



Palestine Polytechnic University

College of Information Technology and Computer Engineering

Motivation Game To-Do Tasks

أدرينالين | Adrenaline

Team members:

Lubna Qadi

Noor Al-Jubeh

Supervisor:

Liana Tamimi

Hebron - Palestine

2022-2023

## إهداء

بسم الله الرحمن الرحيم الرحيم، والحمد لله رب العالمين الذي وفقنا وأعانا على إنهاء هذا المشروع والخروج به بهذه الصورة المتكاملة، فبالأمس القريب بدأنا مسيرتنا التعليمية ونحن نتحسس الطريق برهبة وارتباك، فرأينا أن ( تخصص علم الحاسوب ) هدفاً سامياً وحباً وغاية تستحق السير لأجلها، وإن مشروعنا يحمل في طياته طموح طالبات يحلمون أن تكون أمتهم العربية كالشامة بين الأمم.

وانطلاقاً من مبدأ أنه لا يشكر الله من لا يشكر الناس، فإننا نتوجه بالشكر الجزيل للأستاذة الفاضلة المعلمة الدكتورة ( ليانا التميمي ) التي رافقتنا في مسيرتنا لإنجاز هذا المشروع وكانت لها بصمات واضحة من خلال توجيهاتها وانتقاداتها البناءة والدعم الأكاديمي، كما نشكر عائلاتنا التي صبرت وتحملت معنا ورفدتنا بالكثير من الدعم على جميع الأصعدة ، ونشكر الأصدقاء والأحباب وكل من قدم لنا المعنوي من كلمات طيبة و تشجيع لإنهاء ما بدأناه ، هي رحلة بدأناها ولم نكن نعلم ما نهايتها استعنا بالله وتوكلنا عليه و شددنا الهمم و سعينا لنصل إلى هذه النهاية التي لولا دعم الكثير و وقوفهم معنا طوال الطريق ما كنا وصلنا لها وكما يقال من طلب العلا سهر الليالي فكم من ليلة سهرناها وكم من وقت ضحينا به من أجل الوصول إلى هذه المحطة لنرفع رؤوس كل من عرفنا ، لنكون فخرًا لوطننا الحبيب فلسطين الغالية على قلوبنا ، فالشكر من القلب لكل من رافقتنا في رحلتنا هذه و جعل ما كنا نحلم به حقيقة وواقع نهايته أجمل مما تخيلناه ، والصلاة والسلام على اشرف الخلق والمرسلين سيدنا محمد صلى الله عليه وسلم

# Abstract

---

Since video games have become so popular recently, it was important to use entertainment to encourage students to finish their academic work. This is especially true of university students, who are constantly looking for ways to distract themselves from their studies.

The goal of the project is to create a game that will assist students in completing the academic tasks they set out to perform by giving them rewards for finishing tasks after they have verified that they have been finished.

The application will be a car racing game where the vehicles and tracks are chosen for racing and the racing car is developed using the rewards earned from tasks completion. After completing these tasks, he will receive a financial reward or anything else he will perform on the game's development, in order to improve his performance in the race and try to raise his points and get higher points, which will motivate the student to complete his tasks and compete against other students, increasing the productive competition between the students. This means that the student will try to complete his tasks in order to be able to obtain rewards that will develop his racing car or buy a new car with higher specifications.

## الملخص

---

مع انتشار ألعاب الفيديو في الأونة الأخيرة، نرى الشباب وخاصة طلاب الجامعات يبحثون عن طرق لإلهاء أنفسهم، بسبب عدم وجود دافع للدراسة لديهم، لذلك كان لا بد من استخدام ألعاب الفيديو لتحفيز الطلاب على إكمال مهامهم الدراسية. تقوم فكرة المشروع على عمل لعبة للمساعدة على إنجاز المهام الدراسية التي يحددها الطالب للقيام بها عن طريق الحصول على مكافأة عند إنجاز مهمة ما بعد التأكد من أن المهمة قد تم إنهاؤها .

حيث سيكون التطبيق عبارة عن لعبة سباق سيارات يتم فيها تحديد السيارات والمسارات المختارة للسباق ويتم تطوير سيارة السباق عن طريق المكافآت التي حصل عليها من أدائه المهام، وعند أداء هذه المهام سوف يحصل على مكافأة مالية أو شيء آخر يعمل على تطويره في اللعبة، مما يعني ذلك أن الطالب سيعمل على محاولة إنجاز مهامه ليتمكن من الحصول على المكافآت التي ستعمل على تطوير سيارة السباق الخاصة به أو شراء سيارة جديدة بمواصفات أعلى ليتمكن من تحسين أدائه في السباق ومحاولة رفع نقاطه والحصول على نقاط أعلى، مما سيعمل على تحفيز الطالب على إنجاز مهامه و منافسة الطلاب الآخرين وهذا سيعمل على زيادة المنافسة الإنتاجية بين الطلاب.

## Table of Contents

|                                                     |           |
|-----------------------------------------------------|-----------|
| إهداء                                               | 2         |
| Abstract                                            | 3         |
| الملخص                                              | 4         |
| Table of Contents                                   | 5         |
| List of Tables                                      | 7         |
| List of Figures                                     | 8         |
| <b>Chapter 1: Introduction</b>                      | <b>10</b> |
| 1.1 Overview                                        | 11        |
| 1.2 Motivation and Importance                       | 11        |
| 1.3 Scope of the Project                            | 13        |
| 1.4 Objectives                                      | 13        |
| 1.5 Timeline/ Project Scheduling                    | 14        |
| 1.6 Project Description                             | 16        |
| 1.7 Expected Result                                 | 16        |
| 1.8 Overview of the Document                        | 16        |
| <b>Chapter 2: Background</b>                        | <b>17</b> |
| 2.1 Overview                                        | 18        |
| 2.2 Theoretical Background                          | 18        |
| 2.3 Literature Review                               | 21        |
| <b>Chapter 3: Requirements Specification</b>        | <b>23</b> |
| 3.1 Functional requirements                         | 24        |
| 3.1.1 Functional requirements as user requirement   | 24        |
| 3.1.2 Functional requirements as system requirement | 24        |
| 3.2 Non-functional requirements                     | 25        |
| 3.2.2 Software Quality Attributes                   | 25        |
| 3.3 Domain requirements                             | 25        |
| 3.4 Context Diagram                                 | 26        |
| 3.5 Use Case Diagram                                | 27        |
| 3.6 Functional Requirement Analysis                 | 28        |
| <b>Chapter 4: Software Design</b>                   | <b>34</b> |
| 4.1 Introduction                                    | 35        |
| 4.2 Architectural Design                            | 35        |
| 4.3 Class Diagram                                   | 37        |
| 4.4 Database Mapping                                | 38        |
| 4.5 Database Tables                                 | 39        |
| 4.6 Sequence Diagram                                | 43        |
| 4.7 User Interface                                  | 44        |
| 4.7.1 Theoretical                                   | 44        |
| 4.7.2 Page Design                                   | 45        |
| 1. Start Page                                       | 45        |

|                                          |           |
|------------------------------------------|-----------|
| 2. Sign up/ Sign in Page                 | 45        |
| 3. Car Showroom Page                     | 46        |
| 4. Tracks Page                           | 46        |
| 5. Race Track Type Page                  | 47        |
| 6. Tasks Page                            | 47        |
| 7. Profile Page                          | 48        |
| 8. Race Page                             | 48        |
| <b>Chapter 5: Software Demonstration</b> | <b>49</b> |
| 5.1 Introduction                         | 50        |
| 5.2 Technology And Tools Used            | 50        |
| 1. Unity Engine                          | 50        |
| 2. Unity Assets Store                    | 50        |
| 3. Blender                               | 51        |
| 4. C# Programming language               | 51        |
| 5. Visual Studio                         | 51        |
| 6. SQLite                                | 51        |
| 7. GitHub                                | 51        |
| 5.3 Implementation Details               | 52        |
| 5.3.1 User Authentication:               | 52        |
| 5.3.2 Task Management:                   | 52        |
| 5.3.3 Car Customization and Progression: | 52        |
| 5.3.4 Race Selection and Modes:          | 52        |
| 5.3.5 Rewards System:                    | 53        |
| 5.3.6 Game Mechanics:                    | 53        |
| 5.3.7 Track Design and Environment:      | 53        |
| 5.3.8 Race Selection and Modes:          | 53        |
| 5.3.9 Database                           | 53        |
| 5.3.9 Build Settings:                    | 54        |
| 5.3.10 Finish Line:                      | 55        |
| 5.3.11 Terrain Trees:                    | 55        |
| 5.3.12 Way Point Circuit:                | 56        |
| 5.4 Implementation Issue                 | 57        |
| 5.5 Game View                            | 57        |
| 5.5.1 Game Screen                        | 57        |
| 1. Start Page                            | 57        |
| 2. Sign Up/ Sign In Page                 | 58        |
| 3. Help Page                             | 58        |
| 4. Tasks OR Game Page                    | 59        |
| 5. Select Track Page                     | 59        |
| 6. Race Track Type Page                  | 60        |
| 7. During Racing                         | 60        |
| 8. Quit Menu Page                        | 61        |

|                                              |           |
|----------------------------------------------|-----------|
| 9. Race Results Screen                       | 61        |
| 10. Car Showroom Page                        | 62        |
| 11. Interlagos Track Screen                  | 62        |
| 12. Open World Track Screen                  | 63        |
| 13. Tasks Page                               | 63        |
| 14. Check The Task Page                      | 64        |
| 15. Claim Reward Page                        | 64        |
| 16. Profile Page                             | 65        |
| <b>Chapter 6: Testing</b>                    | <b>66</b> |
| 6.1 Introduction                             | 67        |
| 6.2 Functional Requirements Testing          | 67        |
| 6.2.1 Scenes Transformations                 | 67        |
| 6.2.2 Game Screen                            | 71        |
| 6.2.3 Result Screen                          | 72        |
| 6.2.4 Tasks Screen                           | 73        |
| 6.5 Sounds Testing                           | 74        |
| <b>Chapter 7: Conclusion and Future Work</b> | <b>75</b> |
| 7.1 Conclusion                               | 76        |
| 7.2 Future Work                              | 76        |
| <b>References</b>                            | <b>79</b> |

## List of Tables

### Chapter 1: Introduction

|                           |    |
|---------------------------|----|
| Table 1. 1: Project tasks | 14 |
| Table 1. 2: Gantt Chart   | 15 |

### Chapter 3: Requirements Specification

|                                 |    |
|---------------------------------|----|
| Table 3. 1: Start the game      | 28 |
| Table 3. 2: View profile        | 28 |
| Table 3. 3: Select Racing Car   | 29 |
| Table 3. 4: Develop Racing Car  | 29 |
| Table 3. 5: Select Racing Track | 30 |
| Table 3. 6: Select racing type  | 30 |
| Table 3. 7: Start the Race      | 31 |
| Table 3. 8: View Tasks          | 31 |
| Table 3. 9: Add Task            | 32 |
| Table 3. 10: Edit Task          | 32 |

|                            |    |
|----------------------------|----|
| Table 3. 11: Delete Task   | 33 |
| Table 3. 12: Exit the game | 33 |

## Chapter 4: Software Design

|                             |    |
|-----------------------------|----|
| Table 4. 1: Database Tables | 39 |
| Table 4. 2: Task Table      | 39 |
| Table 4. 3: Player Table    | 40 |
| Table 4. 4: Car Table       | 41 |
| Table 4. 5: Race Table      | 41 |
| Table 4. 6: Track Table     | 42 |

## Chapter 6: Testing

|                                              |    |
|----------------------------------------------|----|
| Table 6.1: Scenes Transformations Test Cases | 67 |
| Table 6.2: Tasks Or Game Page Test Cases     | 68 |
| Table 6.3: Select Race Track Page Test Cases | 69 |
| Table 6.4: Car Showroom Page Test Cases      | 70 |
| Table 6.5: Game Screen Test Cases            | 71 |
| Table 6.7: Result Screen Test Cases          | 72 |
| Table 6.8: Tasks Screen Test Cases           | 73 |
| Table 6.9: Sounds Test Cases                 | 74 |

## List of Figures

### Chapter 1: Introduction

|                                      |    |
|--------------------------------------|----|
| Figure 1. 1: Average Gaming Playtime | 11 |
| Figure 1. 2: Average Gaming Playtime | 12 |
| Figure 1. 3: Gaming Stats, by age    | 13 |

### Chapter 2: Background

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| Figure 2. 1: Usage Category                                                         | 20 |
| Figure 2. 2: The relevant differences in means, along with t-test numbers, p-values | 20 |

### Chapter 3: Requirements Specification

|                               |    |
|-------------------------------|----|
| Figure 3. 1: Context diagram  | 26 |
| Figure 3. 2: Use case diagram | 27 |

## Chapter 4: Software Design

|                                     |    |
|-------------------------------------|----|
| Figure 4. 1: MVC Design Pattern     | 36 |
| Figure 4. 2: Class Diagram          | 37 |
| Figure 4. 3: Database Mapping       | 38 |
| Figure 4. 4: Sequence diagram       | 43 |
| Figure 4. 5: Visible Light Spectrum | 44 |
| Figure 4. 6: Start Page             | 45 |
| Figure 4. 7: Sign-up/ Sign-in Page  | 45 |
| Figure 4. 8: Car Showroom Page      | 46 |
| Figure 4. 9: Tracks Page            | 46 |
| Figure 4. 10: Race Track Type Page  | 47 |
| Figure 4. 11: Tasks Page            | 47 |
| Figure 4. 12: Profile Page          | 48 |
| Figure 4. 13: Race Page             | 48 |

## Chapter 5: Software Demonstration

|                                      |    |
|--------------------------------------|----|
| Figure 5.1: Build Settings           | 54 |
| Figure 5. 2: Finish Line             | 55 |
| Figure 5.3: Terrain Trees            | 55 |
| Figure 5.4: Way Point Circuit        | 56 |
| Figure 5.5: Start Page               | 57 |
| Figure 5.6: Sign Up / Sign In Page   | 58 |
| Figure 5.7: Help Page                | 58 |
| Figure 5.8: Tasks OR Game Page       | 59 |
| Figure 5.9: Select Track Page        | 59 |
| Figure 5.10: Race Track Type Page    | 60 |
| Figure 5.11: During Racing Screen    | 60 |
| Figure 5.12: Quit Menu Page          | 61 |
| Figure 5.13: Race Results Screen     | 61 |
| Figure 5.14: Car Showroom Page       | 62 |
| Figure 5.15: Interlagos Track Screen | 62 |
| Figure 5.16: Open World Track Screen | 63 |
| Figure 5.17: Tasks Page              | 63 |
| Figure 5.18: Check The Task Page     | 64 |
| Figure 5.19: Claim Reward Page       | 64 |
| Figure 5.20: Profile Page            | 65 |

# Chapter 1: Introduction

1.1 Overview

1.2 Motivation and Importance

1.3 Scope of the Project

1.4 Objectives

1.5 Timeline/ Project Scheduling

1.6 Project Description

1.7 Expected Result

1.8 Overview of the Document

## 1.1 Overview

Our program is a motivational game that allows students to add the academic tasks they need to in order to continue their studies, such as studying for examinations, completing assigned projects, or finishing their homeworks. The student can create, alter, and delete tasks, setting the due date and time for the tasks, and after completing a predetermined number of tasks, he will receive a reward that will allow him to improve his performance in the race. The reward might be cash, a new racing car with better specifications, or a newly opened track.

The game will give players the option to select the race they want to participate in, as well as the track and the type of race they want to participate in, such as a normal or timed race. The student will try to accomplish his tasks in order to advance in the game since every time his points increase, he will receive a higher badge. As a result, the student will have effectively incorporated both his productive and enjoyable sides, increasing both his productivity and academic achievement.

## 1.2 Motivation and Importance

The majority of students have difficulty organizing their time so that they can't complete the tasks that are required of them, such as studying for exams, completing projects, or even just carrying out their duties. As a result, they turn to applications that help them organize these tasks, such as to-do lists, but these are quickly discarded because they do not encourage the student to continue and complete his tasks, and the student remains the same because he does not carry out the tasks that are required of him. Since video games have become very popular in recent years, as shown by the studies below, "Based on current data, the average gamer is spending 8.6 hours per week playing video games, which equates to roughly 1.2 hours per day



Figure 1. 1 : Average Gaming Playtime

If the average of 8.6 hours per week stays consistent throughout an entire year, the total time spent playing video games in one year would add up to **two and a half weeks**.

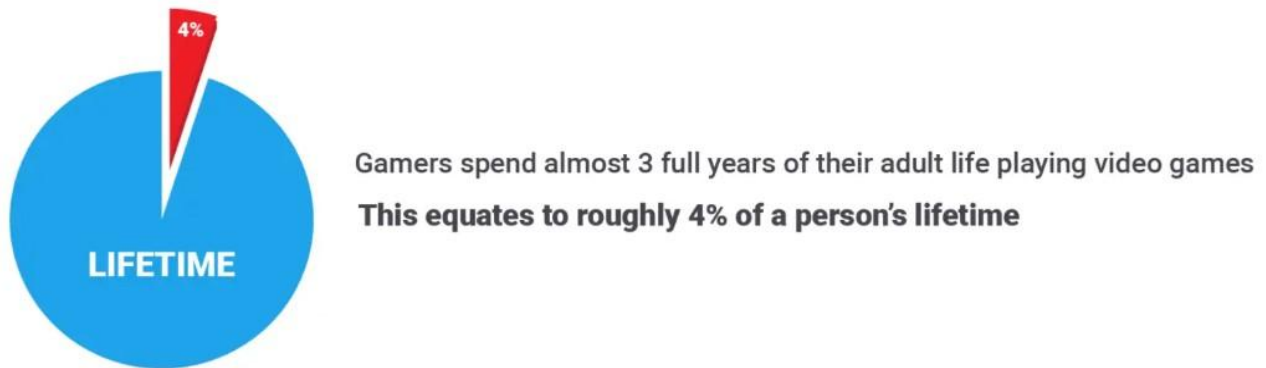


Figure 1. 2: Average Gaming Playtime

Assuming that the same numbers stayed consistent throughout someone's entire adult life, the total gaming playtime would equate to **2.8 years, which is 4% of a person's lifetime.**" [\[1\]](#)

As we saw in the studies, students who are addicted to games spend at least 1 to 3 hours playing them every day, which has a negative impact on their academic performance and their GPA. From this vantage point, many have attempted to develop educational games that make use of the gameplay style and encourage students to learn while having fun. Therefore, the motivation to use these games as a means of effectively improving academic performance and achievement without creating educational video games. By connecting tasks and completing them with the improvement of the racing cars in the game and gaining more points, one can rise above the other students' ranks. When the student attempts to improve his performance in the game, it is necessary for him to complete his academic assignments in order to receive the benefits that cannot be achieved in any other way.

## 1.3 Scope of the Project

The project is a motivational game that is intended for both male and female university students as statistics show that the most time playing on video games by the players who are his age between 18 and 34 as Figure 1. 3 shown, and the initial platform is for Windows PCs.

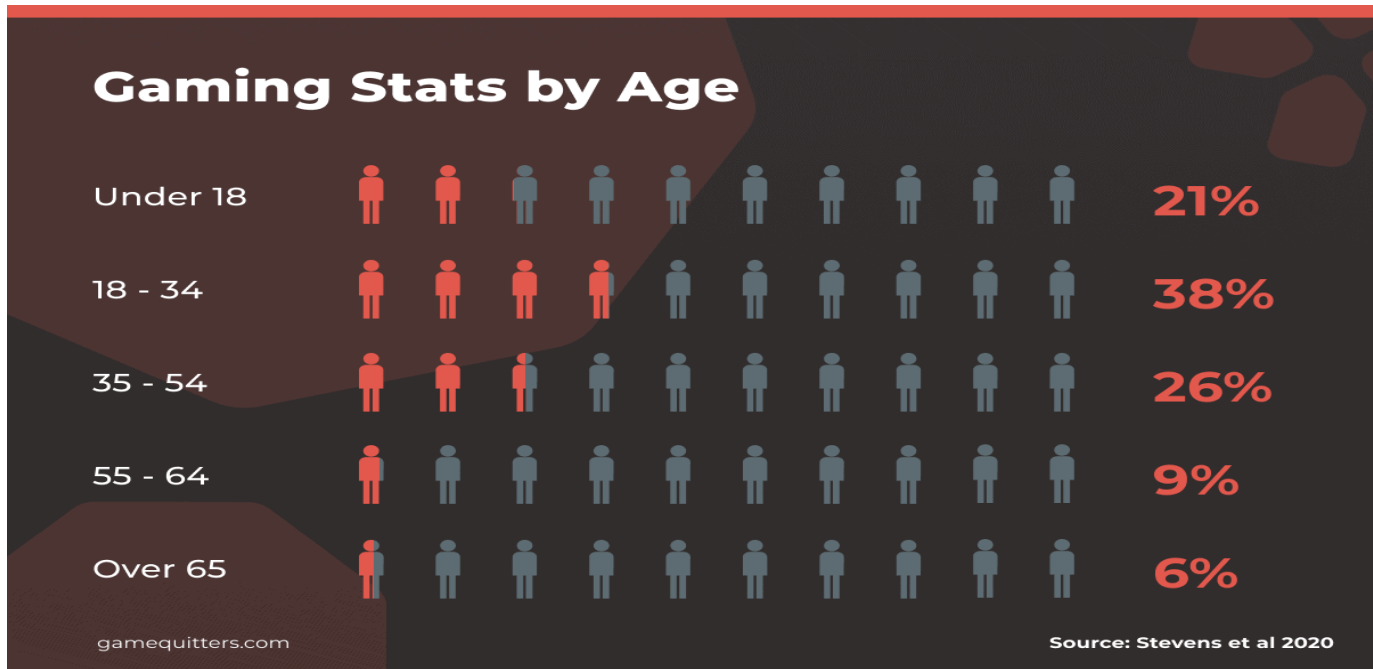


Figure 1. 3: Gaming Stats, by age [\[2\]](#)

## 1.4 Objectives

1. Developing a motivation game.
2. Encouraging students to finish their tasks so they can progress in the game.
3. Encouraging students to complete their tasks on time.
4. Increasing academic results and aiming for a higher academic rate.

## 1.5 Timeline/ Project Scheduling

We will follow the software development life cycle to achieve the objectives. And as shown in Table 1.1, these are the tasks that we have to do. Also, table 1.2 shows the number of weeks required to achieve each task.

Table 1. 1: Project tasks

| Task number | Task name                               | The time required in weeks      |
|-------------|-----------------------------------------|---------------------------------|
| <b>1</b>    | System definition and planning          | <b>5</b>                        |
| <b>2</b>    | Determine the Project requirements      | <b>4</b>                        |
| <b>3</b>    | Description of the project Requirements | <b>4</b>                        |
| <b>4</b>    | System design                           | <b>4</b>                        |
| <b>5</b>    | System development and programming      | <b>6</b>                        |
| <b>6</b>    | Integration and system testing          | <b>3</b>                        |
| <b>7</b>    | Documenting the website                 | <b>Along the working period</b> |

Table 1. 2: Gantt Chart

|                                         | First semester |      |       |  | Second semester |      |       |
|-----------------------------------------|----------------|------|-------|--|-----------------|------|-------|
| Weeks                                   | 2-6            | 7-10 | 11-14 |  | 2-6             | 7-11 | 12-14 |
| System definition and planning          |                |      |       |  |                 |      |       |
|                                         |                |      |       |  |                 |      |       |
| Determine the Project Requirements      |                |      |       |  |                 |      |       |
|                                         |                |      |       |  |                 |      |       |
| Description of the project Requirements |                |      |       |  |                 |      |       |
|                                         |                |      |       |  |                 |      |       |
| System design                           |                |      |       |  |                 |      |       |
|                                         |                |      |       |  |                 |      |       |
| System development and programing       |                |      |       |  |                 |      |       |
|                                         |                |      |       |  |                 |      |       |
| Integration and system testing          |                |      |       |  |                 |      |       |
|                                         |                |      |       |  |                 |      |       |
| System documentation                    |                |      |       |  |                 |      |       |
|                                         |                |      |       |  |                 |      |       |

Table Key

|  |                                  |
|--|----------------------------------|
|  | Estimate time to finish the task |
|  | Real time to finish the task     |
|  | Holiday between semesters        |

## 1.6 Project Description

The game's user interface will allow students to choose the tasks they want to perform and how long it will take them to finish them. If they finish the tasks on time, they will receive a reward that will encourage them to keep going.

The programming will have two main sections. The first section will involve adding, modifying, and deleting tasks with a specified deadline for completing them. and the student will get more points when he completes a specific number of tasks, raising his total points in the race. Upon reaching a specified number of points, the player will get a specific badge, where there will be many badges in the game that determine the player's ranking among other players in terms of the number of tasks that have been completed and the points he has obtained; the badge may be bronze, silver, or gold, and with each specific number of tasks that are done, the player will get a reward that he can use to develop his car, open new cars, or open a new track for racing, and every time his points increase and he gets an advanced badge, the difficulty of the race will vary without the need for the player to select the level of difficulty.

The racing game will be in the second section. There are two racing modes in the game: the first is a timed race, while the second is an AI-car race. There are initially three cars in the game, but only one of them will be accessible. As you progress in the game and earn more points, the other two cars will become available. There are three different tracks available to choose from in the race.

To develop the game, we'll use the C# programming language and the Unity engine.

## 1.7 Expected Result

A game that motivates students to complete their tasks that they must do to improve their academic performance and try to get a higher GPA.

## 1.8 Overview of the Document

In chapter one, we reviewed the problems facing college students today. Also, we gave an overview and a short description of the project. In chapter two, we will describe the theoretical background of the project and related projects. In chapter three, we will discuss requirements and specifications. In chapter four, we will show the UI, the first design for the game, and both the sequence and class diagram.

## Chapter 2: Background

2.1 Overview

2.2 Theoretical Background

2.3 Literature Review

## 2.1 Overview

This chapter briefly describes the theoretical background of the project and related projects.

## 2.2 Theoretical Background

Video games are an interactive entertainment technology that may be played on a variety of platforms, such as smartphones, XBox devices, and virtual reality headsets. [3]

Children, teenagers, and young people now spend their lives mostly through playing electronic video games. Because of their popularity and ability for enjoyment, the games, which were based on attractive animations, were really successful. The games have improved to the point that they gave the user a sense of actually playing and interacting on the ground. [4]

The first video game, known as the space computer, was developed and proved to be commercially successful in 1971. This marked the beginning of a new entertainment sector that would later grow in the late 1970s in the United States, Japan, and Europe.

Due to the release of numerous inexpensive, low-quality games like Atari games, the American video game industry then experienced a significant decline that resulted in the financial collapse of many companies in 1983. This decline lasted for at least two years before a Japanese company called Nintendo arrived to produce games from 1989, such as playing cards, the Super Mario game is an example of these games. a year that continues to be remembered as a turning point in the history of video gaming.

Games have advantages and disadvantages, just like any form of entertainment. The following are some advantages of video games:

1. An enjoyable form of entertainment that satisfies certain emotional requirements that cannot be satisfied in any other way.
2. By taking part in group play, participants can improve social communication and strengthen the roots of affection and love.
3. Children's intelligence and focus are increased and developed by video games.

The following are some disadvantages of video games:

1. Wasting time and spending a lot of time sitting down each day, which has an impact on academic performance and daily task completion
2. There could rarely be a rise in anxiety, stress, and emotional instability.
3. Social isolation and a distance from reality brought on by a game addiction [5]

And as shown in the negatives of video games, addiction was one of the biggest issues that most players faced because they lost themselves in the games and wasted a lot of time. This led to the development of bad habits that did not previously exist, such as lateness, procrastination, isolation, and many other things.

What, therefore, is addictive, and why do the majority of players become addicted to these games?

Addiction occurs when a behaviour or habit entirely consumes you, negatively affecting everyday functioning and producing serious issues with mental, social, or physical health. People have recently developed a strong addiction to the ambition to win in the games they play, and they anxiously await the chance to constantly work to increase their score and unlock higher badges.

But what is the root of this extreme game addiction? And how this sometimes causes individuals to spend lengthy periods of time combining the night and the day without even considering their need for rest or other necessities.

Scientists have not yet been able to pinpoint the primary cause of this addiction, but they have hypothesised that it may be brought on by the need to sate certain emotional needs by winning and receiving rewards in the game. It satisfies desires that are impossible to fulfil in reality, working the brain to release the dopamine hormone, which leads to extreme happiness and the onset of addiction to a particular behaviour. Neuroscientists have discovered that the minds of drug addicts and video game addicts share many similarities.[6]

Many studies have found a strong correlation between playing video games and low academic rates(GPA) and a close association with bad habits of wasting time, procrastination, and poor time management, as shown. This issue has specifically impacted students and their academic achievement as a result of the addiction that many people who play video games experience. According to the findings of the following study, which included 143 university students whose information was gathered.

|                  | Usage Category |               | Total |
|------------------|----------------|---------------|-------|
|                  | High           | Low           |       |
| Users            | 50             | 93            | 143   |
| Video game usage | > 13.38 hours  | < 13.38 hours |       |

Figure 2. 1: Usage Category [2]

|                                                 | Low Usage<br>(Mean) | High<br>Usage<br>(Mean) | T-test | p-value | Sig. |
|-------------------------------------------------|---------------------|-------------------------|--------|---------|------|
| <b>** p &lt; .05 * p &lt; .10</b>               |                     |                         |        |         |      |
| <b><i>Current college GPA</i></b>               | 3.41                | 3.22                    | 2.04   | 0.043   | **   |
| <b><i>Grades Negatively Impacted</i></b>        | 1.61                | 1.88                    | -1.67  | 0.097   | *    |
| <b><i>Skiping Class</i></b>                     | 1.11                | 1.16                    | -0.61  | 0.541   |      |
| <b><i>Lose Sleep</i></b>                        | 1.82                | 2.50                    | -3.82  | 0.000   | **   |
| <b><i>Trouble Paying Attention in Class</i></b> | 2.28                | 2.60                    | -1.70  | 0.091   | *    |
| <b><i>Hours Spent Studying in a Week</i></b>    | 10.97               | 13.46                   | -1.30  | 0.196   |      |

Figure 2. 2: The relevant differences in means, along with t-test numbers, p-values [2]

As we previously discussed, a dependence on video games leads to undesirable habits including time wastage, laziness, and delay. Studies have also shown a strong link between electronic game addiction and inferior academic performance.

When students were in school, their parents were there to support them academically and assist them improve their grades. As students advance and transfer to the university level, however, their sense of responsibility for their academic success decreases. According to data gathered from 143 university students, there is a strong association between spending a lot of time playing video games and academic success. When the hours were longer than usual, report shows that this issue contributed to students skipping classes and failing to show up for them, which means that the student misses his lectures in order to play video games, and that their sleep duration decreased as a result of their addiction to them, which resulted in As evidenced by the numbers above, concentration was spread throughout the study, which made it hard to become addicted to video games. a factor that is underrated. Rather, it is a real issue that teenage people of this age are dealing with, one that causes them to lose sight of their goals, become addicted to video games, and isolate themselves from society while believing that their lives are a simulation of reality. As a result, they are content with achieving fake accomplishments and rising through the ranks in the world of games. [7]

## 2.3 Literature Review

### **Microsoft To-Do.**

#### **Description:**

Is a kind of app that is generally used to maintain our day-to-day tasks or list everything that we have to do, with the most important tasks at the top of the list and the least important tasks at the bottom. It is helpful in planning our daily schedules. We can add more tasks at any time and delete a task that is completed. [8]

#### **Features:**

The application allows for the specification of time, date, task class, addition, modification, and deletion of tasks, as well as the ability to monitor all tasks necessary for the day that the tasks have been established and send reminder messages to complete the tasks.

#### **Limitations:**

Not linked to Apple Calendar, file storage on the To Do list is constrained (that is, you cannot keep the same files in To Do that you had for projects with Wunderlist). Therefore, you must keep those files in other folders other than To Do. Furthermore, the application does not encourage the user to complete his tasks.

## **Habitica.**

### **Description:**

“Habitica is a free habit-building and productivity app that treats your real life like a game. With in-game rewards and punishments to motivate you and a strong social network to inspire you, Habitica can help you achieve your goals to become healthy, hard-working, and happy”. [9]

### **Features:**

The application aims to add, modify, and delete tasks and see the tasks that need to be done with rewards when completing tasks, as when completing tasks you get rewards that enable you to develop the character that represents you and buy clothes for her as well as raise your points to compete with friends and get gold that you can use to purchase things. The application enables the establishment of groups in which individuals compete with each particular number of activities that are completed, starting with purchasing a lot of the things you want to buy. The competition then occurs amongst friends by fulfilling the tasks.

### **Limitations:**

There is no actual game in which you can use your character and really compete, such as a fighting game between characters, and win real prizes as a result of the challenge in the game. Instead, the application aims to let you develop your character within the application, compete with points, and complete tasks. The only things you can do with the app are complete activities, get points, and compete for views and points.

# Chapter 3: Requirements Specification

## 3.1 Functional requirements

### 3.1.1 Functional requirements as user requirement

### 3.1.2 Functional requirements as system requirement

## 3.2 Non-functional requirements

### 3.2.1 Performance Requirements

### 3.2.2 Software Quality Attributes

## 3.3 Domain requirements

## 3.4 Context Diagram

## 3.5 Use Case Diagram

## 3.6 Functional Requirement Analysis

## 3.1 Functional requirements

The functional requirements were identified in the early stages of the project's development and include the following:

### 3.1.1 Functional requirements as user requirement

- The player shall be able to start the game.
- The player shall be able to exit the game and return to home.
- The tasks must be creatable and editable by the player.
- The player must be able to estimate how long it will take to complete the task.
- The player will be able to check how many tasks he has finished.
- The player will have the option of choosing the car to race.
- The player must be able to identify the race track.
- The player can select whether the race will be against the time or not.
- The player can view his profile to see how many races he's won and lost, how many cars he owns, and how many tasks he's finished and still needs to accomplish.
- The car can be developed and customised by the player.
- The player will be able to check his position on a race map during the race.
- The player shall be able to view the highest score in the game.
- The player will be allowed to play again after the game is over.

### 3.1.2 Functional requirements as system requirement

- The race's cars and tracks will be shown in the game.
- The game will show the tasks screen so that the player can add and update his tasks.
- When a player reaches the finish line, the game will end.
- When a certain number of tasks have been finished, the game gives rewards.

## 3.2 Non-functional requirements

Regarding non-functional requirements, our work focused on the following:

### 3.2.1 Performance Requirements

- The user should not have to wait too long for the car racing game to begin and for the car to appear.
- The system must be operational 7 days a week, 365 days a year.
- Response time must not exceed 4 seconds.
- Login must be completed in less than 3 seconds.

### 3.2.2 Software Quality Attributes

- **Ease of Use:** The game interfaces are straightforward and user-friendly, and switching between them is simple. The student can add tasks and edit them, develop his own racing car, and start the race with simplicity.
- **Portability:** All users who have windows PCs can use this application.
- **Compatibility:** The application should run on a windows operating system efficiently.
- **Flexibility:** The system is simple to modify and improve when necessary.
- **Ease of Learning:** The game can be learned in just a few easy steps according to the system's straightforward and simple running concept. The system will contain a feature that will allow it to read the game's instructions, add tasks, and modify them into the task interface.
- **Responsiveness:** When the car hits another car or a certain obstacle, causing the car's speed to reduce and its direction to shift, the game responds quickly with impact sounds that are provided within 60 milliseconds of the crash. The race map also allows the player to see where he is in comparison to other players.

## 3.3 Domain requirements

The game application runs on Windows PCs, and the student needs access to the Internet in order to start the application, add tasks, and play.

## 3.4 Context Diagram

Figure 3.1 shows the components of our system and the interaction between them.

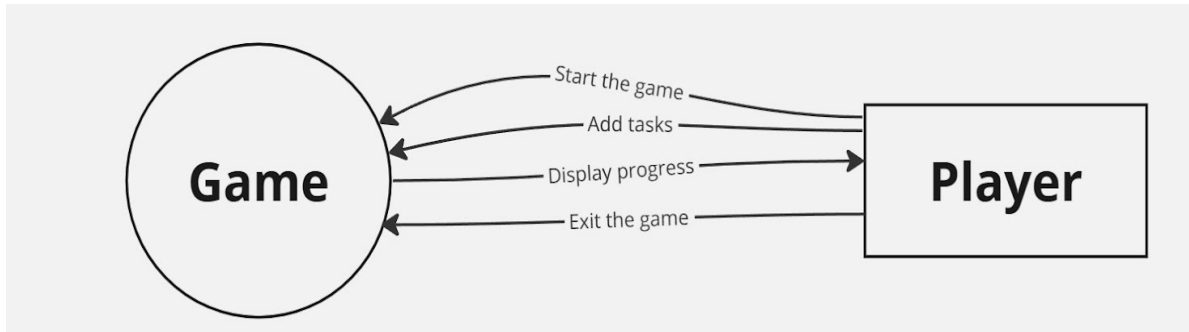


Figure 3. 1: Context diagram

### 3.5 Use Case Diagram

The interactions between the system and its actors, in this case, the player, are depicted in Figure 3.2. So, it demonstrates how the player interacts with and uses the system.

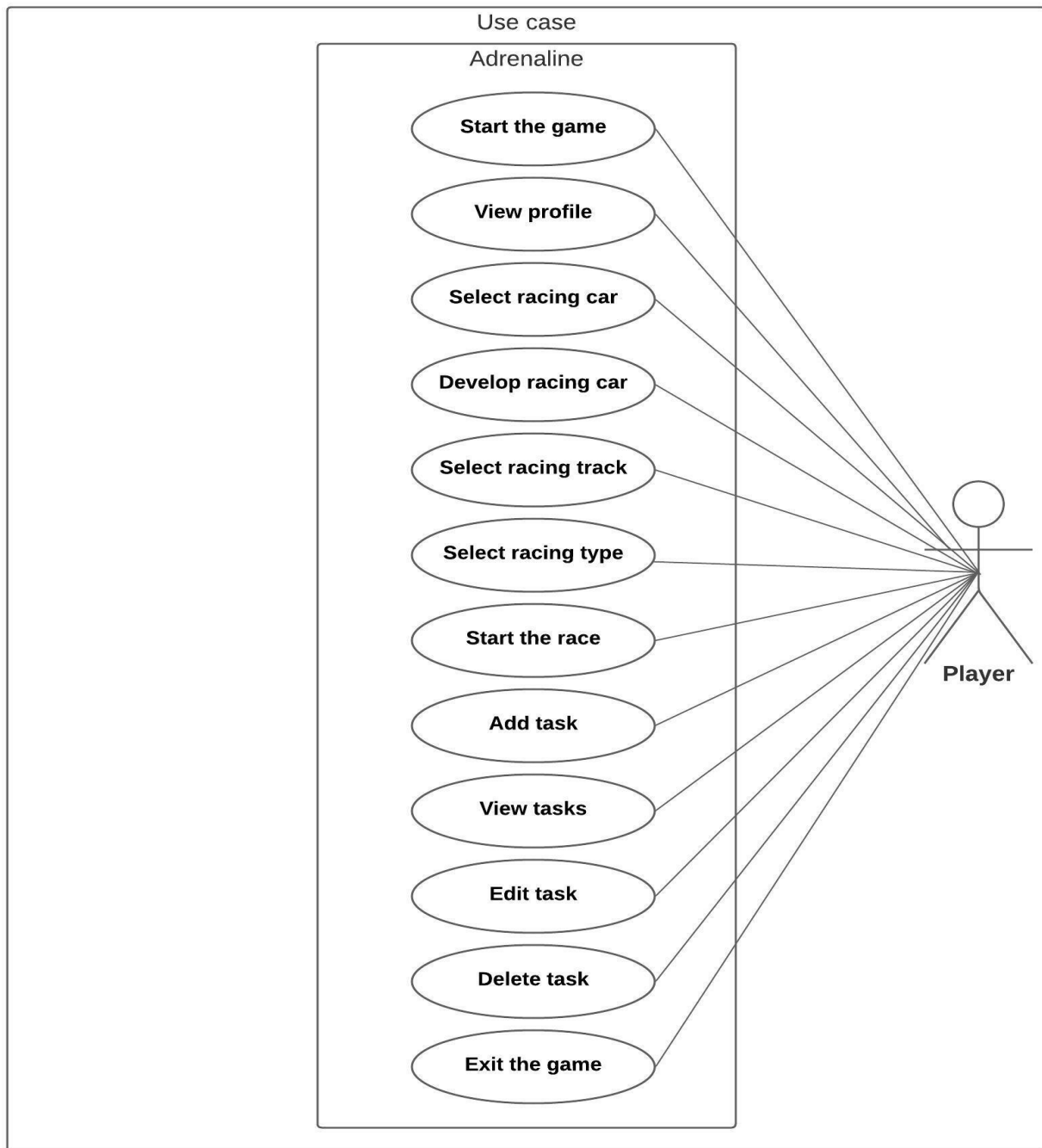


Figure 3. 2: Use case diagram

## 3.6 Functional Requirement Analysis

Table 3.1 shows the details of starting the game functionality.

Table 3. 1: Start the game

| Use Case      | Start the game                                                                                       |
|---------------|------------------------------------------------------------------------------------------------------|
| Goal          | Start the race and go to car showroom page to choose a racing car                                    |
| Preconditions | To be at the main start page                                                                         |
| Scenario      | When a player clicks the game's start button, a car showroom page for choosing a racing car appears. |
| Exceptions    | Internet malfunction                                                                                 |

Table 3.2 shows the details of the viewing profile functionality.

Table 3. 2: View profile

| Use Case      | View profile                                                                                                                                                                         |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Goal          | Go to the player profile page to view his personal information                                                                                                                       |
| Preconditions | The player should be at the one of these pages : car showroom, tracks page, racing type page, or tasks page<br>profile picture, two options appear one of                            |
| Scenario      | When the player clicks his profile picture, two options appear; one of them is the profile button, and when the player clicks the profile button, a player profile page will appear. |
| Exceptions    | Internet malfunction                                                                                                                                                                 |

Table 3.3 shows the details of selecting racing car functionality.

Table 3. 3: Select Racing Car

| Use Case      | Select racing car                                                                                                                                                                                                                                                |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Goal          | The player choose the racing car to use it in the race                                                                                                                                                                                                           |
| Preconditions | The player should be at the start page                                                                                                                                                                                                                           |
| Scenario      | When the player has already signed in the game and after the game starts, the car showroom page appears and a list of available cars will appear to choose a racing car for the race, and we have in the page options to go to the tracks page or buy a new cars |
| Exceptions    | Internet malfunction                                                                                                                                                                                                                                             |

Table 3.4 shows the details of developing racing car functionality.

Table 3. 4: Develop Racing Car

| Use Case      | Develop racing car                                                                                                                                                                             |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Goal          | To develop racing cars in order to achieve better results in the race                                                                                                                          |
| Preconditions | The user should be at the car's showroom page                                                                                                                                                  |
| Scenario      | When the develop button is clicked on the car showroom page, we can observe the speed, handling, and braking capabilities of each car. If the player has enough money, he can develop his car. |
| Exceptions    | Internet malfunction                                                                                                                                                                           |

Table 3.5 shows the details of selecting racing track functionality.

Table 3. 5: Select Racing Track

| Use Case      | Select racing track                                                                                                                                      |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Goal          | The player selects the racetrack he wants to use.                                                                                                        |
| Preconditions | The user should be at the car showroom page                                                                                                              |
| Scenario      | When the player clicks the Race Tracks button, a list of available tracks will appear on the Racing Tracks page, allowing him to choose the one he wants |
| Exceptions    | Internet malfunction                                                                                                                                     |

Table 3.6 shows the details of selecting racing-type functionality.

Table 3. 6: Select racing type

| Use Case      | Select racing type                                                                                                                                                                                                      |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Goal          | To determine the type of the race                                                                                                                                                                                       |
| Preconditions | The player should be at the racing track's page                                                                                                                                                                         |
| Scenario      | When the player choose the track, the racing track type page appears that allowed the player to specify the type of the race (against the time or normal race against AI cars), number of laps, and number of opponents |
| Exceptions    | Internet malfunction                                                                                                                                                                                                    |

Table 3.7 shows the details of the starting race functionality.

Table 3. 7: Start the Race

| Use Case      | Start race                                                                                                |
|---------------|-----------------------------------------------------------------------------------------------------------|
| Goal          | Go to the race page and start racing                                                                      |
| Preconditions | The player should be at the race track type page                                                          |
| Scenario      | When the player clicks the begin race button, the race page will appear and the player can start the race |
| Exceptions    | Internet malfunction                                                                                      |

Table 3.8 shows the details of viewing tasks functionality.

Table 3. 8: View Tasks

| Use Case      | View tasks                                                                                                                    |
|---------------|-------------------------------------------------------------------------------------------------------------------------------|
| Goal          | Go to tasks page to view tasks, check task, add new task, edit task, or delete task                                           |
| Preconditions | The player should be at the car showroom page                                                                                 |
| Scenario      | When the player clicks the Add Task button, the tasks page will appear to check task, add new task, edit task, or delete task |
| Exceptions    | Internet malfunction                                                                                                          |

Table 3.9 shows the details of adding task functionality.

Table 3. 9: Add Task

| Use Case      | Add task                                                                                                                                              |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Goal          | To add a new task to his list of tasks to attempt to complete them                                                                                    |
| Preconditions | The player should be at the task's page                                                                                                               |
| Scenario      | When the player clicks the adding task button, He can write the task he wants to complete and determine the date and the expected time to complete it |
| Exceptions    | Internet malfunction                                                                                                                                  |

Table 3.10 shows the details of editing task functionality.

Table 3. 10: Edit Task

| Use Case      | Edit task                                                                                                                   |
|---------------|-----------------------------------------------------------------------------------------------------------------------------|
| Goal          | To edit tasks from his list of tasks                                                                                        |
| Preconditions | The player should be at the task's page                                                                                     |
| Scenario      | When the player clicks the edit task button, He can edit the task he wants to complete or the expected time to complete it. |
| Exceptions    | Internet malfunction                                                                                                        |

Table 3.11 shows the details of deleting task functionality.

Table 3. 11: Delete Task

| Use Case      | Delete task                                                                                  |
|---------------|----------------------------------------------------------------------------------------------|
| Goal          | To delete tasks from his list of tasks                                                       |
| Preconditions | The player should be at the task's page                                                      |
| Scenario      | When the player clicks the delete task button, He can delete the task from his list of tasks |
| Exceptions    | Internet malfunction                                                                         |

Table 3.12 shows the details of exiting the game functionality.

Table 3. 12: Exit the game

| Use Case      | Exit the game                                                                                                                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Goal          | Ending the game                                                                                                                                                                            |
| Preconditions | The player should be at the one of these pages : car showroom, tracks page, racing type page, or tasks page                                                                                |
| Scenario      | When the player clicks on the profile picture two options will appear, one of them is exit button, and when the player clicks exit button, he will taken him back to sign-up/ sign-in page |
| Exceptions    | Internet malfunction                                                                                                                                                                       |

## Chapter 4: Software Design

### 4.1 Introduction

### 4.2 Architectural Design

### 4.3 Class Diagram

### 4.4 Database Mapping

### 4.5 Database Tables

### 4.6 Sequence Diagram

### 4.7 User Interface

#### 4.7.1 Theoretical

#### 4.7.2 Page Design

1. Start Page
2. Sign up/ Sign in Page
3. Car Showroom Page
4. Tracks Page
5. Race Track Type Page
6. Tasks Page
7. Profile Page
8. Race Page

## 4.1 Introduction

This chapter includes the following: a class diagram, game architecture, a sequence diagram, as well as the graphical interfaces. with an explanation for each.

## 4.2 Architectural Design

Adrenaline: It's a motivational game that combines the ability to identify the tasks that must be completed to enhance academically, such as completing projects and assignments or even studying for exams, with the ability to play and improve performance in the game using the rewards obtained when completing tasks and trying to raise points to get a higher ranking and a higher badge. between the other competitors.

To design the game, we have offered the following patterns: Strategy, Observer, Composite Design Pattern.

The MVC pattern, however, was selected for the game design since it was discovered that it integrates the three patterns that had previously been proposed, for instance, in the following order: The game logic must receive the same kind of input regardless of the input controller (button, gesture, joystick), and Strategy Design Pattern works automatically to give that without interfering with the game. This interaction between the input control unit and the game logic is represented by the control part in MVC, which is called a (controller), and Observer Design Pattern links the classes together without requiring knowledge of how each one of them operates, because communication and interaction between the classes are what matters. One of these illustrates how messages may be delivered between objects without either party needing to know anything about the other, and thus illustrates the game's (model's) logic in MVC, The Composite Design Pattern represents all views (window and main buttons) in the game as this design pattern provides a unified access point for all views of the form, working to provide a single point of access for each view rather than having a different function to access each view and this represent a (view) part in Model-View-Controller Design Pattern.

The three parts of the MVC software-design pattern can be described as follows:

- 1- Model: Manages data and business logic.
- 2- View: Handles layout and display.
- 3- Controller: Routes commands to the model and view parts.

## Model-View-Controller Design Pattern

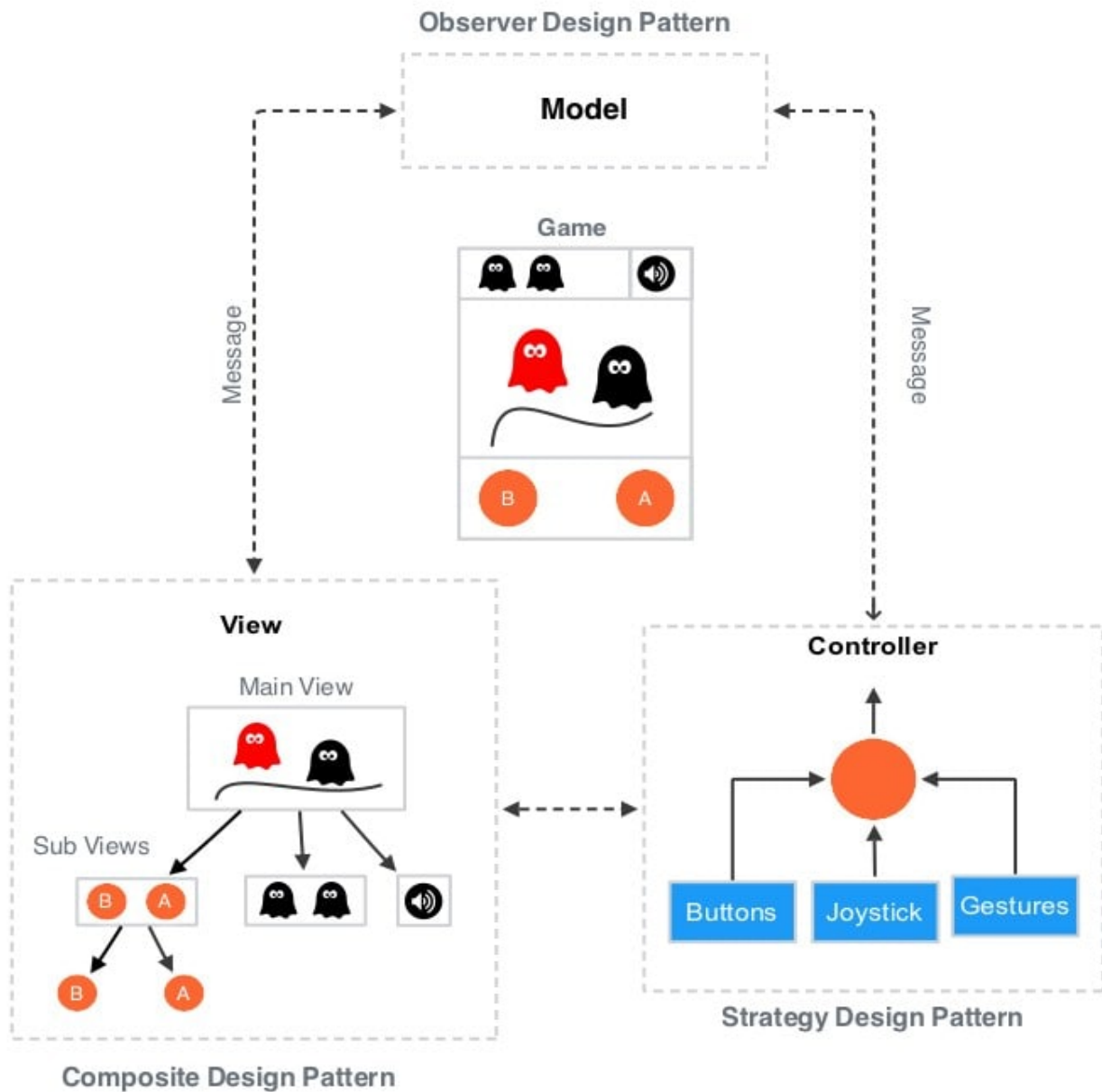


Figure 4. 1: MVC Design Pattern [\[10\]](#)

As shown in the illustration above, the Strategy pattern represents the Controller part of the MVC, the Observer pattern represents the Model part of the MVC, and the Composite pattern represents the View part of the MVC.

## 4.3 Class Diagram

Figure 4.2 shows the main classes in our project:

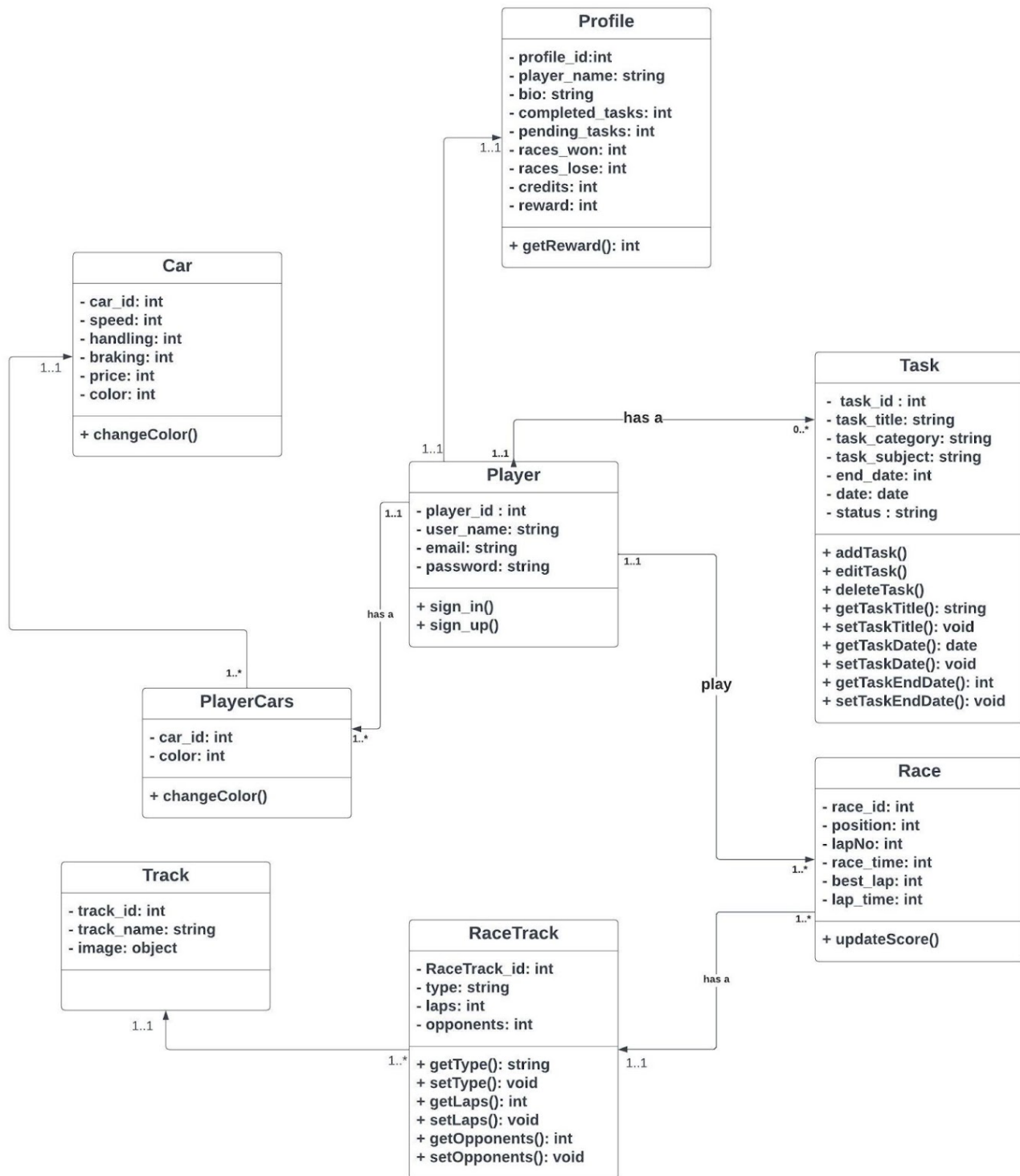


Figure 4. 2: Class Diagram

## 4.4 Database Mapping

Figure 4.3 shows the database mapping:

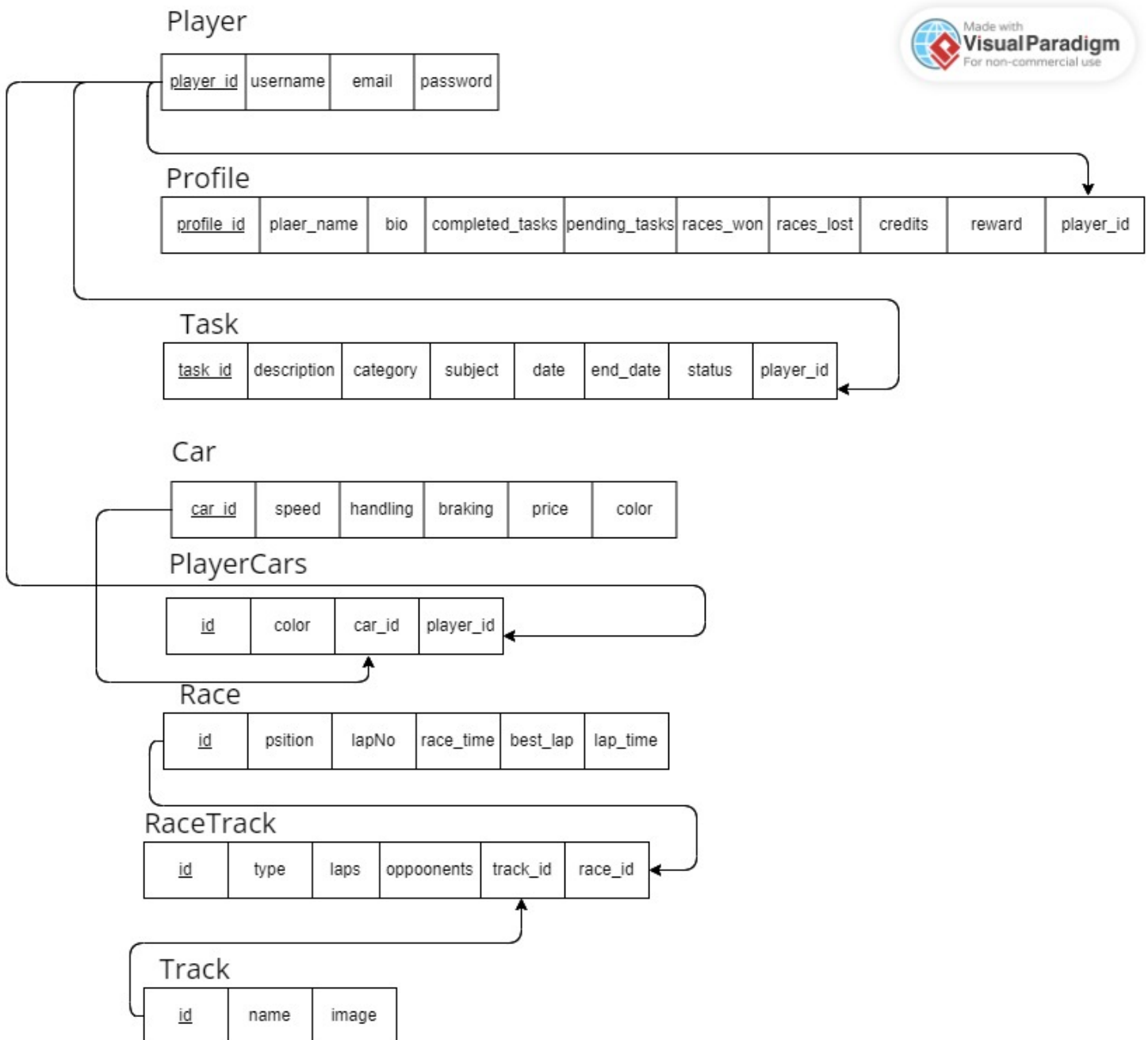


Figure 4. 3: Database Mapping

## 4.5 Database Tables

Table 4. 1: Database tables

| Table  | Table name in database | Table description                   |
|--------|------------------------|-------------------------------------|
| Task   | Task                   | Store information about tasks       |
| Player | Player                 | Store players' information          |
| Car    | Car                    | Store information about racing cars |
| Race   | Race                   | Store information about each race   |
| Track  | Track                  | Store information about tracks      |

### 1. Task Table

Table 4. 3: Task table

| Field name    | Field type | Null | Field length | Field description             |
|---------------|------------|------|--------------|-------------------------------|
| task_id       | int        | No   | 50           | Primary key                   |
| task_title    | Varchar    | No   | 500          | The title of the task         |
| task_category | Varchar    | Yes  | 100          | The category of the task      |
| end_time      | int        | No   | 100          | End time of the task          |
| date          | date       | No   | 100          | The date when the task added  |
| done_task     | bool       | No   | 100          | Check if the task done or not |

## 2. Player Table

Table 4. 2: Player table

| Field name      | Field type | Null | Field length | Field description                                    |
|-----------------|------------|------|--------------|------------------------------------------------------|
| player_id       | int        | No   | 50           | Primary key                                          |
| username        | Varchar    | No   | 100          | The player's username (varies from player to player) |
| password        | Varchar    | No   | 100          | The player's password                                |
| player_name     | Varchar    | No   | 100          | The player's Full name                               |
| bio             | Varchar    | Yes  | 500          | Bio for player profile                               |
| image           | Custom     | Yes  | 500          | Image for player profile                             |
| number_of_cars  | int        | No   | 50           | The number of cars the player owns                   |
| completed_tasks | int        | No   | 50           | The number of tasks the player done                  |
| pending_tasks   | int        | No   | 50           | The number of tasks the player want to done it       |
| badge           | int        | No   | 100          | The badge of the player in the game                  |
| races_won       | int        | No   | 50           | The number of races the player won                   |
| races_lose      | int        | No   | 50           | The number of races the player lose                  |
| score           | int        | No   | 200          | The score of the player                              |
| coins           | int        | No   | 200          | The coins of the player                              |
| reward          | int        | No   | 200          | The player rewards from completed tasks              |

### 3. Car Table

Table 4. 4: Car table

| Field name    | Field type | Null | Field length | Field description                                     |
|---------------|------------|------|--------------|-------------------------------------------------------|
| car_id        | int        | No   | 50           | Primary key                                           |
| speed         | int        | No   | 200          | The car race speed                                    |
| handling      | int        | No   | 200          | The car race handling                                 |
| braking       | int        | No   | 200          | The car race braking                                  |
| price         | int        | No   | 200          | The car race price                                    |
| colour        | int        | No   | 50           | The car race colour                                   |
| car_available | bool       | No   | 50           | Check if this car race available for player to buy it |

### 4. Race Table

Table 4. 5: Race table

| Field name | Field type | Null | Field length | Field description                                        |
|------------|------------|------|--------------|----------------------------------------------------------|
| race_id    | int        | No   | 50           | Primary key                                              |
| position   | int        | No   | 50           | The position of the player in the race                   |
| lapNo      | int        | No   | 50           | The number of the lap on which the player is in the race |
| race_time  | int        | No   | 100          | The time spent by the player in the race                 |
| best_lap   | int        | No   | 50           | The best lap for the player in the race                  |
| lap_time   | int        | No   | 200          | The time spent by the player in the lap                  |

## 5. Track Table

Table 4. 6: Track table

| Field name | Field type | Null | Field length | Field description               |
|------------|------------|------|--------------|---------------------------------|
| track_id   | int        | No   | 50           | Primary key                     |
| track_name | Varchar    | No   | 100          | The name of the track           |
| image      | Custom     | No   | 500          | The image of the track          |
| type       | Varchar    | No   | 100          | The type of the race track      |
| laps       | int        | No   | 50           | Number of race laps             |
| opponents  | int        | No   | 50           | Number of opponents in the race |

## 4.6 Sequence Diagram

The diagram in Figure 4.4 shows the interaction between the player and rewards class and shows how the player will claim his rewards.

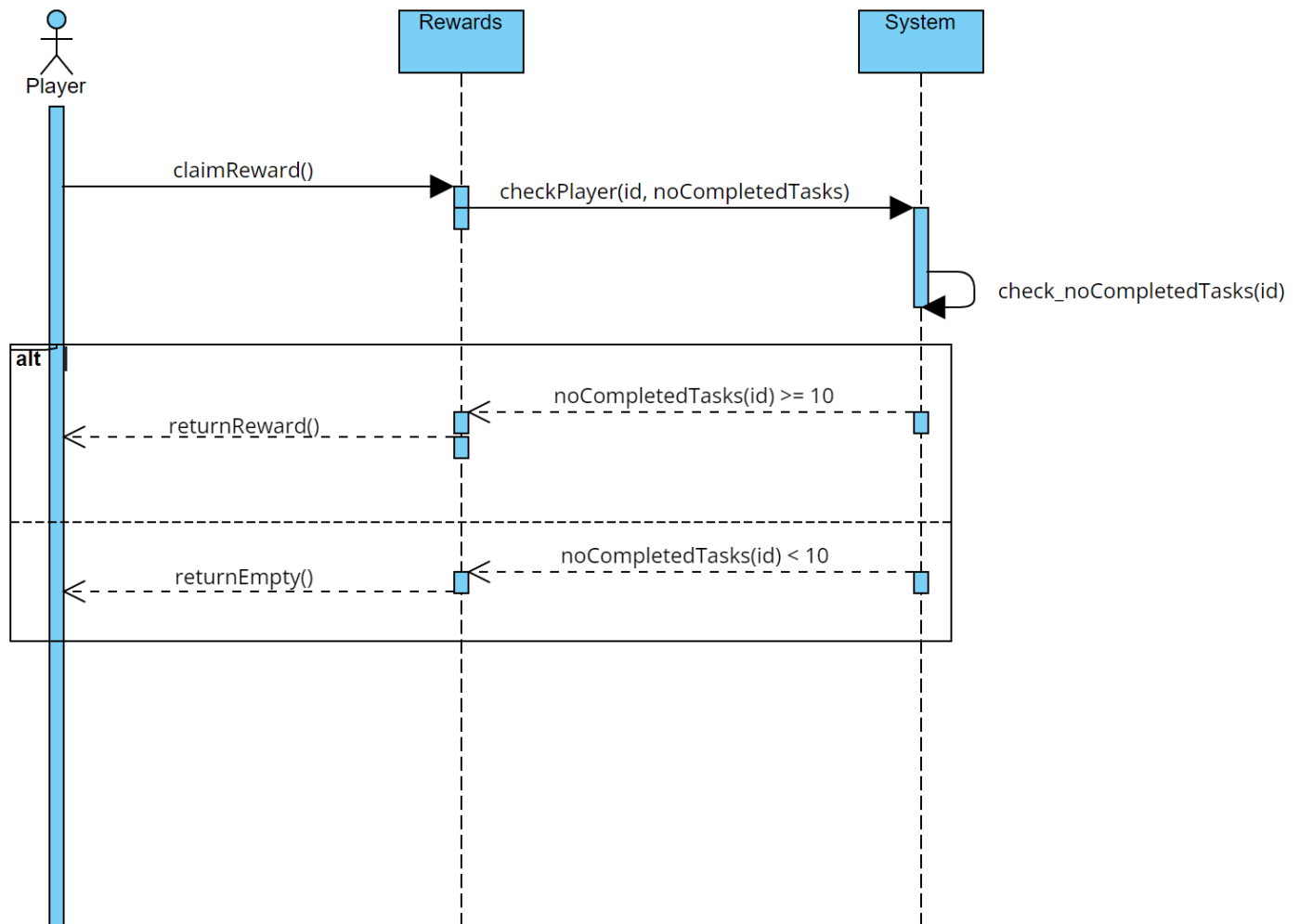


Figure 4. 4 : Sequence diagram

## 4.7 User Interface

### 4.7.1 Theoretical

Colours are essential to design and daily life. Whether or not we like a particular colour, it will affect how we feel when we view it. Colours are connected to memories and ideas as well. Colours are an effective means of communication that may be employed to denote action, affect moods, and even influence reactions. Designers have long felt that colour has a significant impact on our feelings, emotions, and decision-making.

Because blue is a colour that many people prefer, it was picked from the colour wheel to build the game. It is frequently regarded as a non-threatening colour and can have a conventional, conservative appearance. The colour blue encourages feelings of calm and relaxation. When a student is performing study tasks, they may feel anxious or afraid. In these situations, the hormone adrenaline is released. At the same time, the hormone adrenaline is also released when a person feels enthusiastic or excited while performing a task. As a result, the blue colour works to comfort and relax the student, which enables him to complete his tasks in complete comfort. Additionally, the colour blue encourages students to engage in thinking and creative tasks and increases their desire to search for knowledge. As a result, it can be utilised in game design to either tell the player about pertinent facts or challenge him to come up with a clever solution.

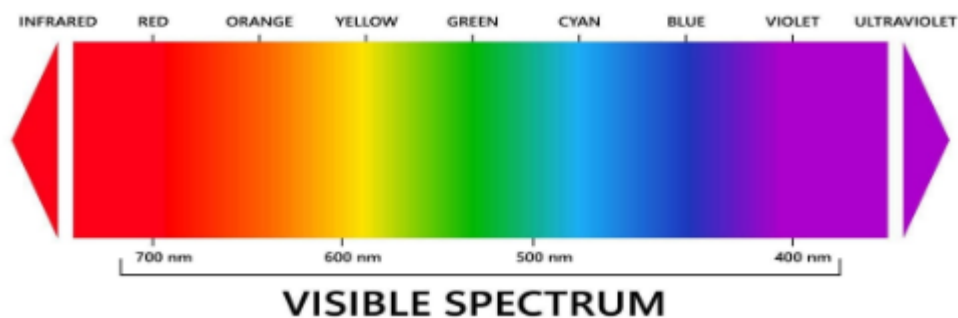


Figure 4. 5: Visible Light Spectrum [\[11\]](#)

As shown in the illustration above, the color that is the most soothing is blue. Because it is a very short wavelength color, it has no stimulating effect.

For this reason, blue is good for making the player feel relaxed.

### 4.7.2 Page Design

We created a basic interface for the game using Figma based on the functional requirements that we determined:

#### 1. Start Page

When the player enters the game, the start page appears, and when he presses Enter, he moves to Sign up/ Sign in Page.



Figure 4. 6: Start Page

#### 2. Sign up/ Sign in Page

Through this page, the player can sign in to the game, and to log in to the game, he needs to enter his username and password in the field designated for that, and when he finishes signing in, he goes to the car showroom page.



Figure 4. 7: Sign-up/ Sign-in Page

### 3. Car Showroom Page

On the car showroom page, the player can buy a new car or upgrade his car, and by pressing the Race Tracks button, he can go to the racing tracks page, and he can go to the tasks page, by pressing the Add Task button.

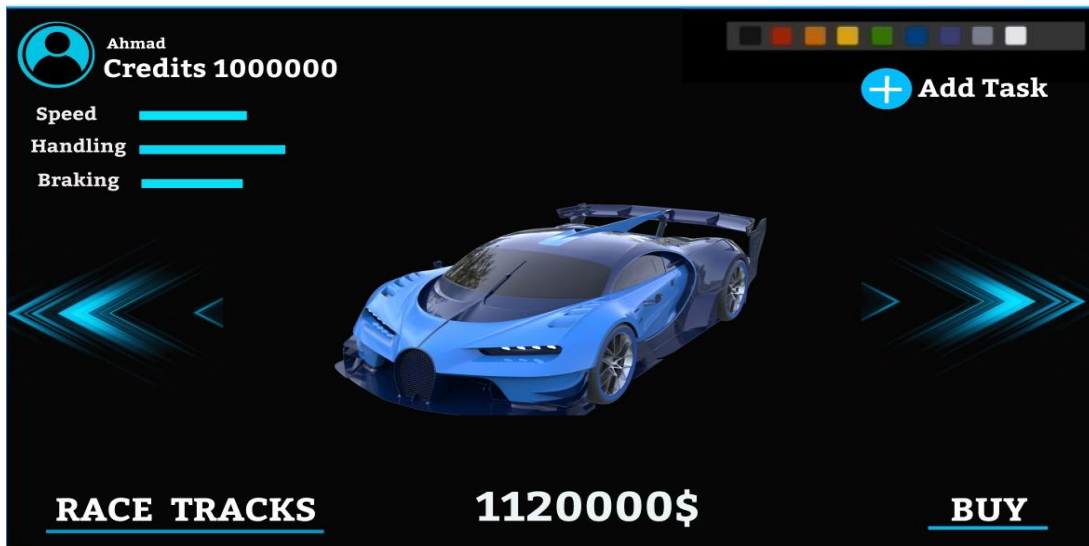


Figure 4. 8: Car Showroom Page

### 4. Tracks Page

On the Race Tracks page, the player can choose a track for the race, and he can return to the Car Showroom page by pressing a Car Showroom button, or start the race by pressing the Start button.



Figure 4. 9: Tracks Page

## 5. Race Track Type Page

After choosing the race track, the race type page appears, in which the player can choose the type of race, the number of race laps, and the number of opponents.

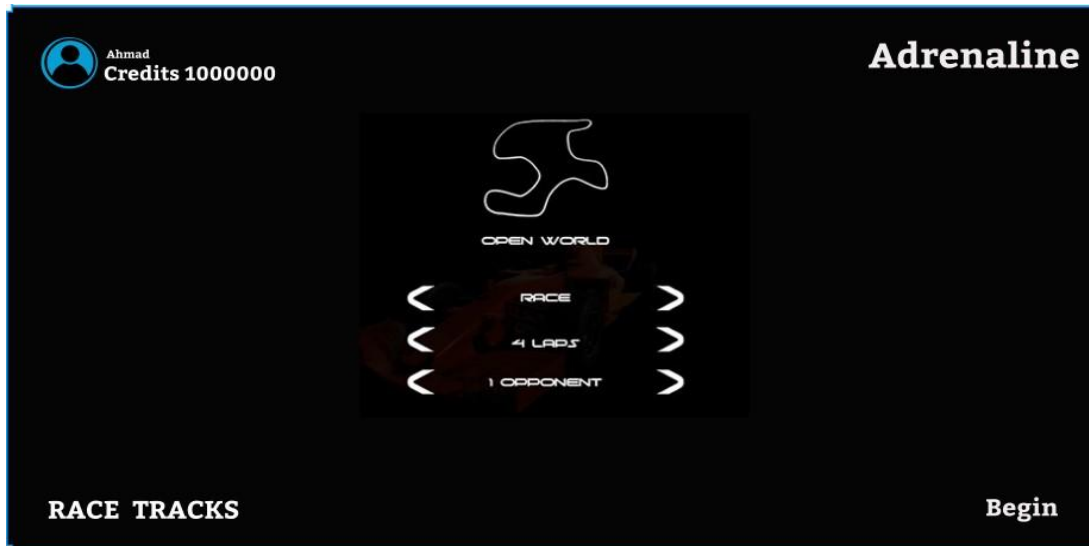


Figure 4. 10: Race Track Type Page

## 6. Tasks Page

On the tasks page the player can check tasks, add new tasks, edit tasks, or delete tasks, and he can go to the car showroom page by pressing a Car Showroom button, or go to the profile page by pressing the Profile button.

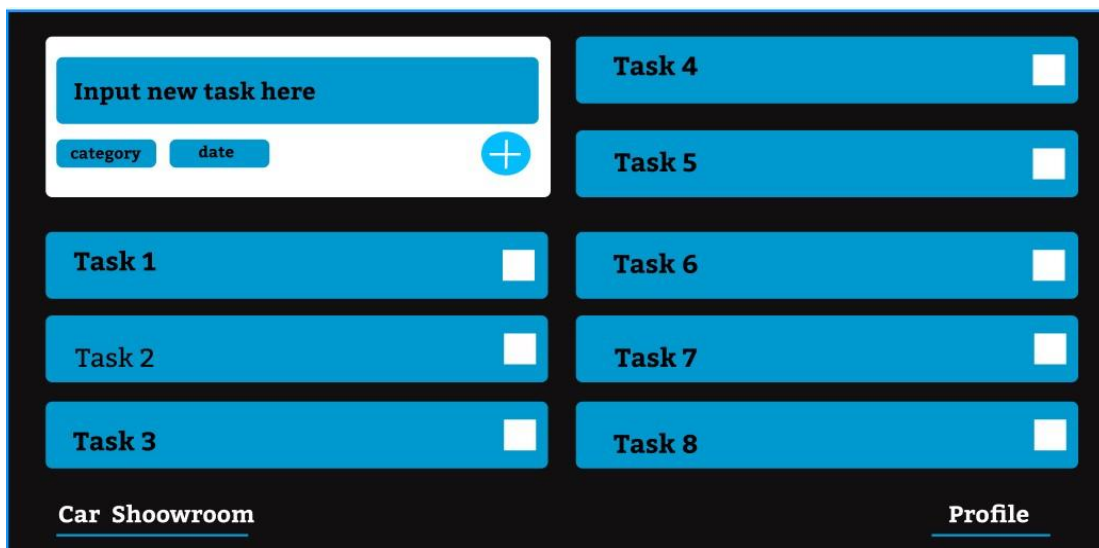


Figure 4. 11: Tasks Page

## 7. Profile Page

On the profile page, the player can see his name, Bio, Score, number of pending tasks, number of completed tasks, number of cars, number of races won, number of lost races, and his badge.

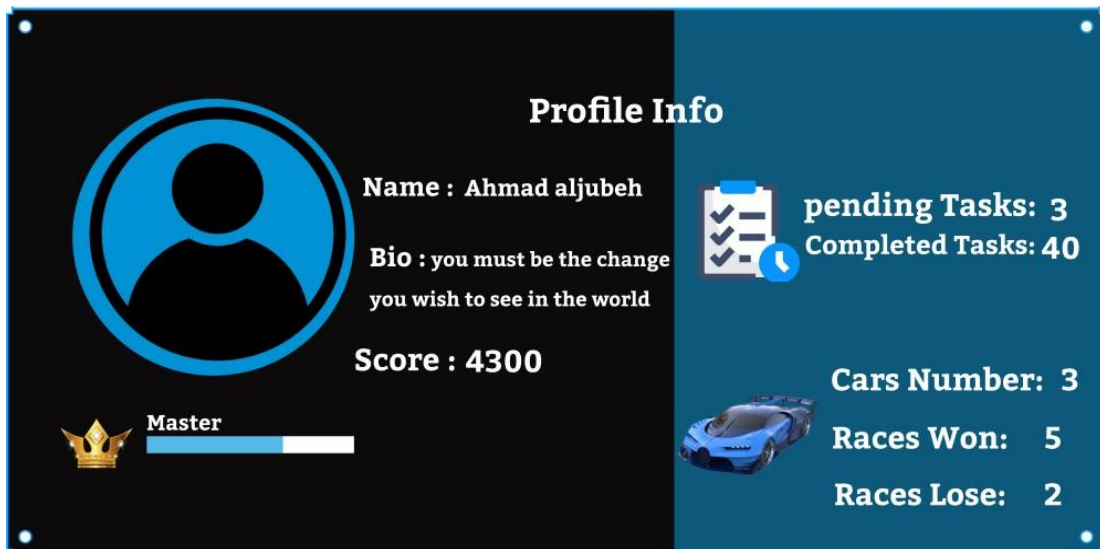


Figure 4. 12: Profile Page

## 8. Race Page

When pressing the Begin button on the race type page, the player will be taken to the race page, where he can start the race and enjoy the game.

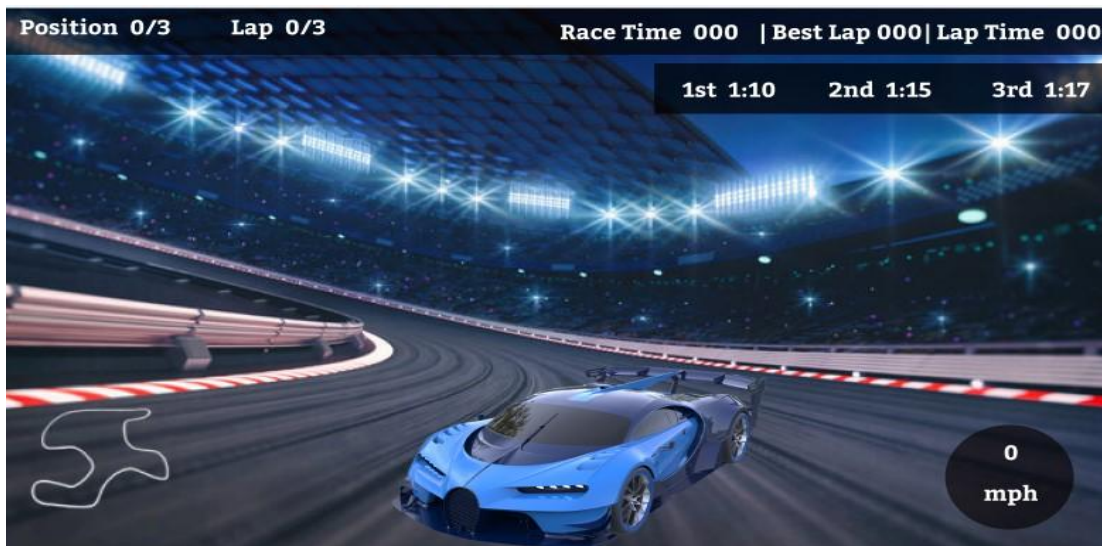


Figure 4. 13: Race Page

## Chapter 5: Software Demonstration

|                               |    |
|-------------------------------|----|
| 5.1 Introduction              | 51 |
| 5.2 Technology And Tools Used | 52 |
| 1. Unity Engine               | 52 |
| 2. Unity Assets Store         | 52 |
| 3. Blender                    | 52 |
| 4. C# Programming language    | 53 |
| 5. Visual Studio              | 53 |
| 6. SQLite                     | 53 |
| 7. GitHub                     | 53 |
| 5.3 Implementation Details    | 53 |
| 5.4 Implementation Issue      | 58 |
| 5.5 Game View                 | 59 |
| 5.5.1 Game Screen             | 59 |

## 5.1 Introduction

This chapter will discuss the technologies that were used in building the game. Also some of the implementation details and issues that faced the project during the implementation.

## 5.2 Technology And Tools Used

### 1. Unity Engine

Unity is a 3D/2D game engine and powerful cross-platform IDE for developers.

As a game engine, Unity is able to provide many of the most important built-in features that make a game work. That means things like physics, 3D rendering, and collision detection. From a developer's perspective, this means that there is no need to reinvent the wheel. Rather than starting a new project by creating a new physics engine from scratch—calculating every last movement of each material, or the way light should bounce off of different surfaces.[12]

The decision to choose Unity as the game development platform was driven by its versatility and support for various database representation methods. Unity's flexibility allows for seamless integration of the necessary database functionality into the game, particularly for the tasks page designed for students.

To enhance the user experience and facilitate task management, we utilised Unity assets for a date picker. These assets, available on the Unity Assets Store website, provide ready-to-use components that enable users to select and set task deadlines conveniently within the game interface.

By leveraging Unity's asset ecosystem, we were able to save development time and effort by incorporating pre-built components rather than developing the date picker functionality from scratch. This decision not only streamlined the implementation process but also ensured a reliable and user-friendly task management system for students.

### 2. Unity Assets Store

**Asset Store:** A Unity website that lets you download premade assets for use in your game development projects. Unity developers can also publish their own premade assets to the Asset Store for other developers to download and use.

Unity Asset Store was used in the project to download the formula one car, sport cars, the trees, the date picker and sci-fi buttons for designing the UI for the game.

### **3. Blender**

Blender is a powerful and versatile open-source 3D creation suite that offers a comprehensive set of tools for modelling, animation, rendering, and more. With its intuitive interface and extensive feature set, Blender has become a go-to software for artists, designers, and animators. It allows users to create stunning visual effects, realistic animations, and detailed 3D models. Its capabilities range from complex character animation to architectural visualisation and even game development. Blender's active and supportive community, coupled with its constant updates and improvements, ensures that users have access to a wealth of tutorials, resources, and plugins to enhance their creative workflow. With its free and open-source nature, Blender has gained popularity in various industries and stands as a testament to the collaborative power of the open-source community.[13]

We use blender to design the race tracks for the game .

### **4. C# Programming language**

C# is a highly expressive, object-oriented programming language used for a broad range of multi-platform software. Developers often write in C# to build projects intended for long-term use, simply because it's easy to maintain its code. Apart from its universality, the strength of C# lies in the fact that it is a modern C language, making it easy to update, upgrade, and rebuild your software's backend. Also C# is the language used in Unity.

### **5. Visual Studio**

It is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring.

### **6. SQLite**

SQLite is a popular, lightweight, and embedded relational database management system (RDBMS). It is not a standalone server-based database like MySQL or PostgreSQL but is designed to be embedded within applications. SQLite is written in the C programming language and is known for its simplicity, efficiency, and ease of use.

### **7. GitHub**

GitHub is a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.[18] Also It

offers the distributed version control and source code management functionality of Git.

GitHub and Git were as important as other technologies that we used. Each update in the project was pushed on github with a message that represents the changes made. So we can go back for any version of the project we want.

## 5.3 Implementation Details

During implementation we always tried to follow the conventions used in Unity engine and C# language. In the Unity engine we made a folder structure and divided files into categories. And for the code we tried to write clean and optimised code.

### 5.3.1 User Authentication:

- Develop a user authentication system to allow students to create accounts and securely log in.
- Implement appropriate security measures to protect user information and ensure secure access to the game.

### 5.3.2 Task Management:

- Develop a user-friendly interface that allows students to add, edit, and delete tasks related to their academic responsibilities.
- Implement features to set due dates and times for tasks, enabling students to organise their workload effectively.
- Track task completion status and provide visual feedback to help students monitor their progress and stay motivated to complete their tasks.

### 5.3.3 Car Customization and Progression:

- Create a system where students can use their earned rewards to improve their racing cars.
- Define car attributes such as speed, acceleration, handling, and appearance customization options.
- Allow students to purchase upgrades or unlock new cars/tracks using the rewards they have earned.

### 5.3.4 Race Selection and Modes:

- Develop a race selection menu where students can choose the race they want to participate in.
- Implement different race modes such as normal races or timed races to provide variety and challenge.
- Allow students to select their preferred tracks and customise race parameters like opponents and laps.

### 5.3.5 Rewards System:

- Design a rewards system where students earn points or currency for completing tasks.
- Define various rewards such as cash, car upgrades, or unlocked tracks based on the accumulated points.

### 5.3.6 Game Mechanics:

- Design the core game mechanics, including player controls, AI opponents, racing physics, and collision detection.
- Implement controls for accelerating, braking, and steering the car.
- Develop AI opponents with varying levels of difficulty, implementing their behaviour and racing strategies.

### 5.3.7 Track Design and Environment:

- Create visually appealing and challenging racing tracks using 3D modelling tools like Blender.
- Design tracks with varying layouts, obstacles, and terrain to provide a diverse racing experience.
- Implement environmental elements such as weather effects, day-night cycles, and dynamic lighting.

### 5.3.8 Race Selection and Modes:

- Develop a race selection menu where students can choose the race they want to participate in.
- Implement different race modes such as normal races or timed races to provide variety and challenge.
- Allow students to select their preferred tracks and customise race parameters like opponents and laps.

### 5.3.9 Database

In our project, we utilized PlayerPrefs to store important player data such as CreditAmount, Bio, RacesWon, RacesLost, CompletedTasks, PendingTasks, and CarsNumber. PlayerPrefs is a convenient class in Unity that allows us to store player preferences between game sessions. It supports storing various data types such as strings, floats, and integers into the user's platform registry. This choice was made due to its simplicity and compatibility with our specific needs.

Additionally, for more complex data storage and management, we implemented SQLite, a lightweight and versatile database engine. SQLite enables us to create and manage tables to store

information related to player registration and tasks. This approach provides a more robust and organized solution for handling extensive player data, ensuring efficient retrieval and manipulation as needed.

In our project, we have utilised Unity's build settings to optimise performance and ensure compatibility across platforms. We have also leveraged terrain customization to create visually immersive racing environments. Additionally, we implemented a waypoint circuit system for accurate navigation and strategic AI opponents.

### 5.3.9 Build Settings:

Build settings in Unity play a crucial role in our project's documentation as they determine the arrangement and configuration of scenes within our car racing game. By accessing the Build Settings window, we can add the necessary scenes to the build index and designate the starting scene. This allows us to control the progression and flow of the game, ensuring smooth transitions between different scenes. Properly configuring the build settings is essential for creating a cohesive and enjoyable gameplay experience, where players can seamlessly move between tasks, races, and other game components.

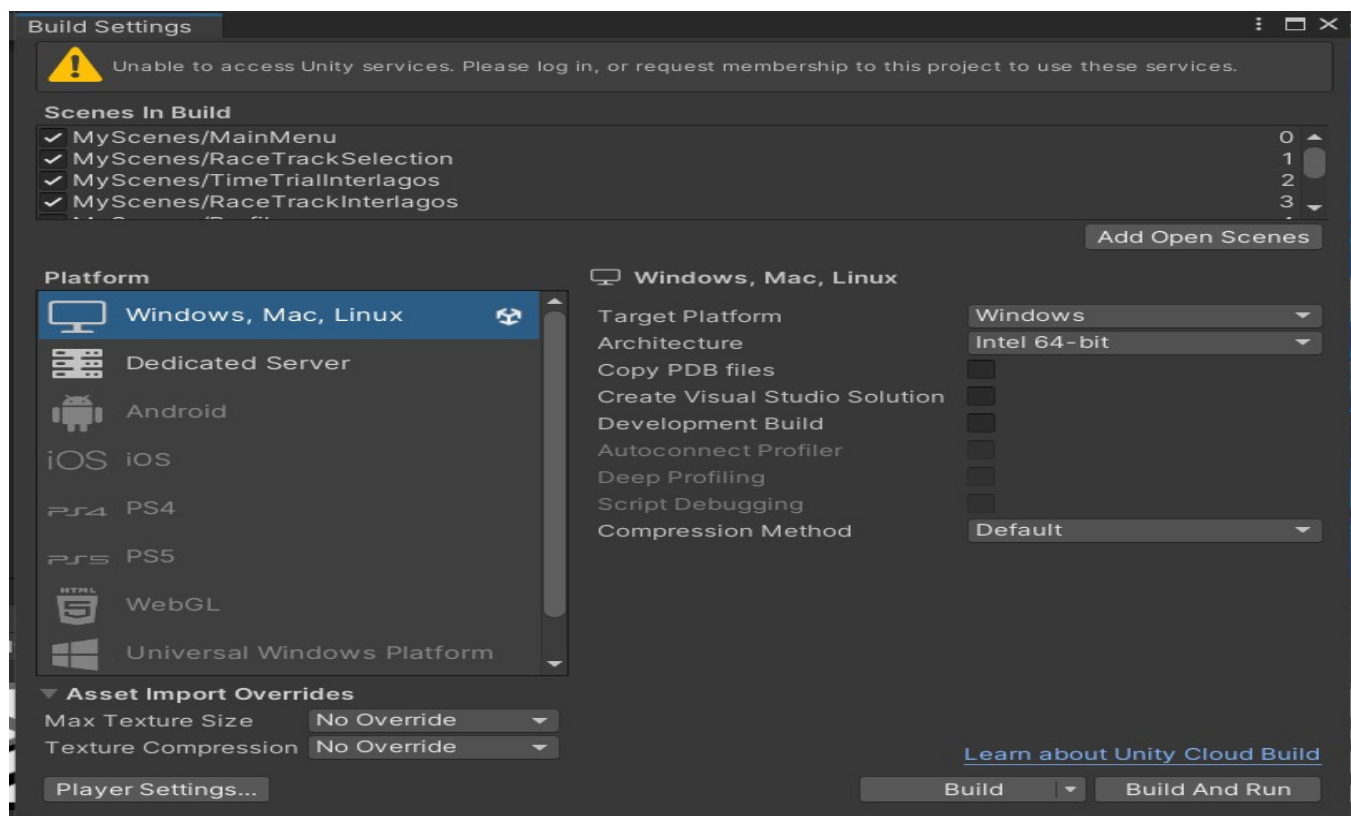


Figure 5.1: Build Settings

### 5.3.10 Finish Line:

Figure 5.2 shows that the properties for finish line used the most important property is 'Is Trigger', This feature allows us to determine whether the player has finished the race or not so that we can perform the appropriate special operations, as they are linked to the OnTiggerEnter function, this function in which we carry out the operations required for each case.

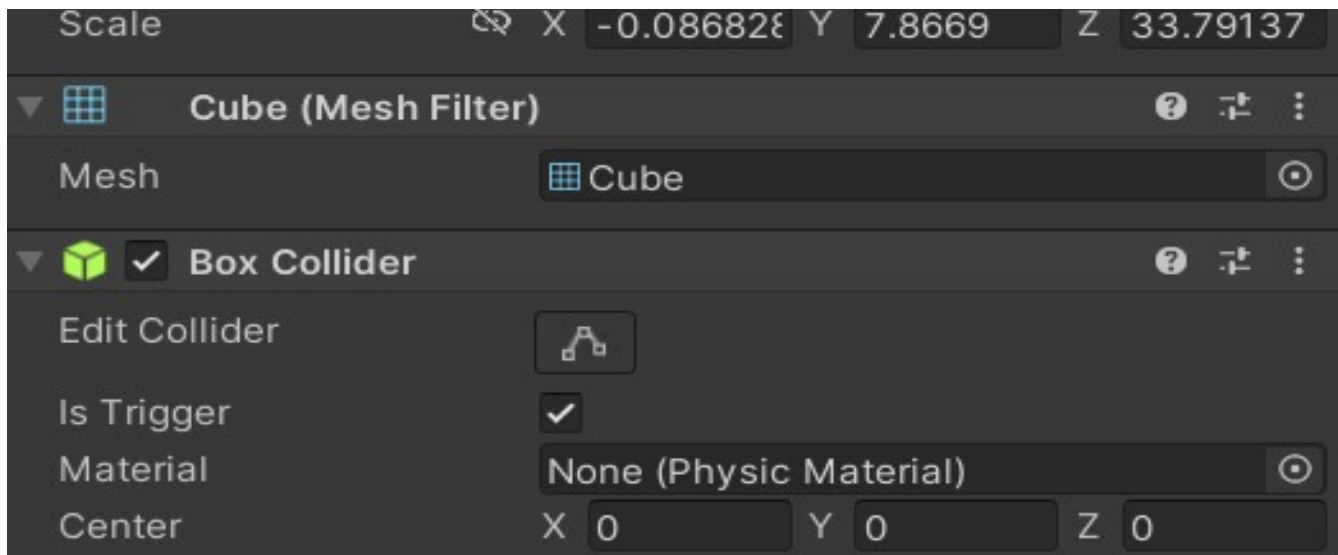


Figure 5.2: Finish Line

### 5.3.11 Terrain Trees:

In our project, we have utilised Unity's tree customization features to populate the racing environments with realistic and varied vegetation, enhancing the visual richness and immersion of the game.

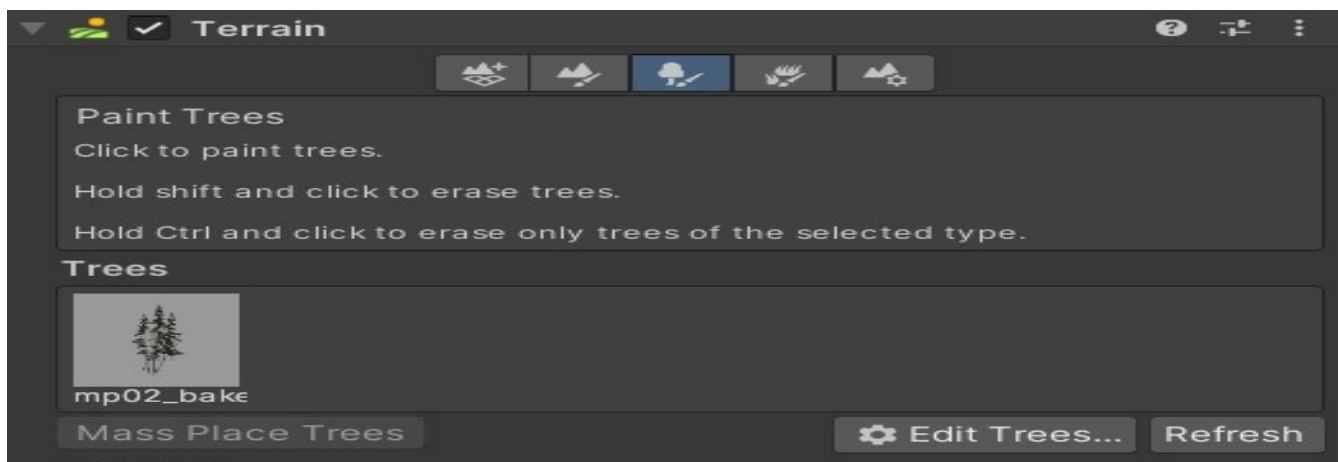


Figure 5.3: Terrain Trees

### 5.3.12 Way Point Circuit:

In our car racing game project, we have implemented a waypoint circuit system to guide the AI-controlled cars along a predefined path. We have defined waypoints as empty game objects strategically placed in the scene, and each waypoint is associated with a script that handles logic and connections specific to the AI cars. By utilising a car control script tailored for AI behaviour, we enable the AI cars to autonomously move towards the current waypoint, detect proximity to waypoints, and progress to the next one accordingly. This waypoint circuit system ensures that the AI cars navigate the track accurately, follow the intended path, and provide challenging opponents for players in our racing game.

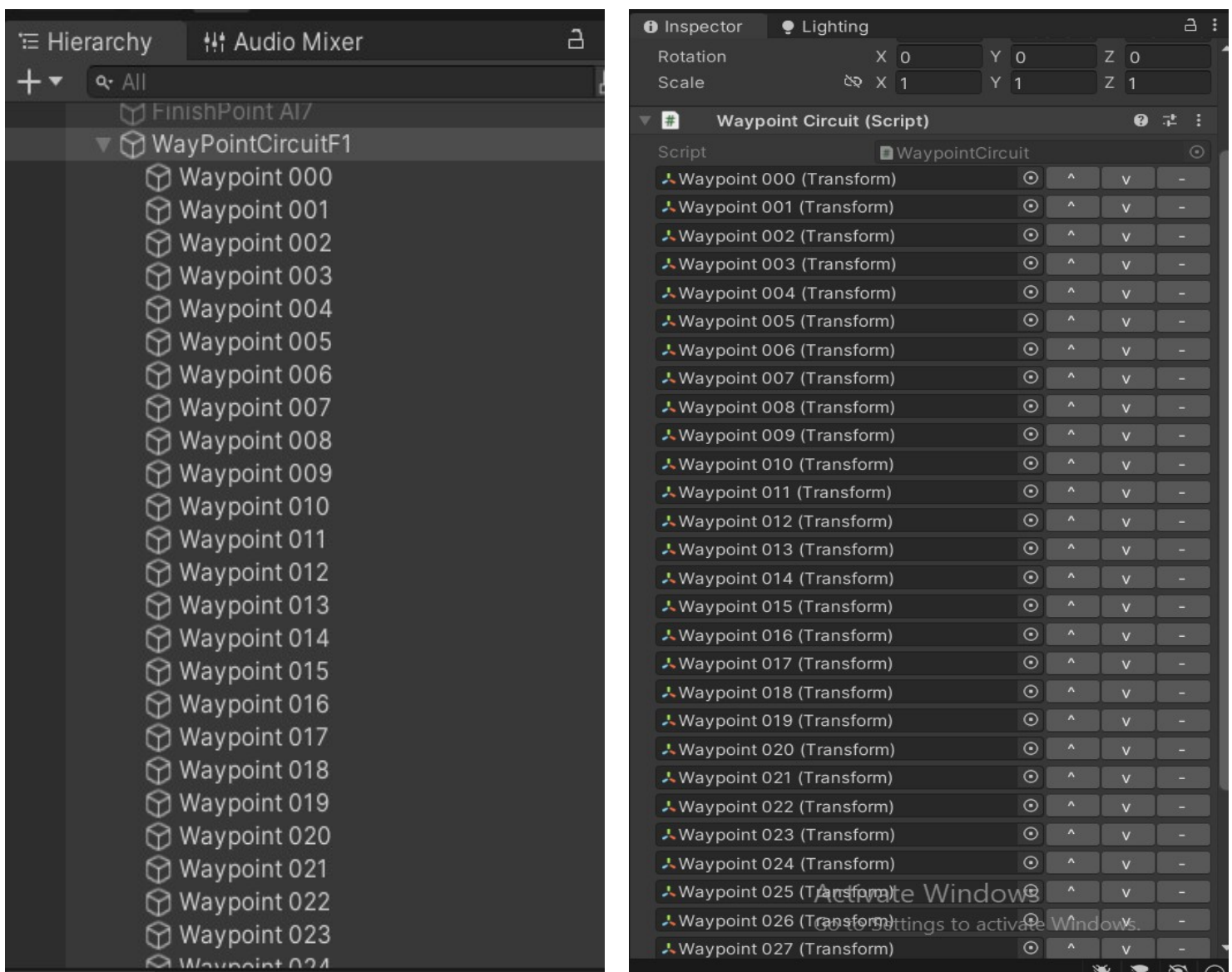


Figure 5.4: Way Point Circuit

## 5.4 Implementation Issue

The challenge we faced while working on the project can be restricted to the lack of knowledge in the Unity engine. Because it was our first time working with it and it was a little different than web or mobile development. As a result, we used the learning by doing method.

## 5.5 Game View

We have 15 basic pages and here are pictures from each of them:

### 5.5.1 Game Screen

Upon pressing the "Enter" key to start, the student will be directed to either sign up or sign in to access the game. Once authenticated, they will have the freedom to choose between two main options: the tasks page or the game page. The tasks page provides a platform where students can add and manage their academic tasks, ensuring they stay organised and on track with their coursework. By diligently completing these tasks, students are rewarded, enabling them to develop their racing cars and increase their chances of winning in the game. On the other hand, if the student opts for the game page, they can customise their experience by selecting the desired racing track, determining the number of opponents they wish to compete against, and setting the number of laps for the race. This flexibility empowers students to tailor their gameplay to their preferences while maintaining a balance between their academic responsibilities and the exhilaration of competitive racing.

#### 1. Start Page



Figure 5.5: Start Page

## 2. Sign Up/ Sign In Page



Figure 5.6: Sign Up / Sign In Page

## 3. Help Page

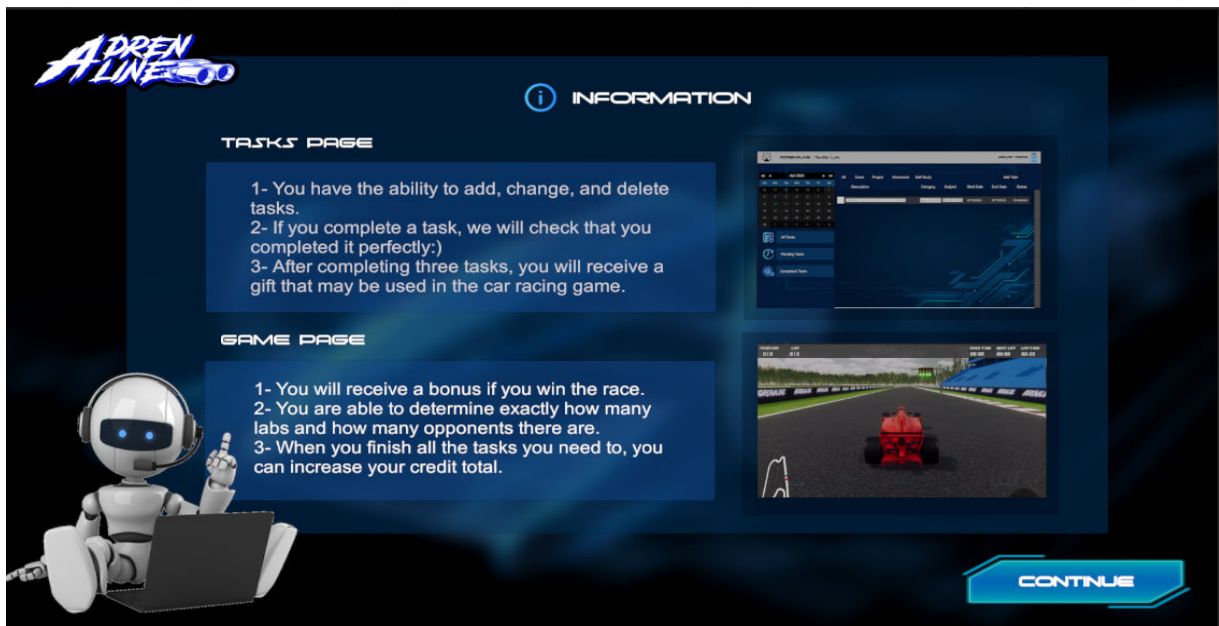


Figure 5.7: Help Page

#### 4. Tasks OR Game Page



Figure 5.8: Tasks OR Game Page

#### 5. Select Track Page

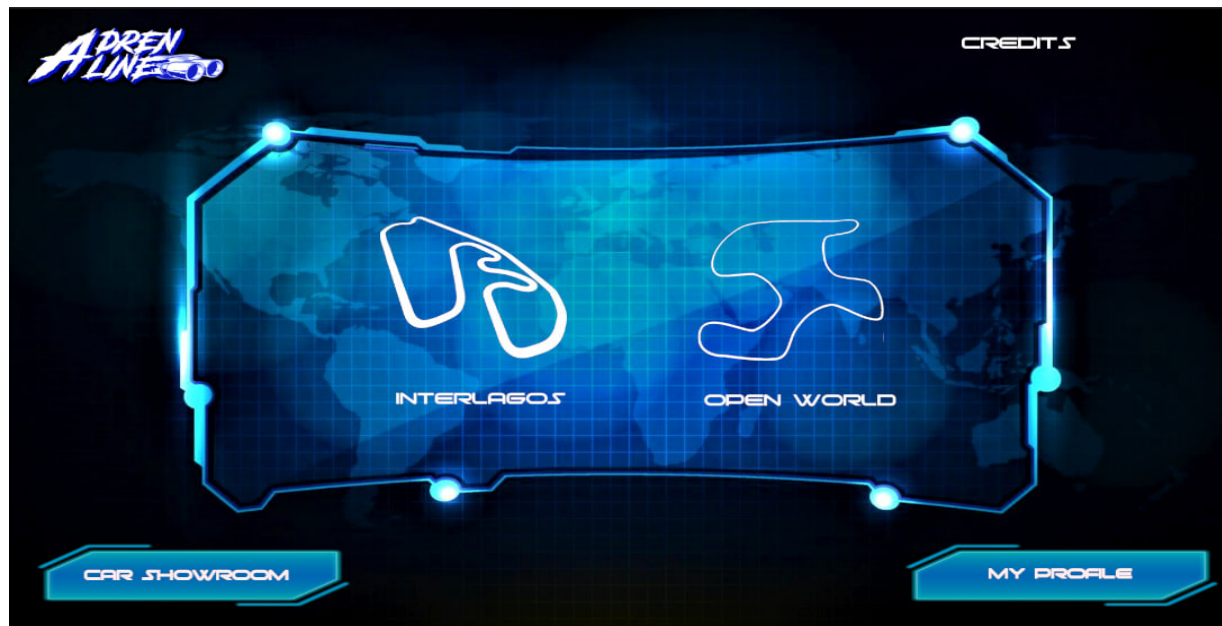


Figure 5.9: Select Track Page

6. Race Track Type Page

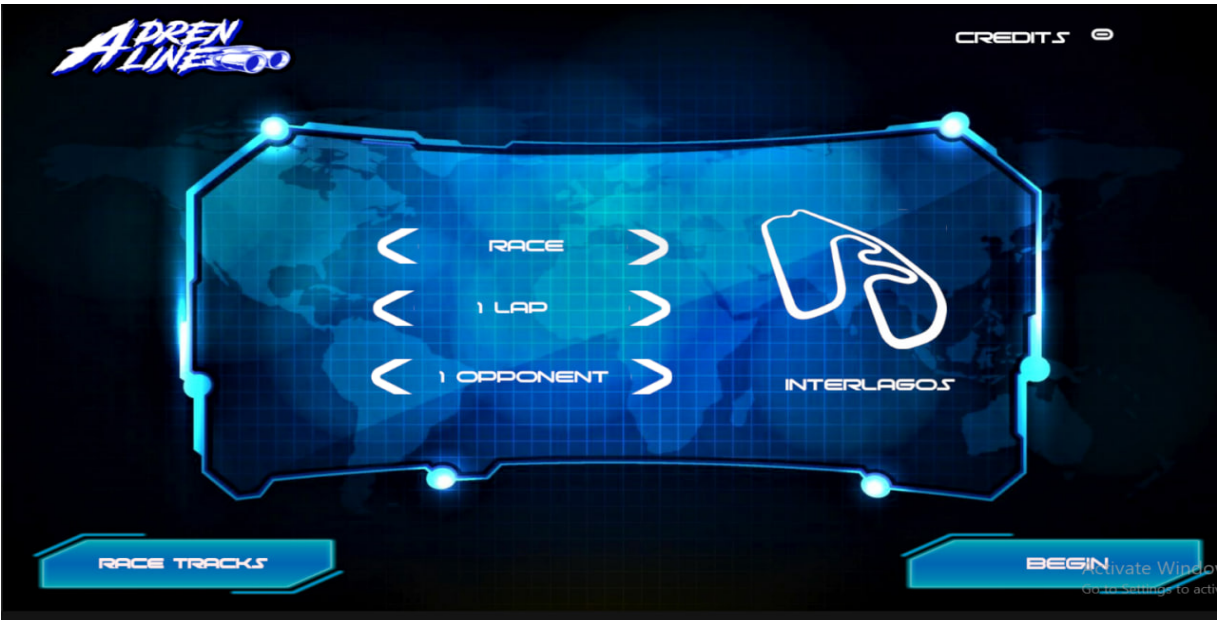


Figure 5.10: Race Track Type Page

7. During Racing



Figure 5.11: During Racing Screen

## 8. Quit Menu Page

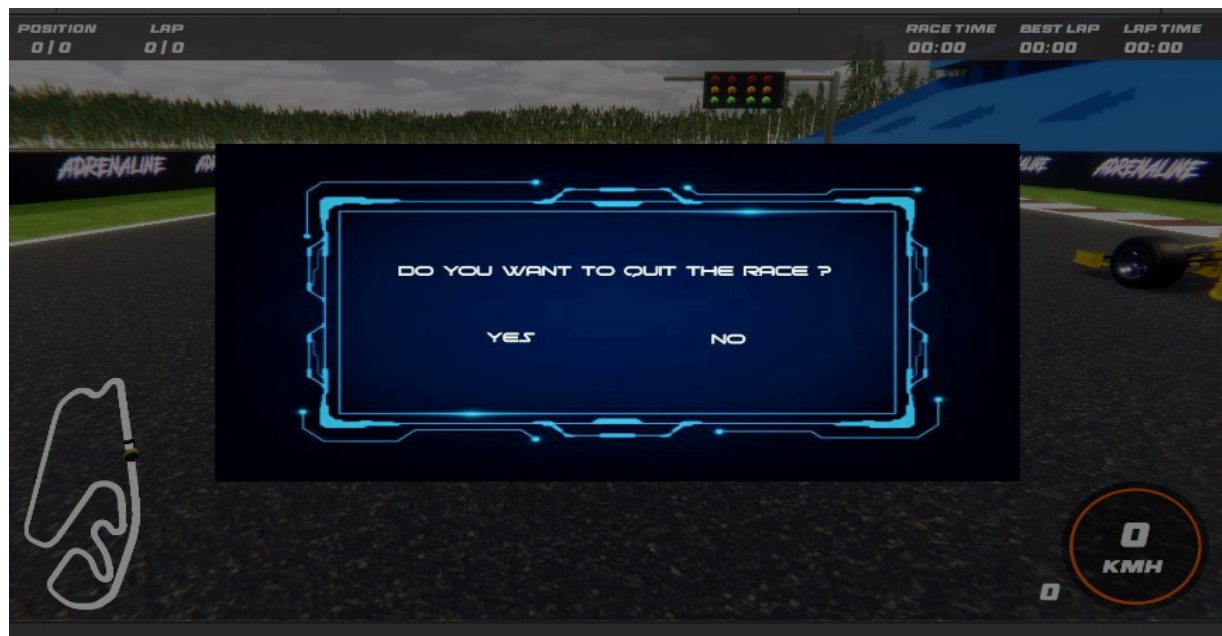


Figure 5.12: Quit Menu Page

## 9. Race Results Screen

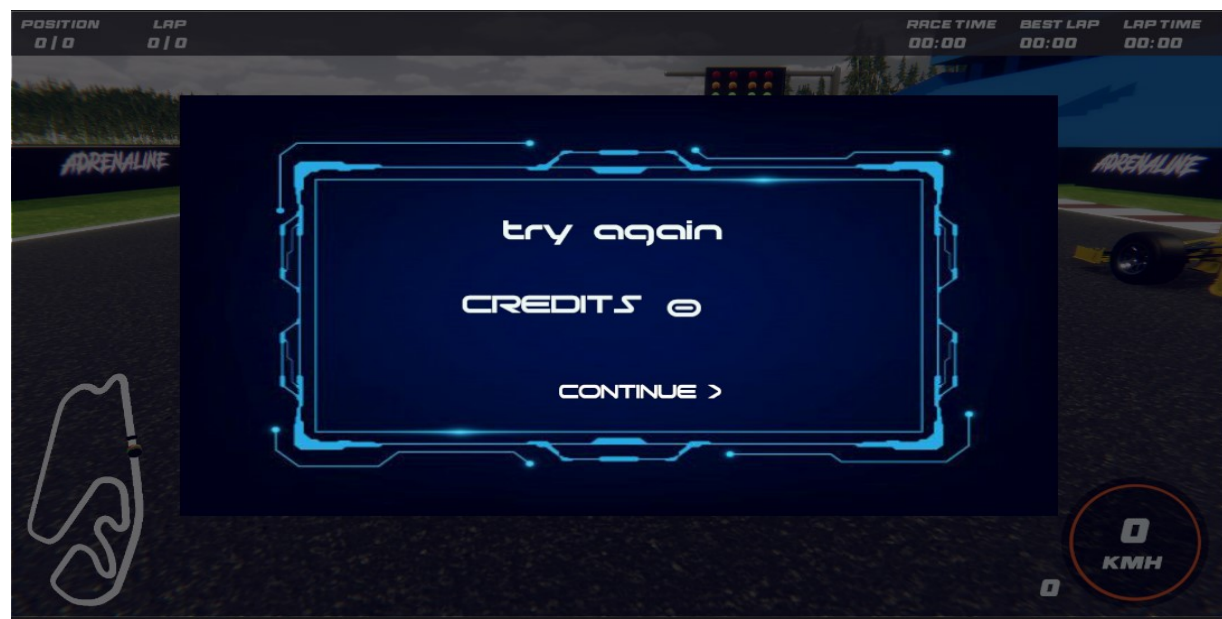


Figure 5.13: Race Results Screen

## 10. Car Showroom Page



Figure 5.14: Car Showroom Page

## 11. Interlagos Track Screen

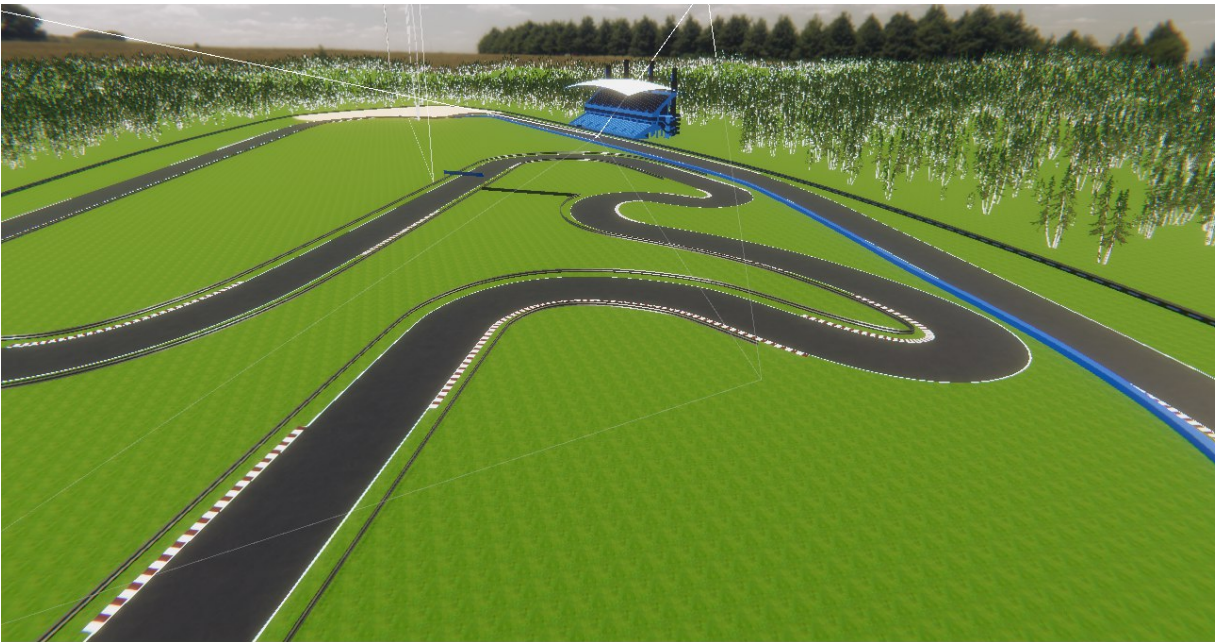


Figure 5.15: Interlagos Track Screen

12. Open World Track Screen

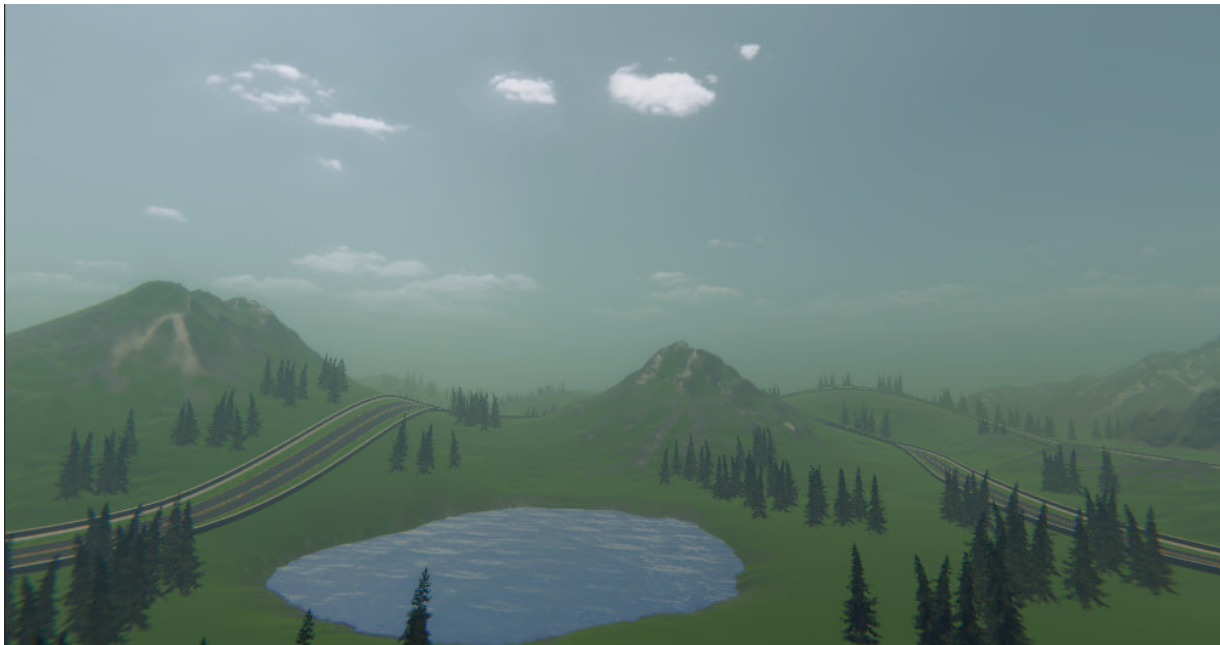


Figure 5.16: Open World Track Screen

13. Tasks Page

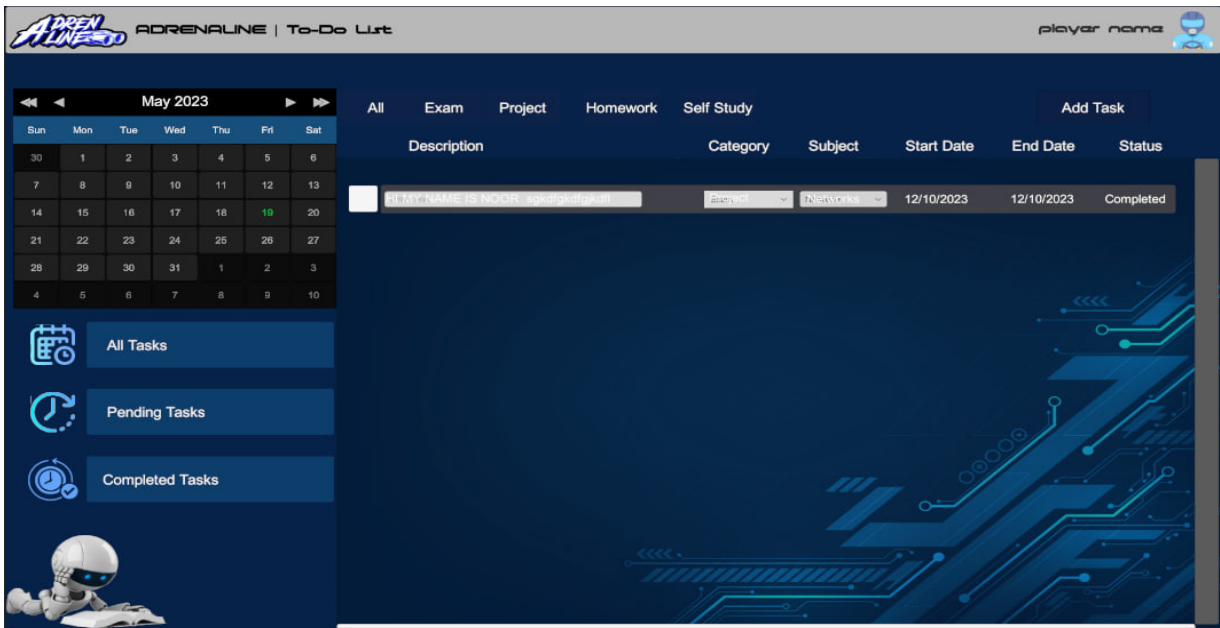


Figure 5.17: Tasks Page

## 14. Check The Task Page

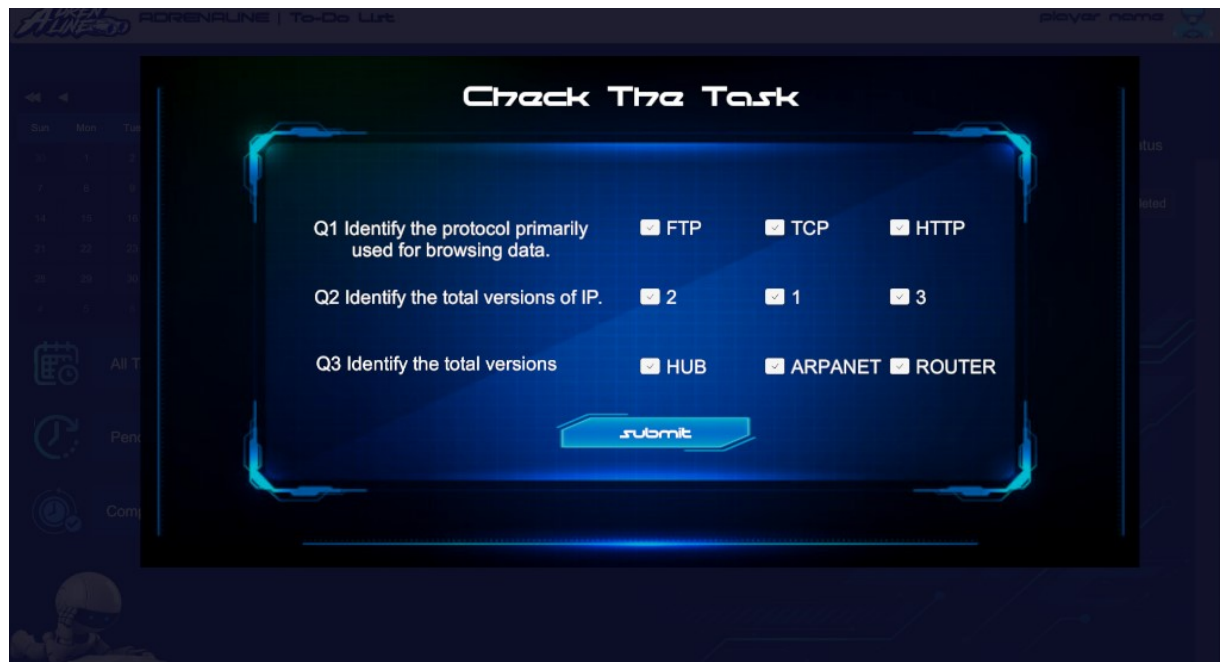


Figure 5.18: Check The Task Page

## 15. Claim Reward Page

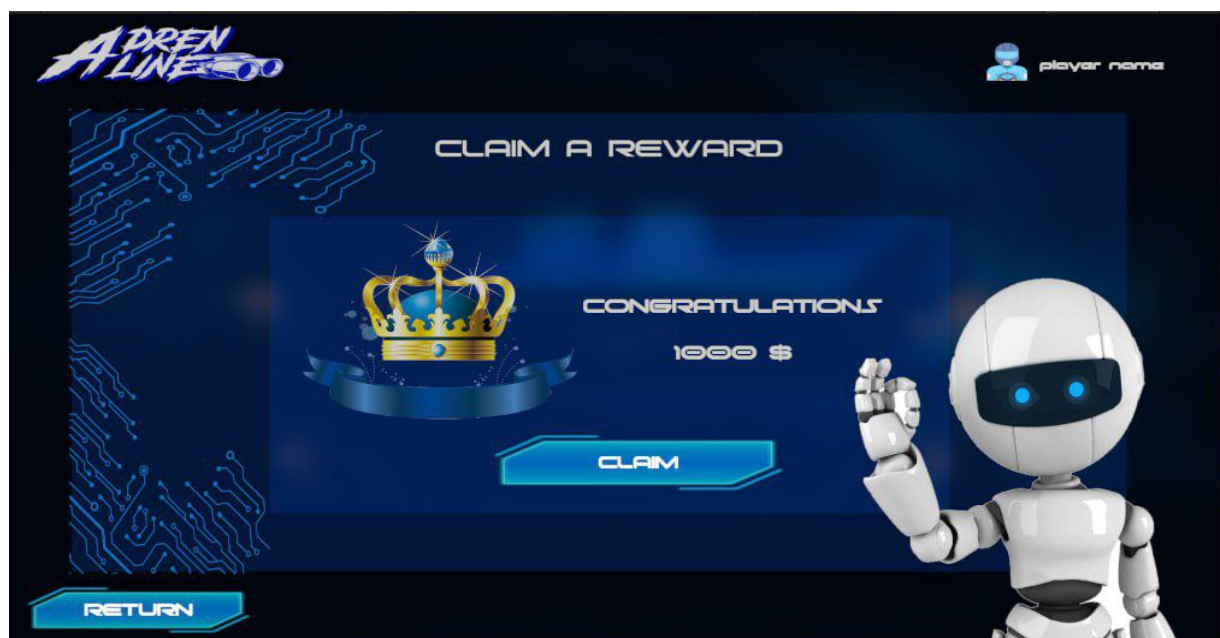


Figure 5.19: Claim Reward Page

## 16. Profile Page

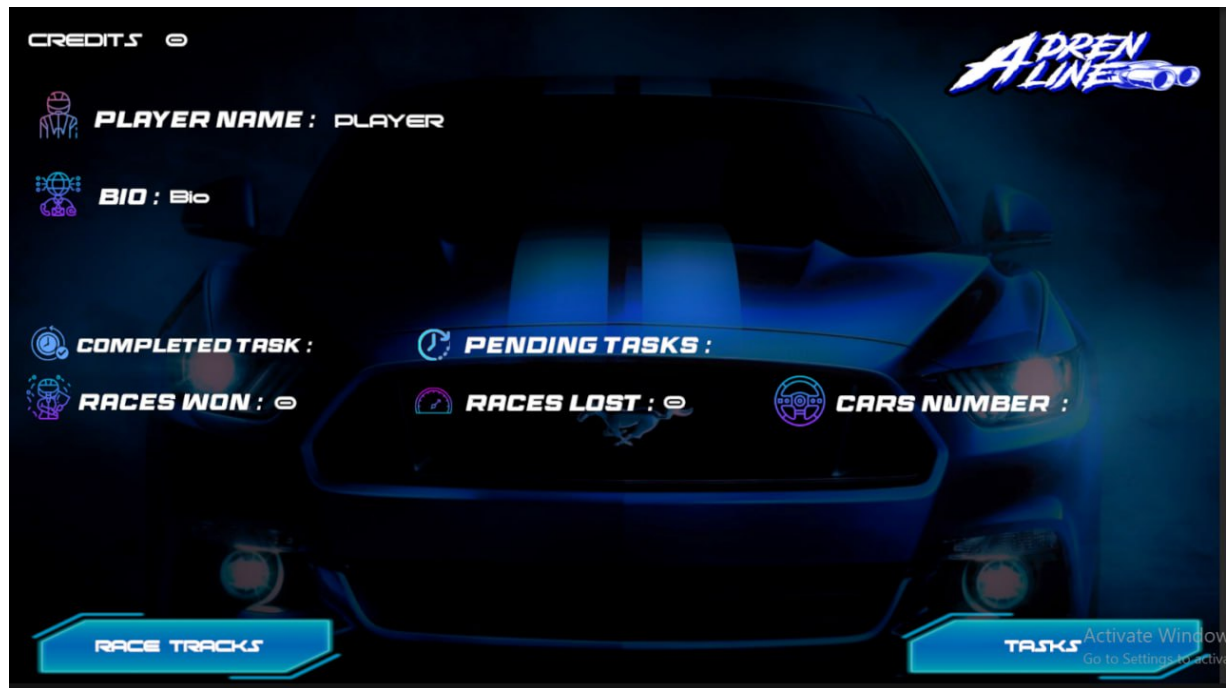


Figure 5.20: Profile Page

## Chapter 6: Testing

|                                     |    |
|-------------------------------------|----|
| 6.1 Introduction                    | 69 |
| 6.2 Functional Requirements Testing | 69 |
| 6.2.1 Scenes Transformations        | 69 |
| 6.2.2 Game Screen                   | 75 |
| 6.2.3 Result Screen                 | 76 |
| 6.2.4 Tasks Screen                  | 77 |
| 6.4 Profile Page Testing            | 79 |
| 6.5 Sounds Testing                  | 80 |

## 6.1 Introduction

This chapter will go over testing for functional requirements.

To ensure that our game works as expected and to assist in managing new changes in specification or implementation.

## 6.2 Functional Requirements Testing

### 6.2.1 Scenes Transformations

Table 6.1: Scenes Transformations Test Cases.

| Test Case                 | Scenario                                                                                                              | Input data                            | Expected Output                             | Actual Output               | Test result |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------|---------------------------------------------|-----------------------------|-------------|
| <b>Start game button.</b> | The Student press Enter Key                                                                                           | Pressing the "Enter" Key              | Go to the Sign Up Page                      | SignUp screen was displayed | Pass        |
| <b>Sign Up Button</b>     | 1- Fill in the required fields, such as username, email address, and password.<br><br>2- Click on the sign-up button. | Username<br>Email Address<br>Password | The user is redirected to the Sign In page. | Sign In page was displayed  | Pass        |
| <b>Sign In Button</b>     | 1- Fill in the required fields, such as email address, and password.<br><br>2- Click on the sign-in button.           | Email Address<br>Password             | Go to the help page                         | Help page was displayed     | Pass        |

Table 6.2: Tasks Or Game Page Test Cases.

| Test Case                   | Scenario                                                                    | Input data | Expected Output               | Actual Output                     | Test result |
|-----------------------------|-----------------------------------------------------------------------------|------------|-------------------------------|-----------------------------------|-------------|
| <b>Play The Game Button</b> | On the “Tasks OR Game Page” Locate and click on the “Play The Game” button. | -          | Go to the “Select Track” page | “Select Track” page was displayed | Pass        |
| <b>Add Task Button</b>      | On the “Tasks OR Game” page locate and click on the “Add Task” button.      | -          | Go to the “Tasks” page        | “Tasks” page was displayed        | Pass        |

Table 6.3: Select Race Track Page Test Cases.

| Test Case                      | Scenario                                                                      | Input data | Expected Output                                                                                                                            | Actual Output                                                                                        | Test result |
|--------------------------------|-------------------------------------------------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|-------------|
| <b>Open World Track Button</b> | On the “Select Track” page locate and click on the "Open World Track" button. | -          | When the player selects the "Open World Track" option, he will be redirected to the Open World Track properties page.                      | The "Open World Track" is selected, and the player is redirected to the open world track properties. | Pass        |
| <b>Car Showroom Button</b>     | On the “Select Track” page locate and click on the "Car Showroom" button.     | -          | When the player clicks the "Car Showroom" button, he will be redirected to the Car Showroom page.                                          | The "Car Showroom" button was clicked, and the player was redirected to the Car Showroom page.       | Pass        |
| <b>Begin</b>                   | On the “Race Track Type” page locate and click on the "Begin" button.         | -          | The game starts with the chosen track and its corresponding characteristics, such as racing type, number of opponents, and number of laps. | Game screen was displayed with the chosen track and its corresponding characteristics.               | Pass        |
| <b>Race Tracks Button</b>      | On the “Race Track Type” page locate and click on the "Race Tracks" button.   | -          | When the player clicks the "Race Tracks" button, he will be redirected to the Select Race Track page.                                      | The "Race Tracks" button was clicked, and the player was redirected to the Race Track page.          | Pass        |

Table 6.4: Car Showroom Page Test Cases.

| Test Case                      | Scenario                                                                | Input data | Expected Output                                                                                                                                                                                                                         | Actual Output                                                                                                                                                                                                                                               | Test result |
|--------------------------------|-------------------------------------------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <b>Buy Button</b>              | On the “Car Showroom” page locate and click on the “Buy” button.        | -          | When the player clicks the “Buy” button<br>If the player owns its price, then its price is deducted from the player, and after that the car becomes the player’s property, but if the player does not have its price, he cannot buy it. | When the player clicks the buy button<br>And the player owned its price, so its price was deducted from the player, and after that the car became the property of the player, but in the event that the player did not have its price, he could not buy it. | Pass        |
| <b>Select Car Color Button</b> | On the “Car Showroom” page Select the needed colour option for the car. | -          | When the player selects a colour then the colour of the selected car will change to the selected colour.                                                                                                                                | When the player selected a colour then the colour of the selected car changed to the selected colour.                                                                                                                                                       | Pass        |

## 6.2.2 Game Screen

Table 6.5: Game Screen Test Cases.

| Test Case                                         | Scenario                                                                     | Input data | Expected Output                                                                                                                                             | Actual Output                                                                                          | Test result |
|---------------------------------------------------|------------------------------------------------------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|-------------|
| <b>Go Message</b>                                 | Wait for the race to begin                                                   | -          | The “Go” message will display in the middle of the screen.                                                                                                  | The “Go” message was displayed in the middle of the screen.                                            | Pass        |
| <b>Hit Barrier</b>                                | The racing car hits a Barrier                                                | -          | A crash will sound and the car will stop.                                                                                                                   | There was a sound of collision and the car stopped                                                     | Pass        |
| <b>Player's Car Movement on Grass</b>             | Control the player's car to move on the grass.                               | -          | The player's car moves with increased difficulty on the grassy terrain                                                                                      | The player's car moved with increased difficulty on the grassy terrain.                                | Pass        |
| <b>Player's Car Moving in the Wrong Direction</b> | Change the direction of movement of the racing car to the opposite direction | -          | If the player's car moves in the wrong direction or deviates from the intended track, a "Wrong Way" message is displayed on the screen to alert the player. | When the player's car moved in the wrong direction, a "Wrong Way" message was displayed on the screen. | Pass        |

## 6.2.3 Result Screen

Table 6.7: Result Screen Test Cases.

| Test Case              | Scenario                                                                                         | Input data | Expected Output                                                                                                                          | Actual Output                                                                                                                          | Test result |
|------------------------|--------------------------------------------------------------------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <b>Game Over</b>       | Player fails to complete the race within time or other players beat him.                         | -          | Go to the game “Result” screen and display the right score                                                                               | “Result” screen was displayed and the right score was displayed too                                                                    | Pass        |
| <b>Win</b>             | The player completes the race in time or reaches the finish line before the rest of the players. | -          | Go to the game “Result” screen which will display the correct score achieved by the player and rewards him with the appropriate credits. | “Result” screen was displayed and it displayed the correct score achieved by the player and rewarded him with the appropriate credits. | Pass        |
| <b>Continue Button</b> | The player click “Continue” button on “Result” screen                                            | -          | Go to the “Select Race Track” Page                                                                                                       | “Select Race Track” Page was displayed                                                                                                 | Pass        |

## 6.2.4 Tasks Screen

Table 6.8: Tasks Screen Test Cases.

| Test Case                     | Scenario                                                  | Input data | Expected Output                                                                | Actual Output                                                                   | Test result |
|-------------------------------|-----------------------------------------------------------|------------|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------|-------------|
| <b>Pending Tasks Button</b>   | The player click “Pending Tasks” button on “Tasks” Page   | -          | The pending tasks for the student will display on the screen.                  | The student's pending tasks were displayed on the screen.                       | Pass        |
| <b>Completed Tasks Button</b> | The player click “Completed Tasks” button on “Tasks” Page | -          | The completed tasks for the student will display on the screen.                | The student's completed tasks were displayed on the screen.                     | Pass        |
| <b>Exam Button</b>            | The player click “Exam” button on “Tasks” Page            | -          | The tasks with the exam category for the student will display on the screen.   | The student's tasks with the exam category were displayed on the screen.        | Pass        |
| <b>Add Task Button</b>        | Locate and click on the "Add Task" button.                | -          | The "Add Task" button is clicked, initiating the process to create a new task. | The "Add Task" button was clicked, initiating the process to create a new task. | Pass        |

## 6.5 Sounds Testing

Table 6.9: Sounds Test Cases.

| Test Case               | Scenario                                                           | Input data | Expected Output                                                      | Actual Output                                                       | Test result |
|-------------------------|--------------------------------------------------------------------|------------|----------------------------------------------------------------------|---------------------------------------------------------------------|-------------|
| <b>Background Music</b> | Moving between scenes and stay in the game for more than 5 minutes | -          | Background sound should play in all scenes and repeat when it finish | Background sound played in all scenes and repeated when it finished | Pass        |

## Chapter 7: Conclusion and Future Work

|                 |    |
|-----------------|----|
| 7.1 Conclusion  | 81 |
| 7.2 Future Work | 82 |

## 7.1 Conclusion

Ultimately, our efforts culminated in the creation of a motivational game designed to assist students in fulfilling their academic responsibilities throughout their university studies. By providing an intuitive interface, students can easily outline and schedule their necessary tasks, including assignments, exams, and lessons. In turn, completing these tasks grants them rewards that can be utilised to enhance their in-game performance by upgrading their racing car.

The primary objective of our game is to instil motivation and combat feelings of boredom that may arise when tackling academic obligations. Through gamification, we have transformed the process of task completion into an engaging and enjoyable experience. Students are now incentivized to complete their assignments promptly, spurred on by the prospect of earning rewards that directly impact their progress within the game.

By seamlessly integrating the academic and gaming realms, our motivational game cultivates a sense of excitement and achievement. Students no longer view their tasks as mere burdens but rather as opportunities to advance their virtual racing endeavours. This novel approach harnesses the power of motivation and enjoyment to propel students toward successfully fulfilling their academic commitments.

In summary, our motivational game serves as a catalyst for students' productivity and task fulfilment by providing them with a user-friendly platform to manage their academic responsibilities. The incorporation of rewards and game development elements incentivizes students to complete tasks in a timely manner, transforming their approach to studying and promoting a more enjoyable and fulfilling educational experience.

## 7.2 Future Work

In our future work, we have exciting plans to enhance the game experience and provide a comprehensive platform for students. Here are the key elements we will focus on:

**Motivational Notifications:** The game will feature motivational notifications designed to inspire and encourage students. These notifications will be delivered to students on their smart devices, ensuring that they stay motivated and focused on their academic tasks. By receiving timely and uplifting messages, students will be encouraged to complete their tasks and achieve their academic goals.

**Device Compatibility:** We are committed to making the game accessible on a wide range of smart devices, ensuring that students can enjoy it on smartphones, tablets, and other mobile devices. This broad compatibility will allow students to engage with the game wherever they are, providing a seamless and convenient experience for managing their academic responsibilities.

**Team Collaboration and Communication:** To foster collaboration and social interaction among students, the game will support team play and enable conversations between students. Students will have the opportunity to form teams, communicate with each other, and work together to complete tasks. This collaborative environment will promote teamwork, communication skills, and peer support, enhancing the overall learning experience.

**Global Competition:** In addition to team collaboration, the game will offer global competition where students can compete with their peers from around the world. This competition can involve completing academic tasks, achieving high scores, or demonstrating overall academic performance. Global leaderboards and rankings will add a competitive element that motivates students to strive for excellence and compare their progress with others.

**AI Integration for Task Validation:** The integration of artificial intelligence (AI) technology will be instrumental in validating completed tasks. The AI system will ensure that tasks are accurately validated, providing a fair and reliable mechanism for tracking academic progress and rewarding students accordingly. This integration will enhance the game's credibility and provide students with valuable feedback on their academic performance.

**Progress Badges:** The game will feature a badge system that rewards students based on their academic progress and achievements. As students complete tasks, meet deadlines, and achieve milestones, they will earn badges as a symbol of their accomplishments. These badges will serve as a visual representation of progress and provide a sense of achievement and motivation.

**Additional Racing Cars and Tracks:** To keep the game exciting and engaging, we will continuously add more racing cars and tracks. As players progress and achieve goals, they will unlock new racing cars with enhanced features and gain access to new tracks, providing fresh challenges and opportunities for exploration.

**Expansion of Academic Resources:** To enrich the learning experience, we will continuously expand the game's academic resources. This includes adding new educational content, interactive lessons, and additional study materials. By providing a diverse range of resources, the game will cater to different learning styles and offer students more opportunities to engage with their academic material.

In summary, our future work aims to create an immersive and motivating game experience for students by incorporating motivational notifications, supporting team collaboration and

communication, enabling global academic competition, integrating AI for task validation, implementing progress badges, and expanding the range of academic resources. These enhancements will encourage students to stay engaged, motivated, and rewarded while effectively managing their academic responsibilities using smart devices.

## References

- [1] **“Average Gaming Playtime Report”, 2020**

<https://gamingsmart.com/average-gaming-playtime-report/> Accessed October 12, 2022

- [2] **“Video Game Addiction Statistics 2022 – How Many Addicted Gamers Are There?” Cam Adair**

<https://gamequitters.com/video-game-addiction-statistics/> Accessed October 29, 2022

- [3] **” What Is A Video Game? A Short Explainer “**

[www.thewrap.com](http://www.thewrap.com) Accessed November 2, 2022

- [4] **"تاريخ ألعاب الفيديو"**

[https://areq.net/m/%D8%AA%D8%A7%D8%B1%D9%8A%D8%AE\\_%D8%A3%D9%84%D8%B9%D8%A7%D8%A8\\_%D8%A7%D9%84%D9%81%D9%8A%D8%AF%D9%8A%D9%88.html#cn-1](https://areq.net/m/%D8%AA%D8%A7%D8%B1%D9%8A%D8%AE_%D8%A3%D9%84%D8%B9%D8%A7%D8%A8_%D8%A7%D9%84%D9%81%D9%8A%D8%AF%D9%8A%D9%88.html#cn-1)

Accessed November 10, 2022

- [5] **"تاريخ ألعاب الفيديو"**

<https://www.ibelieveinsci.com/%D8%AA%D8%A7%D8%B1%D9%8A%D8%AE-%D8%A3%D9%84%D8%B9%D8%A7%D8%A8-%D8%A7%D9%84%D9%81%D9%8A%D8%AF%D9%8A%D9%88/>

Accessed November 10, 2022

- [6] **“Video Game Addiction”**

<https://my.clevelandclinic.org/health/diseases/23124-video-game-addiction> Accessed November 19, 2022

- [7] **“The impact of video games on student GPA, study habits, and time management skills: What's the big deal? ”**

[https://www.researchgate.net/publication/303226124\\_The\\_impact\\_of\\_video\\_games\\_on\\_student\\_GPA\\_study\\_habits\\_and\\_time\\_management\\_skills\\_What's\\_the\\_big\\_deal](https://www.researchgate.net/publication/303226124_The_impact_of_video_games_on_student_GPA_study_habits_and_time_management_skills_What's_the_big_deal) Accessed November 22, 2022

- [8] **“ToDo App in C Language”**

<https://www.google.com/amp/s/www.geeksforgeeks.org/todo-app-in-c-language/amp/>

Accessed December 2, 2022

- [9] **“Gamify your life”**

<https://habitica.com/static/home>

Accessed December 2, 2022

[10] “ **Design Patterns in Game Engine Development**” , APRIL 15, 2016 HAROLD SERRANO

<https://www.haroldserrano.com/blog/design-patterns-in-game-engine-development>

Accessed December 7, 2022

[11] “ **Visible Light Spectrum**”

<https://www.vecteezy.com/vector-art/7695530-visible-light-spectrum-infrared-and-ultraviolet-electromagnetic-visible-color-spectrum-for-human-eye-vector-gradient-diagram-with-wavelength-and-colors-educational-illustration-on-white-background> Accessed December 18, 2022

[12] “ **Common game development terms and definitions | Game design vocabulary**”

Unity, <https://unity.com/how-to/beginner/game-development-terms> Accessed April 18, 2023

[13] “ **blender**”

<https://www.blender.org/> Accessed April 18, 2023