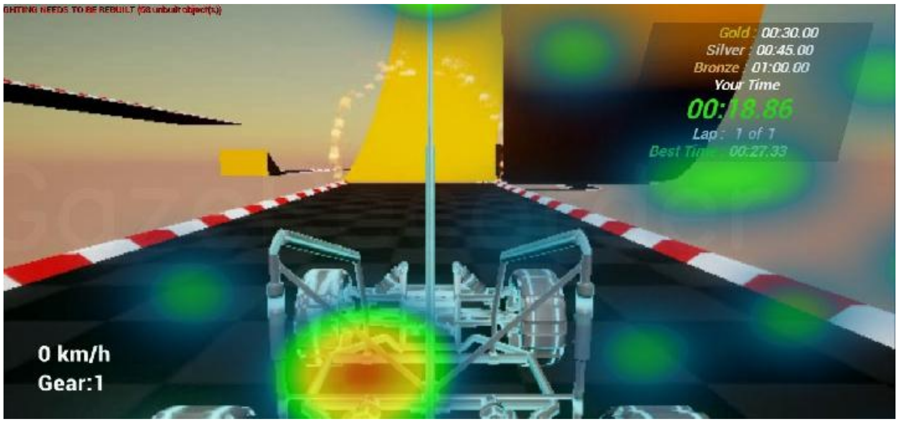


Figure 2-50

Figure 2-50 shows where the average user was focusing during a race. Most of the focus was on the car while there was also a lot of focus on the game HUD. Again, these results help to confirm that the design of the in game screen does not detract from the user's experience while playing the game.

2.5 Graphical User Interface (GUI) Layout

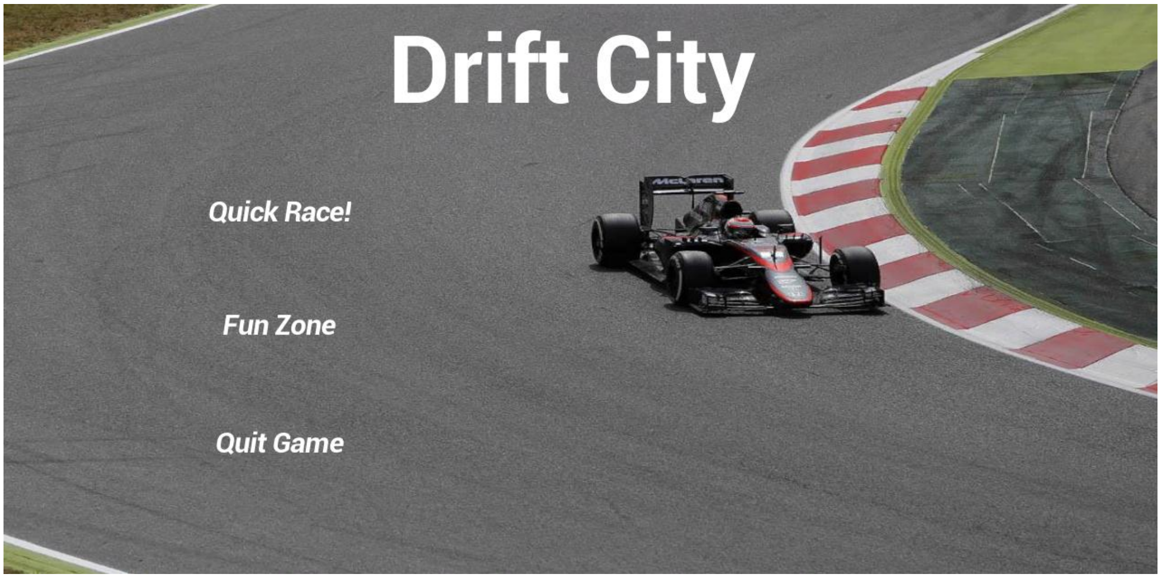
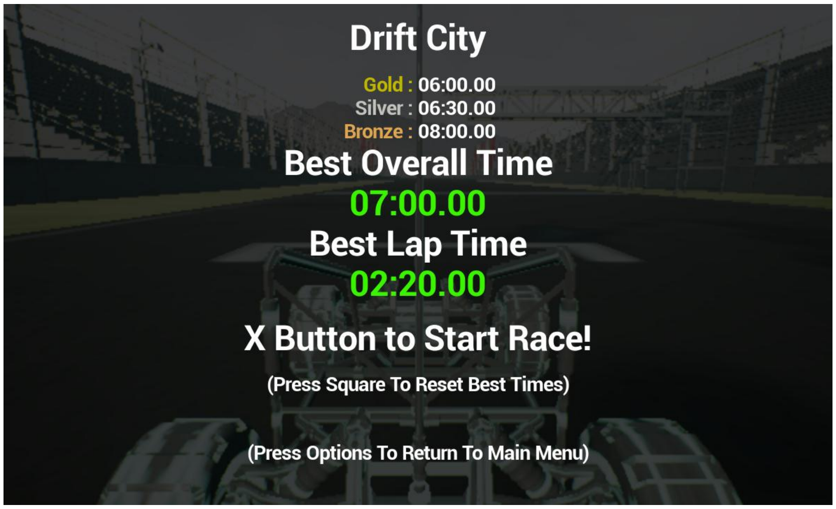


Figure 2-51

Figure 2-51 shows the game’s main menu. There are 3 options for the user. They can start a quick race against the computer, where they will take on the AI in a two lap race. Users can also go to the Fun Zone, where they can play a Rainbow Road-esque map which will test the player’s driving ability. And there is also a Quit Game button which allows the user to exit the game.

Figure 2-52

Figure 2-52 shows how my loading screen appears to the player. It is a very simple design, with a plain black background and small text in the bottom right hand corner letting the user know that the game is currently loading.



Drift City

Gold : 06:00.00
Silver : 06:30.00
Bronze : 08:00.00

Best Overall Time
07:00.00

Best Lap Time
02:20.00

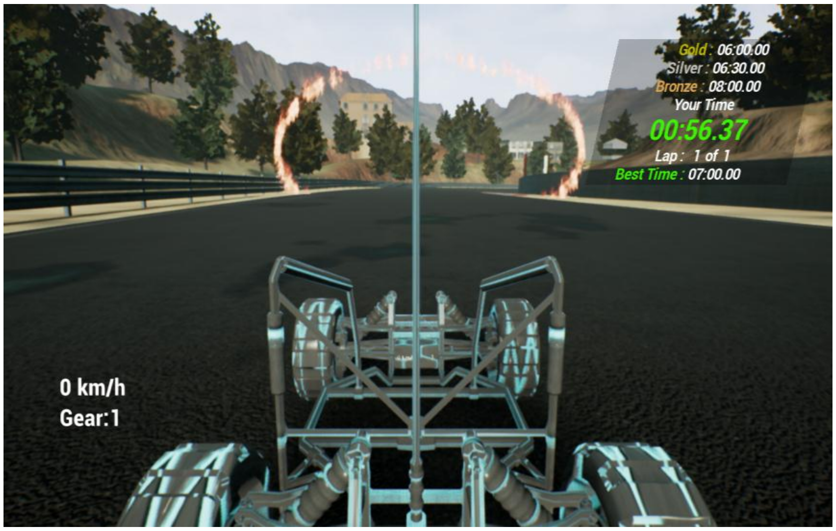
X Button to Start Race!

(Press Square To Reset Best Times)

(Press Options To Return To Main Menu)

Figure 2-53 Splash Screen

This is the Splash Screen GUI. This GUI displays the pre-set gold, silver and bronze times for each map, along with the best overall race time and the best lap time. Here the user can reset the best overall race time and best lap time to their default values by pressing the Square button on the controller. The player can also return to the main menu by pressing the Options button on the controller.



Gold : 06:00.00
Silver : 06:30.00
Bronze : 08:00.00
Your Time
00:56.37
Lap : 1 of 1
Best Time : 07:00.00

0 km/h
Gear:1

Figure 2-54

Figure 2-54 shows what the game looks like when the user is playing a race. The Game HUD can be seen in the top right hand corner, which displays the pre-set gold, silver and bronze times, and also the player’s current time and lap and also the best overall race time for that map. In the distance you can see the checkpoint which the user is required to go through. In the bottom left hand corner of the screen the player’s speed is recorded and displayed along with the gear the car is in.

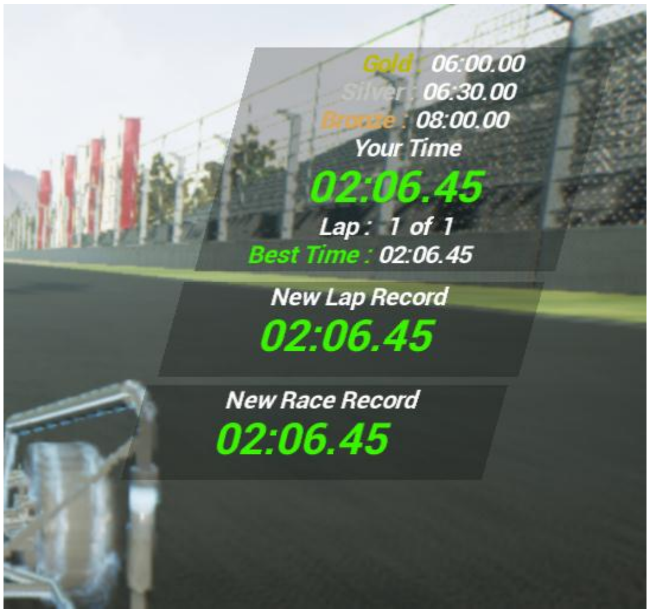


Figure 2-55, Lap and Race Record Displays

When the player manages to beat the best lap and race time record, a pop-up animation will appear on screen displaying the new records.

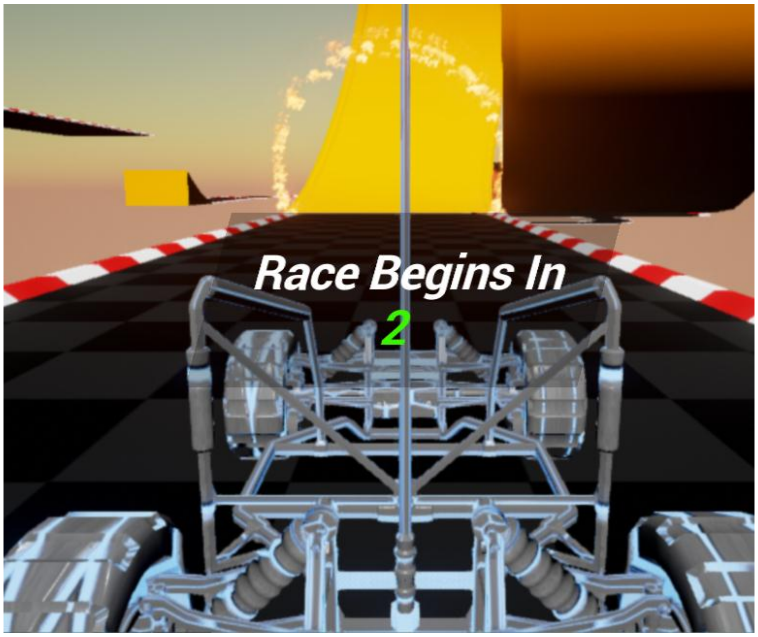
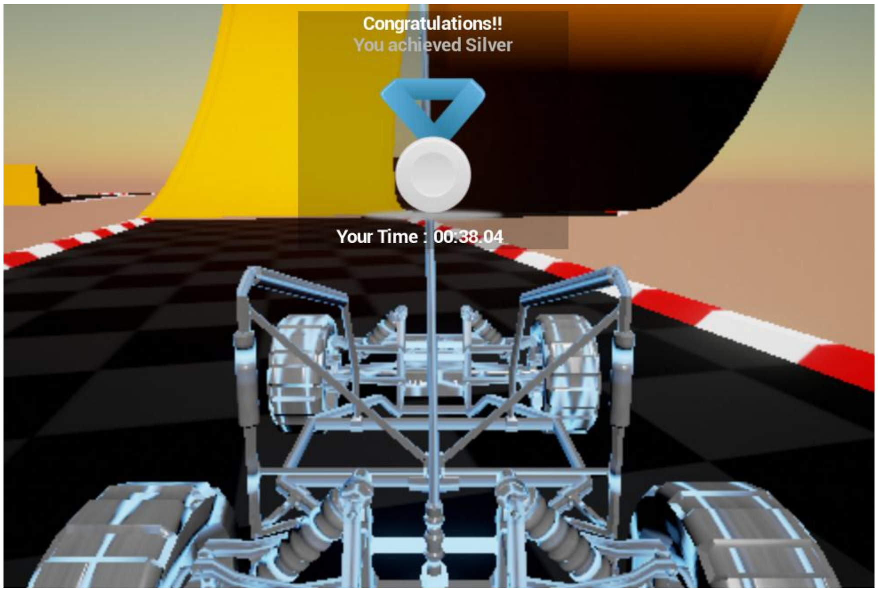


Figure 2-56

In Figure 2-56 we can see the countdown timer which is displayed before a race begins. It counts down from 3 to 0, after which the race begins and the player can move.



Congratulations!!
You achieved Silver

Your Time : 00:38.04

Figure 2-57

Figure 2-57 shows how the awards which the players receive based on their times are displayed. There is a trophy for achieving gold, and medals for receiving silver and bronze respectively. An image of the award is shown along with the player’s time.

2.6 Customer testing

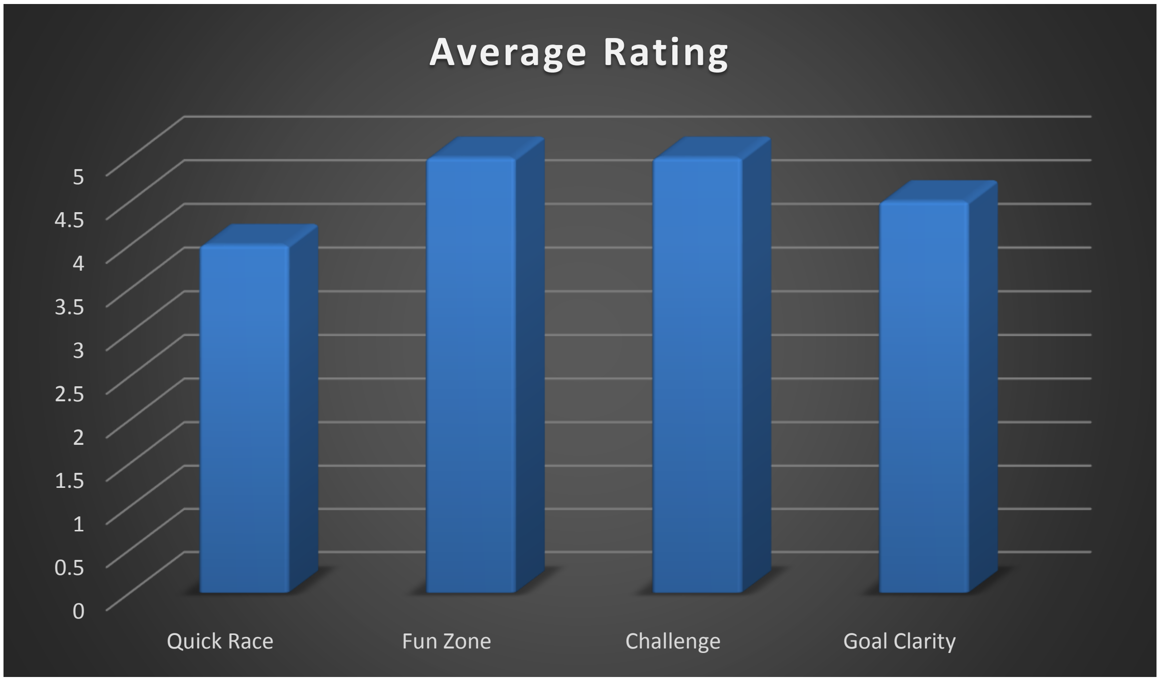
Extensive user testing took place. The reasoning behind this was to determine how the game performed when an average user who wasn't a developer was interacting with the software. The results were quite good during the initial tests. It showed me that the software was easy for people to use and that they enjoyed many features of the game. Doing this also allowed me to uncover many bugs which were present in my game. Over the course of user testing many changes were made to the game based on the feedback I was receiving. The game has continued to improve based on the customers' reactions. The most recent tests carried out were focused on improving the overall gameplay as most of the bugs present had already been addressed.

In order to carry out my user testing, I searched for people with different gaming backgrounds. Out of the 10 participants, 5 considered themselves casual gamers. The other 5 considered themselves to be novice gamers. The 5 casual gamers had more of a grasp as to the games' controls, while the 5 novice gamers had a slight bit more difficulty coming to grasps with them.

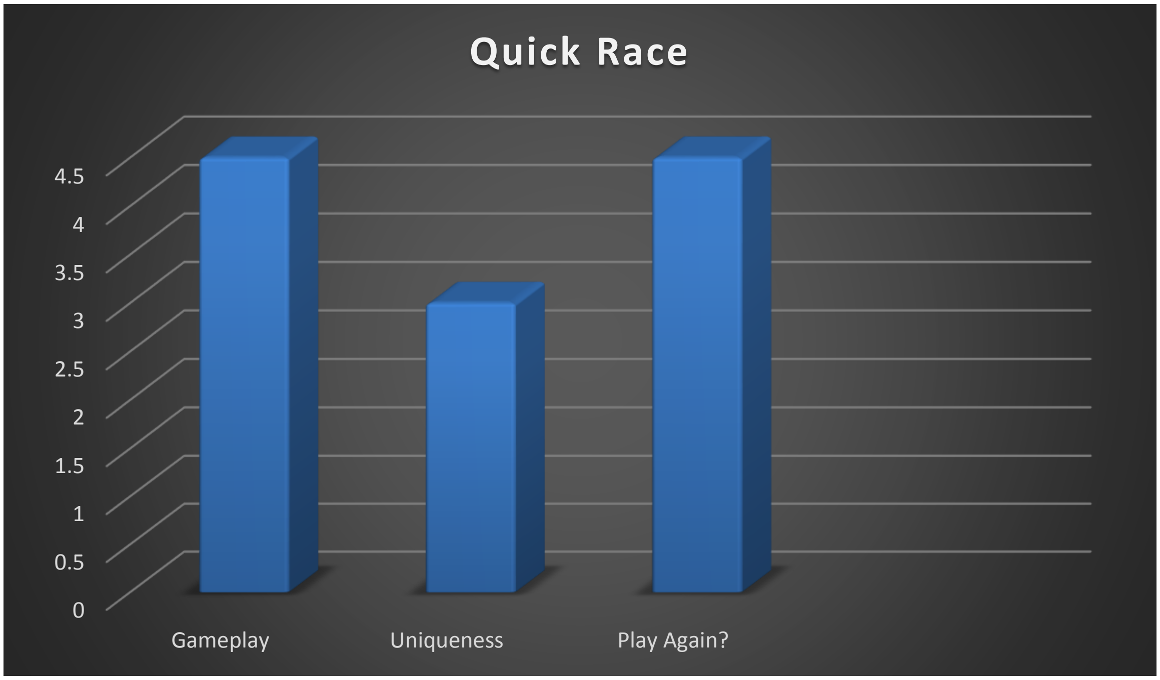
This evaluation allowed me to focus on how the game should provide more instructions to the novice gamers so that they can grasp the games controls and mechanics quicker.

2.7 Evaluation

The testers were asked to fill out a survey where they could provide feedback about the game. The users were asked questions based on the game and asked to rate the different aspects of the game from 1 to 5. The results of the feedback are as follows:



This chart shows that most aspects of my game rate highly among the testers. The lowest scoring category was my Quick Race. This is an aspect of the game where the users partake in a traditional race against AI. We can take a closer look at the reasoning behind these results, if we look at the Quick Race category.



From this chart we can see that although the testers rated the gameplay highly and seemed likely to play this again, they did not believe this section of the game was very unique. This is a problem with making racing games as most aspects have all been done before. In contrast to this, the Fun Zone went down extremely well with the testers.

3 Conclusions

This project takes an already well established game genre and tries to add some innovative aspects. Firstly, it provides the players with the traditional aspects of most racing games, allowing them to compete against the computer. This may not be unique to my game but I believe that it is an important aspect to every racing game. The main piece of innovation in this project is the creation of the Fun Zone, where the players can enjoy a fast paced racing track, jump through rings of fire, fly over ramps, and drive through loop-the-loops. The game saves the best times but also allows the users to reset these times so that they can set their own best times.

There was a number of disadvantages while making this game. Trying to make it unique and break the mould of the stereotypical racing game was the main difficulty that I encountered throughout this project. Many more advanced racing games have the resources and man power to create better, more up-to-date games, in comparison to myself, working on my own on this game over a 9 month period. I believe with a larger team and more time I could have achieved a lot more with this game. If there was additional time, more features could have easily been added to this game.

4 Further development or research

There are a number of ways which this project could be expanded. The first change that I would introduce would be introducing more cars to the game and implementing a car select screen before each race. Also, I would like to introduce car customisation where the user can choose from a number of decals and paints to allow the user freedom over their car design

Also incorporating a multiplayer experience would add an extra sense of competitiveness which often encourages people to constantly revisit the game. Players could race against their friends, or face-off against random opponents online.

Another aspect of the game which I would like to further expand would be the award system. As of now, it just displays the award to the user. With more time and resources, this system could store the award which the player wins and allow them to use these awards to unlock more cars or customization tools. This would act as another incentive for the player to continue playing this game.

This project has been developed and built with Windows PC platforms in mind. Once the expansions and changes are made to the project it could possibly be uploaded to Steam’s Greenlight service. Here, games are submitted with their descriptions, screenshots and videos of gameplay. Steam users can then look at these games and if enough people like the idea of the game, then it can be sold through the Steam store. However, there is an initial fee of €100 when submitting to Steam’s Greenlight service in order to prevent it from being spammed by developers. So, I would not upload to this service unless this project was near completion.

5 References

O'Reilly, Q.(2014, May 4). "Ever Wonder What The Irish Gaming Industry Looks Like Today? This Map Will Show You". *TheJournal.ie*., 2014. Web. 3 February 2016.

http://businessetc.thejournal.ie/irish-gaming-developer-map-1443124-May2014/

"Unreal Engine 4 Is A Professional Suite Of Tools And Technologies For Building High-Quality Games Across A Range Of Platforms". *Unrealengine.com*. N.p., 2016. Web. 4 February 2016.

https://www.unrealengine.com/unreal-engine-4

"1 – Overview". *YouTube*. N.p., 2016. Web. 9 May 2016.

https://www.youtube.com/watch?v=Eh03YShs9L8&list=PLZlv_N0_O1gZAjYuGIwtOHigppYa3CVLh

"Unreal Engine 4 Tutorial - UMG - Basic Menu". *YouTube*. N.p., 2016. Web. 6 February 2016.

https://www.youtube.com/watch?v=yua1KSjJDdY&list=PL3ffC7LUgtpe8yN7oeyWeW2rCeRRf8P6o

"UMG Loading Screen Tutorial". *YouTube*. N.p., 2016. Web. 23 April 2016.

https://www.youtube.com/watch?v=Gwdlpt7yc_c

Wei, H., Moldovan, A.-N. and Muntean, C. (2009) *Sensing Learner Interest Through Eye Tracking*. . In-line Citation: (Wei, Moldovan, and Muntean, 2009)

szydej (2016) GazeRecorder. Available at: https://sourceforge.net/projects/gazerecorder/ (Accessed: 10 May 2016).

6 Appendix

6.1 Project Proposal

Project Proposal

Drift City

Stephen Chapman, x12343316

x12343316@student.ncirl.ie

BSc (Hons) in Computing

Gaming and Multimedia

28/09/15

Objectives

The main objective of my project is to create a fully functioning 3D racing game. The game will consist of features that most players would expect of a racing game. The game will allow players to choose whether to play in first person or third person, modify or change their car and control their own settings. I believe this is important as it will enable players to make the game as personal as possible while still maintaining the same objectives. Players will be able to a custom character or choose from existing characters. This will help to make the experience of this game unique for each player.

There will be a tutorial level at the beginning of the game, which will go through the controls step-by-step and outline the objectives each race contains. Having completed the tutorial, the players will understand how the game works and what they must do. The tutorial will always be open for the player to revisit at any time during the game. Throughout the game the player will unlock more cars, depending on how well they performed in certain races. I think this is a fundamental part in most of the racing game genre. This will act as an incentive for the player to continue playing the game.

Each level of the game will have pre-set times each representing gold, silver and bronze medals. The player will receive the medal which’s pre-set time they have beaten. The better the medal they receive, the better the reward they earn. I feel this will make the player strive to do as well as possible in each race.

Background

The gaming industry has undergone a substantial evolution since the 1970s and is no longer just a fringe activity but instead it has become the mainstream. The gaming industry accounts for €240million in revenue, annually in Ireland. The process of creating a game no longer requires an abundance of employees. With today’s designing tools which are freely available, anybody with an interest to create a game can do so with ease.

In 1974, the first racing game was created by Atari, named Gran Trak 10. Gran Trak 10 was a coin-operated, arcade game which saw the player drive around a twisting track and attempt to beat the clock. Most modern games take advantage of the traditional racing genres elements, such as racing against the clock, customising the car and unlocking more cars. Many of the modern racing games, like Need for Speed and Forza Horizon, allow users to fully customise their cars using the money and/or skill points they have earned throughout the game. This is fun for the player as it allows them to create a unique car so they can add their own personal touch to the game.

As racing games are one of the most popular genres of games around today I found my investigation quite interesting as to what it takes for a game of this genre to become successful. Seeing as racing games are one of the most popular genres in today's market I believe that customers would enjoy a traditional racing game.

I chose to make a racing game because I am a huge fan of racing games. It is possibly my favourite area of gaming and I believe that working on a racing game would be extremely enjoyable. Need for Speed, is perhaps the most well-known title in this genre and its 23rd title will be released in November of this year. What I love most about racing games is that you are able to drive the most amazing cars around the world and put them to the test. Usually these super cars are locked until the player reaches a certain level or completes a certain race, so this makes the user have to play through the game in order to unlock the cars of their dreams.

Technical Approach

Research:

For this project I carried out a lot of research. I was unsure as to what I would like to do for the project so I took my time to decide what would be the most interesting to me. I decided on making a game so I researched the different genres of games, which engine to use and the engines capabilities. I decided to use the Unreal Engine as I believe that it would be the best fit for my project. Using Google, I researched other racing games that were created using this game engine. Before

I was fully decided on creating a racing game, I searched for tutorials on Pluralsight and YouTube that would be beneficial to me and so that I would be able to complete the project and not have to think of another idea. This filled me with the belief that I would be able to successfully finish the project.

Literature Review:

The subject under consideration in this literature review is which features I could add to my racing game in order to give it a unique feeling but still maintain the traditional racing game elements.

As a fan of racing games, such as Need for Speed and Forza, there are a number of elements from each game which I feel could possibly be used in my game. The two games in question are completely contrasting games. Need for Speed is a game with free roam capabilities where you do not necessarily need to do races constantly. Whereas, Forza is more of a racing simulation game as it tries to emulate the racing world. Personally, I enjoy both types of the games but the free roam mode from Need for Speed wins as I am not restricted to just doing races so it is something that I would like to contain in my game if possible. However, I would also like to include some of the more simulation type races from Forza and I think time trial races would be ideal to have in my game.

Requirements Capture:

Interface requirements

- User interface, tool tips, user customisation
- Hardware interface
- Software interface

Time Trials

Dialogue

Mapping

Reputation system

Instances

Implementation:

The implementation of this project will be carried out through the use of the Unreal Game Engine. My aim is to have a number of stages of development over the year. I aim to have the major features of the game complete by the end of the first semester. Throughout the second semester I will then aim to add time trials, dialogue, and a free roam area to explore. Once I have finished this I will then begin the testing process and hope to eliminate all the bugs and errors from the game. When I finally feel that the game has been fully polished off, I will ask a number of willing people to be beta testers and to contribute feedback about how it can be improved and also what they liked about the game.

Special resources required

I don't feel as if there are any special requirements required for the completion of this project. Once the PC hardware is reasonably up to date everything should be ok. This is especially true with the GPU (Graphics Processing Unit) as this is what is most important when creating or even running a game. If it is not reasonably powerful creating or even running the game may become a problem.

Project Plan

See attached MPP file.

Technical Details

The first three generations of the Unreal Engine included a sandboxed scripting language, UnrealScript, which provided a simple interface for gameplay programming that was shielded from the complexity of the C++ engine.

Evaluation

I will evaluate the game through the use of interested testers of this project. I will also use these testers to gather data, for example, which car they use the most, if the times set for the time trials are too difficult and also which designs they customise their cars with the most.

Stephen Chapman 28/09/15

Signature of student and date

6.2 Project Plan

See attached MPP file

6.3 Monthly Journals

6.3.1 September

Student name: Stephen Chapman x12343316

Programme: B.Sc. (Hons) Computing

Month: September

My Achievements

This month, I was able to decide what I was going to do for my project. I was struggling to choose an idea so I didn’t get much done for my project itself.

My contributions to the projects included finalising my idea and finding tutorials which could be useful to me when I start my project.

My Reflection

I was not successful in starting my project, but I submitted my proposal and project plan on time so I was happy with how this turned out, seeing as I had no idea what I was going to do for my project for a long time. Using my project proposal I have outlined many of the tasks that need completion for my project and with the use of a Gantt chart I have given timelines as to when I hope to complete these tasks. I now feel that I am in a good position to make a good attempt at starting my project as I have clear ideas as to what elements I want to include in my game.

Intended Changes

Next month, I will try to make a start on my project, and get started on creating the basic fundamentals in my game, such as the cars and how to move them. Once I get this started I feel I will be able to motivate myself and possibly do even more than just move the cars and hopefully I will have a start-up menu.

I realised that I need to put a lot of work into this project. For a while I thought I would just be able to breeze through it but having seen the amount of work that is required I will have to up my efforts.

Supervisor Meetings

As of yet, I still have not been assigned a supervisor.

6.3.2 October

Student name: Stephen Chapman x12343316

Programme: B.Sc. (Hons) Computing

Month: October

My Achievements

This month, I was able to complete my requirements specification document.

My contributions to the projects included meeting with my supervisor and establishing what would be ideal to make my game more unique and complex.

My Reflection

As of now, I am still yet to start work on my project but I have submitted my requirements specification on time. I am now thinking of an idea which will differentiate my game from other driving games and add a sense of fun to my game.

Intended Changes

Next month, I will try to add more aspects to my game more unique and not just a run of the mill driving game. I will research other driving games on the market today and see whether I can find an aspect which has not been as vastly explored as others. This will allow my game to stand out from others.

Supervisor Meetings

Date of Meeting: 04/11/15

Items Discussed: General idea of the project

Action Items: Arghir said that I should consider aspects of different driving games rather than just the typical driving game, such as Need for Speed. He told me to look at games such as Supertux kart which is more of a cartoon type game, which would add a sense of fun and make it unique.

6.3.3 November

Student name: Stephen Chapman x12343316

Programme: B.Sc. (Hons) Computing

Month: November

My Achievements

This month, I am still yet to start working on my project. This is because I have been focusing on other projects which are due in the next number of weeks.

My contributions to the projects included meeting with my supervisor and establishing what would be ideal to make my game more unique and complex.

My Reflection

As of now, I am still yet to start work on my project but having submitted my requirements specification on time. I am now thinking of an idea which will differentiate my game from other driving games and add a sense of fun to my game.

Intended Changes

Next month, I will try to make time to start the project just to get an idea of the work required.

Supervisor Meetings

Date of Meeting: 04/11/15

Items Discussed: General idea of the project

Action Items: Arghir said that I should consider aspects of different driving games rather than just the typical driving game, such as Need for Speed. He told me to look at games such as Supertux kart which is more of a cartoon type game, which would add a sense of fun and make it unique.

6.3.4 December

Student name: Stephen Chapman x12343316

Programme: B.Sc. (Hons) Computing

Month: December

My Achievements

This month, I was unable to perform any work on my project. This is due to a number of factors. Firstly I had numerous CAs and projects that were due at this time. Also, I was undertaking more hours in my part-time job due to the Christmas period.

My Reflection

Unfortunately I was not able to work on my project this month. My priorities lied with CAs, projects that were due and studying for my exams.

Intended Changes

Next month, I will make a start on my project. Once I have finished my exams I will begin on my project. I have a clear idea of what I now want from my game so I will knuckle down and begin to put in the work.

Supervisor Meetings

I did not meet with my supervisor this month.

6.3.5 January

Student name: Stephen Chapman x12343316

Programme: B.Sc. (Hons) Computing

Month: January

My Achievements

This month, I was able to make a start on my project. I downloaded a pack from the Unreal Marketplace which contained a car and a race track, which suited what I wanted from my game. I created my start menu and also developed a checkpoint system which keeps track of the number of laps and the lap time.

My Reflection

I was delighted with what I had achieved this month. Once I began working on the project I got really interested in it and kept wanting to continue working on it. Although towards the end of the month I was not able to work as much as I would have liked due to the amount of documentation required for the Mid-Point Presentation.

Intended Changes

Next month, I will continue to work on my project as best I can. After the Mid-Point Presentation I plan to incorporate opposing cars into the game, add another track and add a soundtrack to help to bring the game to life.

Supervisor Meetings

I did not meet with my supervisor this month.

6.3.6 February

Student name: Stephen Chapman x12343316

Programme: B.Sc. (Hons) Computing

Month: February

My Achievements

This month, I was focused on preparing a good prototype for the Mid-Point Presentation. I completely finished my HUD which displayed the lap which the player is on, best laps, lap time, and total race time. I was also able to store the time of the player's best laps using Blueprints in Unreal Engine.

My Reflection

I am content with what I have achieved this month. I thought that I had made good progress in my game for the Mid-Point Presentation. However, I only received a result of 51% with which I was quite upset. I plan on meeting up with Arghir again in the near future to hear feedback.

Intended Changes

Next month, I will continue to work on my project as best I can. I plan to introduce AI so that the player can race against the computer. Also, I hope to make a start on my Fun Zone and develop more ideas which would make it more interesting and unique.

Supervisor Meetings

4th February:

When I met with Arghir we went through my technical document, and ways to improve it before the submission.

6.3.7 March

Student name: Stephen Chapman x12343316

Programme: B.Sc. (Hons) Computing

Month: March

My Achievements

This month, I was focused on getting AI implemented into my project. I was able to achieve this and now AI cars drive around the track without further human interaction.

My Reflection

I am very happy with what I have achieved this month. I thought that I had made good progress in my game by implementing the AI. Having been able to achieve this, I should have possibly gone on to do more work with the user interface of the game, such as a creating a pause menu, and car selection before a race begins but I will begin this after exams.

Intended Changes

Next month, I will continue to work on my project once I have finished with my exams. My progress this month has given me some comfort as it was the main feature of the game, so once I am able to work on my game again I will.

Supervisor Meetings

21st March:

When I met with Arghir he told me that I should start working on implementing AI into my game as it was possibly the most important part, and also the best place to pick up marks for complexity of my project.

6.4 Other Materials Used

Gaze Recorder eye tracking software was used for Usability Testing.

6.5 User Manual

Keyboard:	Controller:
W / Up – Move Forward	Left Thumbstick Up – Move Forward
S / Down – Reverse	Left Thumbstick Down – Reverse
A / Left – Turn left	Left Thumbstick Left – Turn Left
D / Down – Turn right	Left Thumbstick Right – Turn Right
R – Reset Times	Square – Reset Times
Q – Quit to Main Menu	Options – Quit to Main Menu
Left Click – Start Race	X – Start Race
P – Respawn	L1 – Respawn
L – Restart Level	Share Button – Restart Level
T – Switch Camera	Triangle – Switch Camera