Functional Specification

CA400

Name:	Kyrylo Khaletskyy
Student Number:	15363521
Date of Completion:	16/11/2018

Table of Contents

1. Introduction

1.1 Overview

The goal of this project is to create a fun, challenging and unique game. AI Race is a 3D racing game targeted at computer gamers and racing game enthusiasts, the game aims to enable users to drive around the track, either on their own or race against enemy cars. Enemy cars will implement various machine learning algorithms in order to drive around a track. AI race is created using Unity Game Engine and written in C#, it implements two main features, Racing mode and the Learning mode.

The Racing mode focuses on the user interaction with the car and track, enabling a smooth and realistic gaming experience. AI Race uses custom car physics, including acceleration curves, steering, drag and camera movements which provides an authentic and unique driving experience not found in any other game. The user will be able to select between a set of different cars, each of these cars will have different speed, handling and drivetrain characteristics. Within Racing mode, the game contains two separate features, a time trial and a competition. In a time trial, a user can race around a track on their own trying to beat their best time or practice for a future competition. In a competition, they will face against enemy cars which use machine learning to drive around the track. Some more technically advanced players may wish to visit the Learning mode, it will allow users to visualise how the computer learns to go around a track, you will see its sensors, most fit in each generation and fitness factor using neural networks. Once sufficiently trained it will be used to race against the users in a competition, or against other algorithms. The end goal is to have the cars trained enough to be able to drive around any track given. The game will be wrapped in an attractive user interface which allows users to navigate through all the aforementioned features.

1.2 Business Context

Video games have been at the forefront of entertainment for decades. The video game industry has been an enormous source of income for companies such as Ubisoft, EA and Rockstar. This project is an entry into the game development industry and its advantages as a potentially profitable business idea. The user demographic will be computer gaming enthusiasts who enjoy various types of racing games. Currently, on Steam, the top 10 racing games alone sold 20 million copies, this doesn’t include games sold on other platforms and consoles. This project also extends into car automation in the real world. The use of various machine learning algorithms to autonomously drive around the track ties in with the recent gain in popularity of the car automation industry. Companies such as Tesla, Google and Nvidia have invested millions into the development of car automation making this project a very valuable learning experience.

1.3 Glossary

- **Steam:**
  PC gaming platform where users can buy games/software, play games and interact with other users online.
- **Unity:**
  Unity is a cross-platform game engine. The engine can be used to create both 3D and 2D games as well as simulations for its many platforms.
- **Racing Mode:**
  A mode within AI Race which contains its gaming features such as time trial and competition.
- **Learning Mode:**
  A mode within AI Race which allows users to visualise how the computer learns to go around a track, you will see its sensors, most fit in each generation and fitness factor using neural networks
- **Time Trial:**
  A racing mode where a user attempts to set the best time possible on a given track.
- **Competition:**
  A racing mode where a user faces against enemy cars in a race around a given track.

2. General Description

2.1 Product / System Functions

Main Menu/User Interface

AI Race will present users with a well-designed and aesthetically pleasing user interface which will be easy to navigate by any skill level. When a user launches the game, they will be greeted with the main menu screen, where a user can choose to continue with Racing mode or Learning mode, alternatively, they can navigate to the help section.

Game Mode Selection

Once in Racing mode, a user has the option between two types of races, a time trial or a competition. After a player selects time trial, they will be asked to select a car they want to race with, each car will have different speed and handling characteristics. Once the game loads the player will be able to drive around a track, have their time recorded and attempt to beat their personal best. Here a user will also be able to practice for a competition. In competition, a player will have the same options as in time trial but instead of driving on their own they experience what it is like to race against enemy cars which implement various machine learning algorithms to automate their driving styles.

Car Physics/Controls

AI Race uses all custom car physics and controls, this gives the game a unique feel not found in any other game. The game implements an acceleration curve which will have max speed and acceleration rate as variables which change depending on which car you pick. This is the same for the drag curve and steering angles. AI Race will use keyboard input for car controls, this will make the game accessible to a wider demographic, as every computer has a keyboard.

Machine Learning Visualisation

Once a user selects the learning mode, they will be able to view how a car learns to drive around a track using neural networks. A user will be able to see the evolution of a car's decisions based on its previous mistakes. Throughout the course of its evolution, basic details will be shown such as the car sensors and previous paths. Once a car is sufficiently trained it will be used to race against a user in competition mode. Of course, a user does not have to use the Learning mode to race, this is an optional mode for more technical users.

Help

From the main menu, a user will have the option to read the help screen, where they will be given basic information about the game, how to navigate it and how to control a car they wish to drive. A visual guide will be shown which buttons control the car and quick tips on how to get the most out of AI Race.

2.2 User Characteristics & Objectives

The game is aimed at anyone with a personal computer and an interest in video games, while the target audience will be young people it is expected that all age groups will have an interest in the game. As the user demographic is very broad it is essential to make the game and user interface usable in all scenarios for all age groups. There should be an intuitive learning curve to using/playing this game. In competition mode, the objective of the player is to traverse through a race track and attempt to beat the enemy cars, in time trial mode the objective of the player is to practice for a race and attempt to set the best time possible. Some more technically advanced players may wish to visit the machine learning mode where they can observe how a car learns to go around a track, but this is not mandatory.

Initial survey wishlist:

1. Implement different cars, each with different driving characteristics (speed, acceleration, drivetrain, handling, drag).
2. Implement multiple tracks to race on.
3. When in learning mode display a visual representation of the algorithm's attributes.
4. Implement soundtrack and game sounds which correlate with the car movements.
5. Provide a help section on the game showing how to control a car.

2.3 Operational Scenarios

USE CASE 1	Game launch and mode selection	
Goal in Context	User selects “Racing” or “Learning” mode.	
Scope & Level	System.	
Preconditions	User has successfully downloaded and installed AI Race.	
Success End Condition	User selects either “Racing” or “Learning” mode.	
Failed End Condition	User remains on the main menu.	
DESCRIPTION	**Step**	**Action**
	1	User launches AI Race.
	2	User is presented with the loading screen.
	3	User presses start.
	4	User selects mode.
EXTENSIONS	**Step**	**Branching Action**
	1a	AI Race must be installed on the user’s computer.
	1b	AI Race must be capable of running the game.
	2a	Game doesn't load, error shown.
	4a	User must select the desired mode to proceed.
VARIATIONS	**Step**	**Branching Action**
	5	User selects “help”
	5a	User is presented with a help screen containing a visual guide on how to play the game and control the car.
	6	User exits AI Race.

USE CASE 2	Learning mode	
Goal in Context	User observes the machine learning process.	
Scope & Level	System.	
Preconditions	User has successfully launched AI Race and selected Learning mode.	
Success End Condition	Car learns to successfully traverse a track over a period of time.	
Failed End Condition	Car doesn't progressively learn to traverse a track.	
DESCRIPTION	**Step**	**Action**
	1	User selects Learning mode.
	2	User is presented with the loading screen.
	3	Car attempts to traverse around a track.
	4	Car begins to traverse around the track over time, becoming progressively better over time.
	5	User observes learning process.
EXTENSIONS	**Step**	**Branching Action**
	1a	User selects the “Racing” mode instead, AI race will continue with “Racing” mode.
	4a	Car fails to progress around the track, error shown.
VARIATIONS	**Step**	**Branching Action**
	6	User returns to the main menu

USE CASE 3	Racing mode	
Goal in Context	User selects car and race type (time trial or competition).	
Scope & Level	System.	
Preconditions	User has successfully launched AI Race and selected Racing mode.	
Success End Condition	User begins to play with the selected car and Racing mode.	
Failed End Condition	User returns to the main menu.	
DESCRIPTION	**Step**	**Action**
	1	User selects Racing mode.
	2	User is presented with two race types
	3	User selects “Time Trial”
	4	User is presented with a selection of cars to choose.
	5	User selects car and proceeds with the game.
	6	User is presented with the loading screen.
	7	Game begins, and the user can play AI Race.
EXTENSIONS	**Step**	**Branching Action**
	3a	User selects “Competition”
VARIATIONS	**Step**	**Branching Action**
	8	User returns to the main menu

USE CASE 4	Time trial	
Goal in Context	User plays selected race type.	
Scope & Level	System.	
Preconditions	User selected “Time Trial” and picked the desired car.	
Success End Condition	User successfully traverses the track.	
Failed End Condition	User cannot successfully traverse the track.	
DESCRIPTION	**Step**	**Action**
	1	User is presented with the loading screen.
	2	Once loaded, the user will be able to freely drive around the track.
	3	The system records each lap time.
	4	Best lap time is shown to the user.
	5	Player uses keyboard buttons to control the torque, braking and steering of the car.
EXTENSIONS	**Step**	**Branching Action**
	6	User pauses the game
VARIATIONS	**Step**	**Branching Action**
	6a	User returns to the main menu

USE CASE 5	Race	
Goal in Context	User plays selected race type.	
Scope & Level	System.	
Preconditions	User selected “Competition” and picked the desired car.	
Success End Condition	User and enemy cars successfully traverse the track.	
Failed End Condition	User or enemy cars cannot successfully traverse the track.	
DESCRIPTION	**Step**	**Action**
	1	User is presented with the loading screen.
	2	Once loaded, the user will be facing against three other cars.
	3	The system will present the user with a countdown timer of them a race will start.
	4	Player uses keyboard buttons to control the torque, braking and steering of the car.
	5	Once the race starts user will attempt to beat the enemy cars.
	6	Enemy cars will be attempting to beat the player.
	7	Either user or the enemy cars win the race.
EXTENSIONS	**Step**	**Branching Action**
	8	User pauses the game.
VARIATIONS	**Step**	**Branching Action**
	8a	User quits to the main menu

2.4 Constraints

Hardware Constraints

Unity provides an option to export to multiple mediums such as computers, phones and consoles. For this project, AI Race will be exported to Windows PC. Based on Steam statistics Windows is the most popular computer gaming platform, also this is the system I use personally which will make it more feasible in terms of testing and evaluation. The only hardware constraint which AI Race faces is a player needs a PC which is powerful enough to run the game smoothly, this will need to be tested and guidance system requirements should be implemented in the documentation of the game. All development, testing and modelling of the game will be done on a Windows PC.

User Constraints

First and foremost, AI Race must comply with ease of use. The game intends to have a smooth framerate and be user-friendly in all scenarios. As mentioned previously the expected demographic will be very broad, while PC gamers tend to be "tech savvy", accessibility must be kept in mind throughout the implementation of the user interface. In terms of enemy car AI, the difficulty bust be kept challenging, but engaging so that a user is not bored when playing the game. During user testing an average lap time of all users could indicate how fast the AI should also traverse around the track, therefore the speed should be limited to a degree of the user capability.

Technical Constraints

The main technical components of AI Race are 3D models, C# scripts and AI algorithm implementations. The only technical constraints this project faces are limitations within the Unity Game Engine and my personal ability to implement and integrate these features. While this project plans to implement multiple AI algorithms, it may be wiser to only use one that performs the best in the final implementation, this will depend on how each performs during the development stage.

Time Constraints

AI Race is a 4th-year project for Computer Applications in DCU. This project is due on the 19th of May 2019. The aim is to complete AI Race and all supporting documentation before that date to allow time for exam study, time constraints may have an impact on the scope of the project. Currently, the plan is to implement multiple AI algorithms in order to see which one performs fastest around the track, if time will be running low, implementing just one algorithm may be a fallback option.

3. Functional Requirements

3.1 Car Physics & Controls

Description:

AI Race borrows its control layout from majority racing games that use arrow keys for throttle and steering, while space is used for braking. A user will have complete control over the movement of a car on the X and Z axis. The physics of the car such as acceleration, drag and center of mass are all implemented to mimic real car physics as close as possible with the time given to finish this project.

Criticality:

Controls must be intuitive and have a likeness with other racing games. Physics must be lifelike, substantial inconsistencies between the game physics and real-life physics can have a negative impact on the game, players can feel frustrated if a movement they expect doesn't happen.

Technical issues:

It is important to be mindful of the time spent on physics and controls, while it is important that they are realistic it is also important to finish other features on time with the project deadline. It is crucial the project doesn't get delayed perfecting and improving the car physics, as the possibilities are endless, after all, this project is not focusing on car physics simulation.

Dependencies:

Controls are essential to AI Race, without the player being able to control the car it cannot be called a game. Car physics must resemble how a car might move in real life, a player will be dependant on the consistency of these movements to compete with other cars. The controls will be dependant on the input provided by the user, while the physics will be dependant on the Unity game engine.

3.2 Machine Learning & AI

Description:

AI Race will be implementing multiple machine learning algorithms to automate cars to traverse a track. An evaluation will decide which algorithm performs best. Depending on the results, either the best algorithm will be in the final implementation of AI Race, or multiple algorithms will be used in order to simulate the different driving style of real-life racing drivers. For example, if a particular algorithm continuously fails to traverse a track or performs very poorly compared to others, it would be impractical to use it in the final implementation. Similarly, if multiple perform well, they will all be included in the final version of AI Race. The types of machine learning algorithms will be decided depending on the feasibility of integrating each within Unity. Preliminary research suggests that neural networks would be perfectly suited as one of the algorithms. This reinforcement learning model is often used for self-driving cars and would be perfect for the Learning mode of the game as the cars will learn from their previous mistakes to improve their future laps.

Criticality:
The implementation of this feature is extremely important, the majority of the workload of this project lies with machine learning. Without an adequate implementation of machine learning the cars cannot be automated to race against a user and the “Learning” mode will be obsolete.

Technical issues:
A significant technical issue will be the implementation of various machine learning algorithms, they need to be efficient and able to work in real time while a race is on. My ability to develop them within Unity and apply current and future knowledge of machine learning will be a considerable hurdle, as currently, I do not have a lot of experience in that area. As mentioned previously it might not be practical to use more than one machine learning algorithm. During the development stage, I will evaluate whether to use one or more algorithms in the final version of AI Race.

Dependencies:
Two main features of the system will be dependant on the implementation of machine learning. The “learning” mode which will be used as a visual guide on how the car learns to drive around a track, and the automation of enemy cars which will be racing against the user.

3.3 User Interface

Description:
Apart from when the user is in a game, the user interface will be used to navigate through the game menu. AI Race interface contains of the main menu, game mode selection, race selection, car selection and help screen. A theme will be chosen before the development process and will be used throughout all the menus, this will keep the game consistent and accessible for all users.

Criticality:
It is crucial to create a consistent and functional user interface. When a user is navigating through the menu they should never be lost, everything must be labelled concisely, and the text can be easily readable. Most importantly objects that are clickable must be differentiated from objects that are not supposed to be interacted with.

Technical issues:
A small technical difficulty would be to stay consistent throughout the project. To avoid this, rules for how the user interface behaves and looks must be set before the development of the user interface, as a result, design changes won’t hinder the aesthetic of the game.

Dependencies:
The navigation through the game menu and game mode selection is dependant on an effective execution of the user interface.

3.4 Racing Mode

Description:

After the user selects to play AI Race, they will have to option to play in a time trial or a competition. When a player picks a desired game mode, they will be asked to select a car they want to race with. In a time trial, a user will attempt to set the best time possible. In a competition, the user will be racing against enemy cars as mentioned in section 3.2.

Criticality:

The game modes are an essential part of AI Race as they make it a game. Both modes are a showcase of the features previously mentioned. Racing features such as time tracking, lap tracking and start/finish timers need to be accurate and enforced on both the player and the AI, otherwise a player might feel cheated. For example, if an enemy car finishes second, yet the system displays it as being first.

Technical issues:

A slight technical issue is my ability to adequately enforce all the rules mentioned above. All data must be accurate and reflected as such to the player.

Dependencies:

AI Race depends on the different game modes to showcase the features such as car physics and machine learning algorithms.

3.5 Help

Description:

The help screen will provide a visual guide for users not familiar with PC racing games. It will also describe how to navigate the game menu and game mode selections. The help screen can be accessed from the main menu.

Criticality:

It is crucial to explain how to navigate and play AI Race. While the demographic is mostly focused on computer gaming enthusiasts, it is essential to make the game accessible to users never before familiar with racing and computer games. It might also be beneficial to provide an installation guide for AI Race in the user manual.

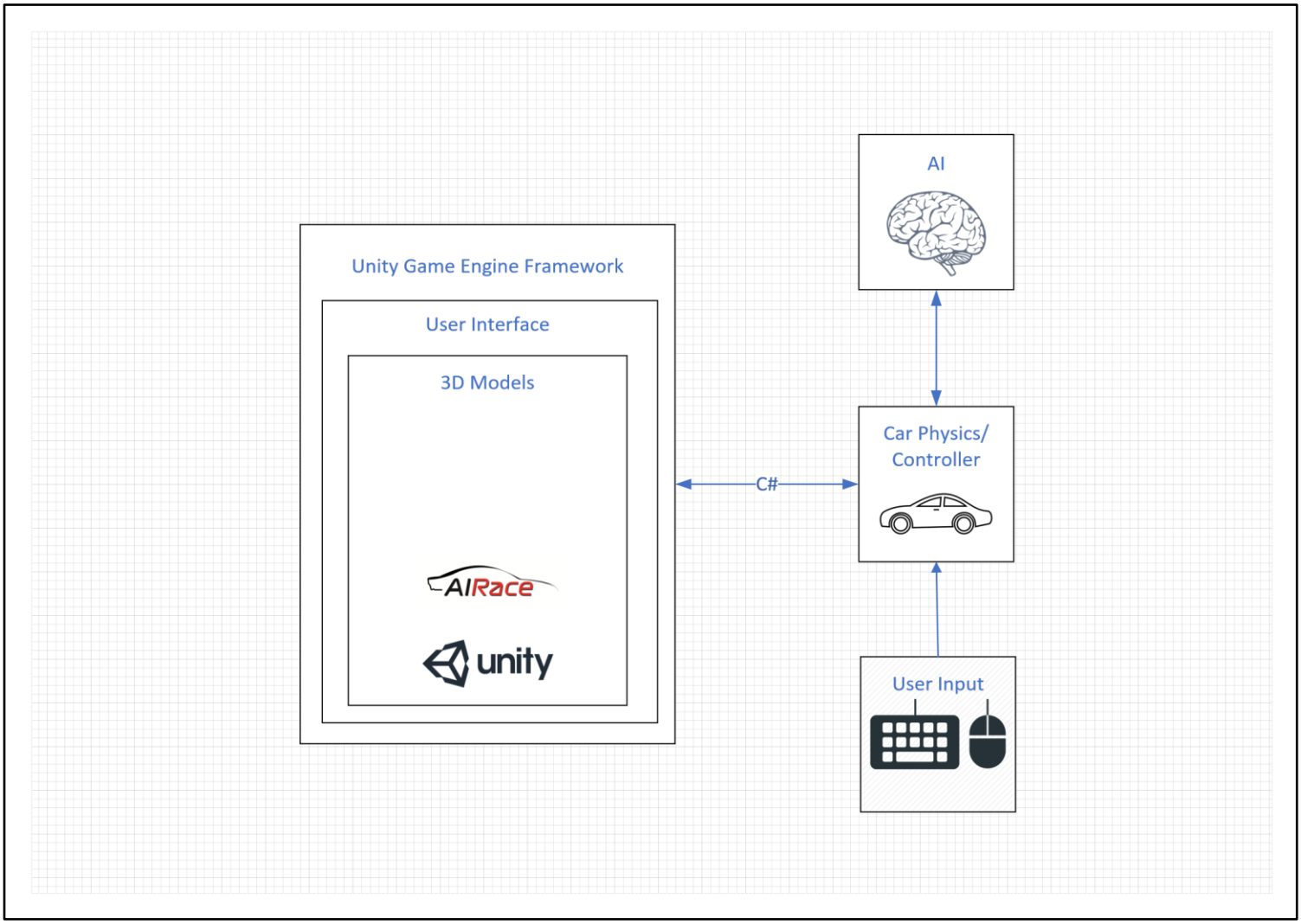
Technical issues:

Help screen within AI Race does not present any technical difficulties at this point.

Dependencies:

Some users will be dependant on the help section to explain how to play and navigate the game. The help screen itself will be dependant on a correct implementation of the main menu and user interface.

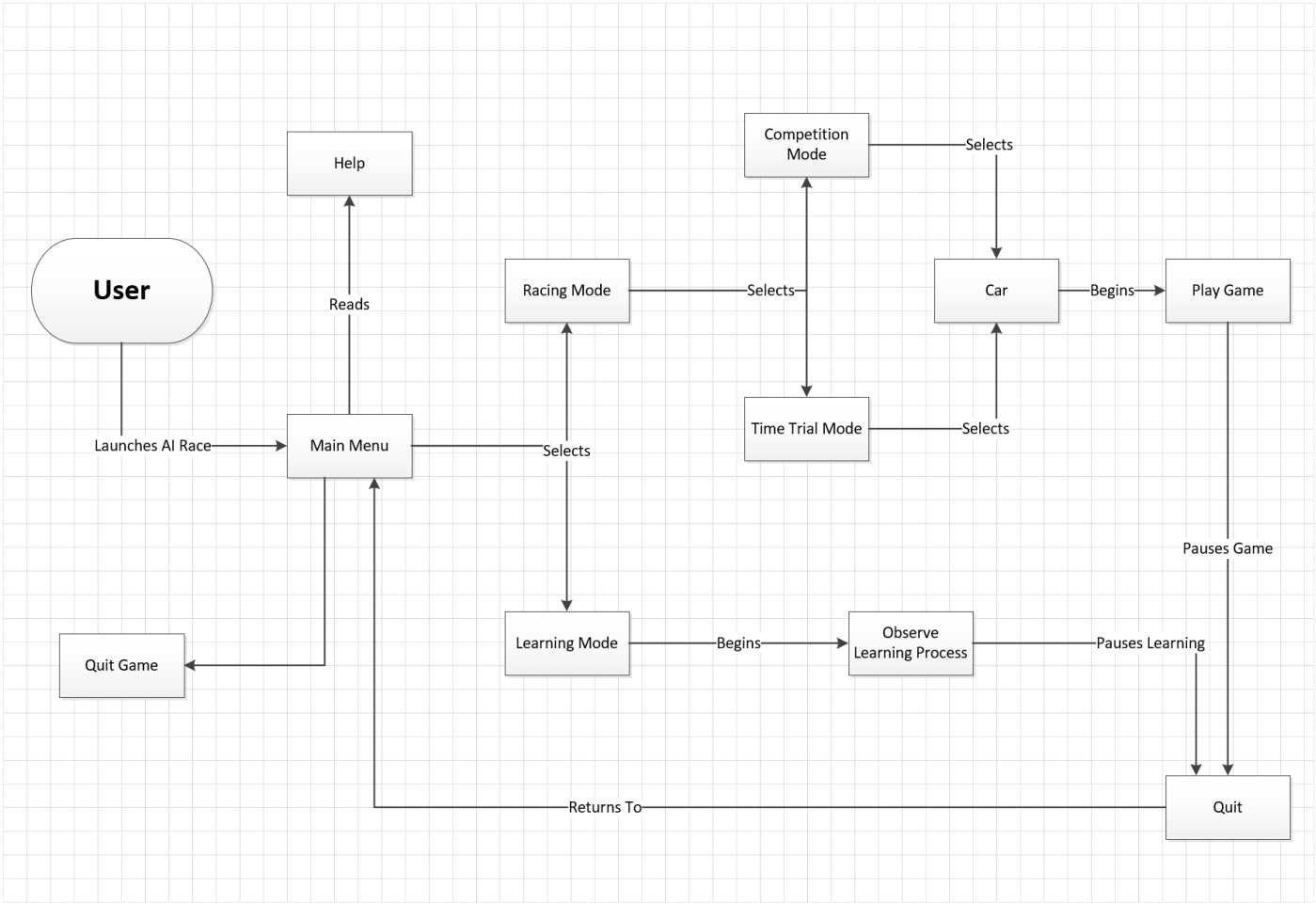
4. System Architecture



The above system architecture diagram shows the interaction between the Unity game engine and its components. The user input and AI interact with the car physics/controller. In turn, the car physics/controller interacts with the Unity game engine to control the 3D models.

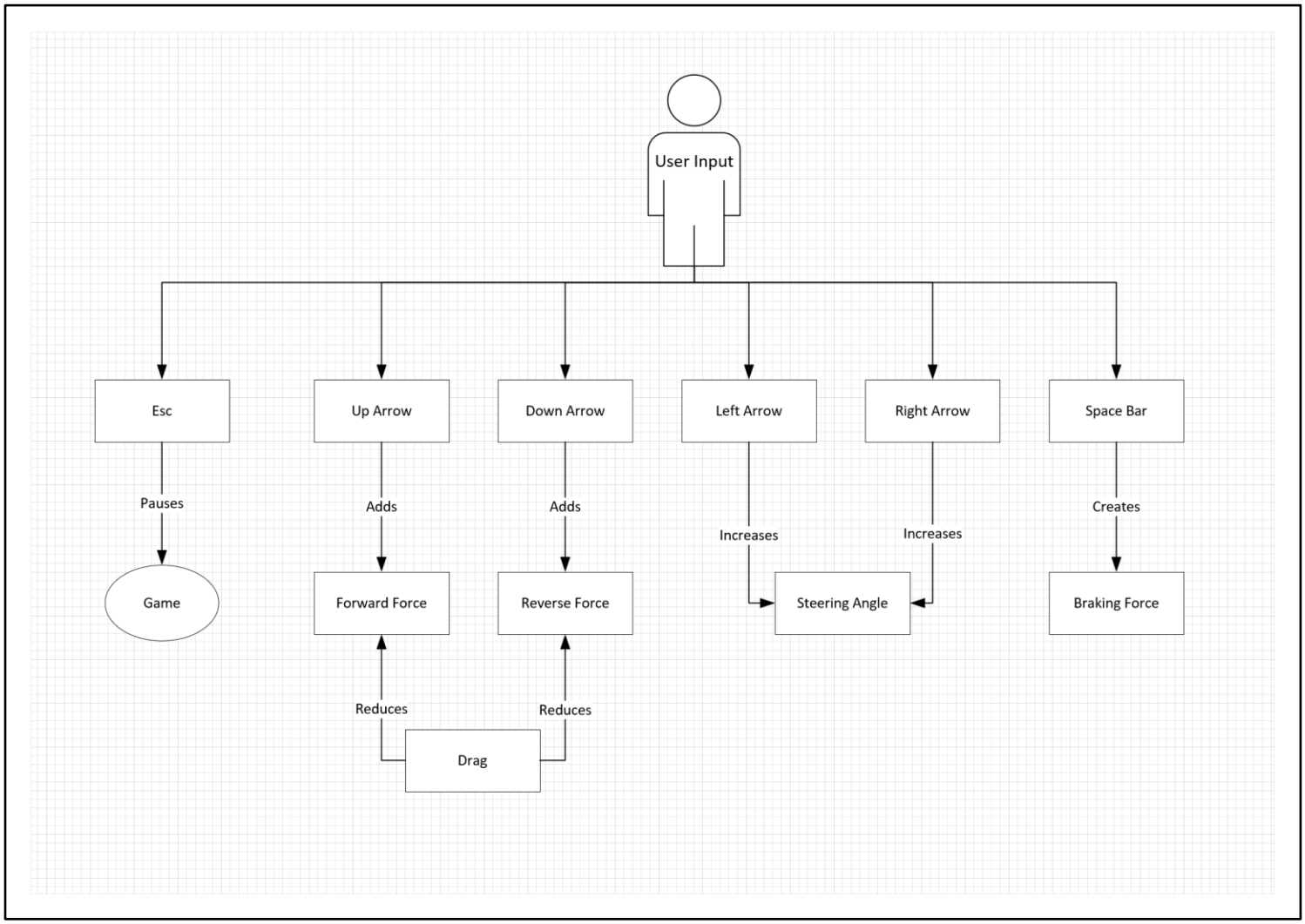
5. High-Level Design

5.1 Process Diagram



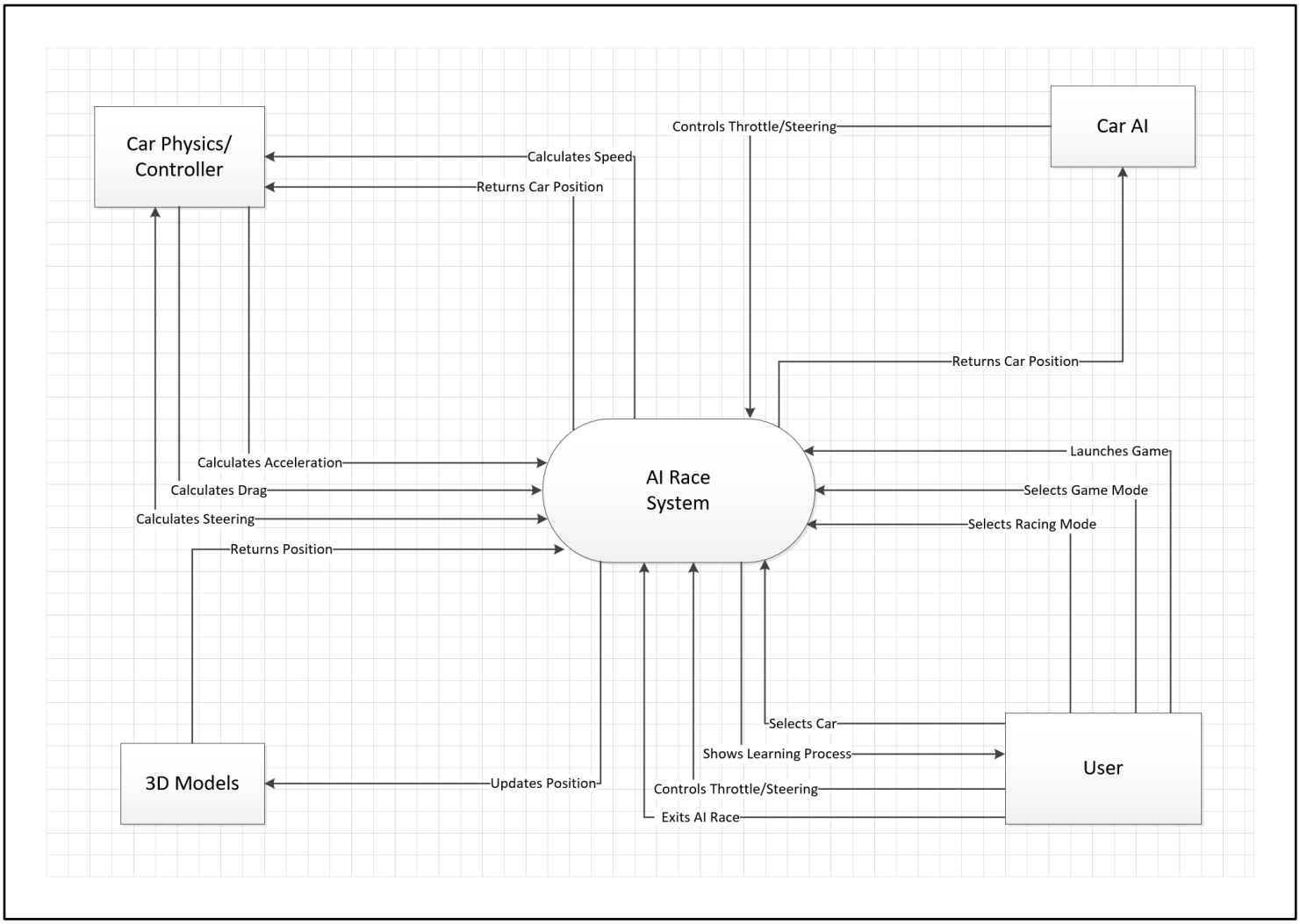
The process diagram above is a graphical representation of the movement of processes through the system. It shows how each process works at step by step intervals in the order that they must be carried out. For example, when a user launches AI Race they will be met with the main menu, from where they can select the help screen, racing mode or learning mode.

5.2 User Input Process Diagram



The user input process diagram above shows a more in-depth version of the “Play Game” process from the previous diagram. The user input and the process that follows that input is shown.

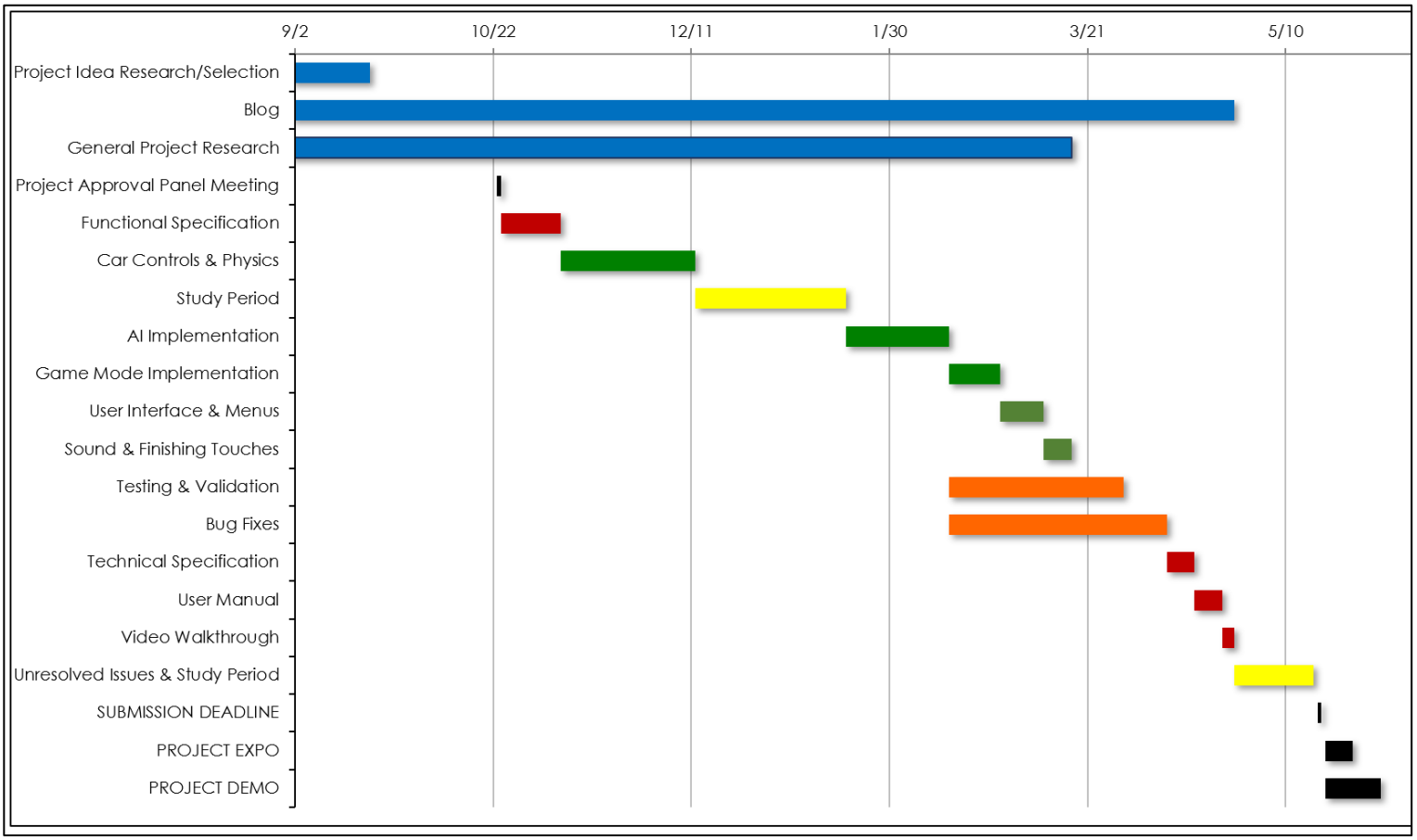
5.3 Context Diagram



The context diagram above shows how each feature or person interacts with the system, each feature shows a set of functions it controls within the system and a set of functions that the system controls within the feature.

6. Preliminary Schedule

6.1 Gantt Chart



The above chart is a graphical representation of the intended schedule for the features mentioned in the functional specification thus far. Each task has an intended start and finish date (exact dates can be viewed in the timetable on section 6.2), with time left for exam study and unforeseen issues.

6.2 Timetable

TASKS	START	END	DAYS
Project Idea Research/Selection	9/1	9/21	20
Blog	9/1	4/27	238
General Project Research	9/1	3/17	197
Project Approval Panel Meeting	10/23	10/24	1
Functional Specification	10/24	11/8	15
Car Controls & Physics	11/8	12/12	34
Study Period	12/12	1/19	38
AI Implementation	1/19	2/14	26
Game Mode Implementation	2/14	2/27	13
User Interface & Menus	2/27	3/10	11
Sound & Finishing Touches	3/10	3/17	7
Testing & Validation	2/14	3/30	44
Bug Fixes	2/14	4/10	55
Technical Specification	4/10	4/17	7
User Manual	4/17	4/24	7
Video Walkthrough	4/24	4/27	3
Unresolved Issues & Study Period	4/27	5/17	20
SUBMISSION DEADLINE	5/18	5/19	1
PROJECT EXPO	5/20	5/27	7
PROJECT DEMO	5/20	6/3	14

7. Appendices

Unity:

https://unity3d.com

Car Acceleration/Torque:

http://lancet.mit.edu/motors/motors3.html

Steam:

https://store.steampowered.com/stats

3D Models:

https://assetstore.unity.com