|  |
| --- |
| Team 5 |
| CO2401 Software Development |
| Group Assignment - Part 1: System Design |

|  |
| --- |
| James Gregory, Matthew Cross & Michael Byrne |

Contents

[Part A - Analysis of the requirements 2](#_Toc411449836)

[Use-Case Diagram 2](#_Toc411449837)

[Brief Overview 3](#_Toc411449838)

[Detailed Use-Cases 4](#_Toc411449839)

[1 4](#_Toc411449840)

[2 5](#_Toc411449841)

[3 5](#_Toc411449842)

[Noun Phrase Analysis 6](#_Toc411449843)

[Description 6](#_Toc411449844)

[Candidate Classes 6](#_Toc411449845)

[Domain Classes 7](#_Toc411449846)

[Domain Class Diagram 8](#_Toc411449847)

[Part B - Design Class Diagram 9](#_Toc411449848)

[Part C - Sequence Diagrams 10](#_Toc411449849)

[1 10](#_Toc411449850)

[2 10](#_Toc411449851)

[3 10](#_Toc411449852)

[Part D - State Transition Diagrams 11](#_Toc411449853)

[1 11](#_Toc411449854)

[2 11](#_Toc411449855)

[Part E - Configuration Management Plan 12](#_Toc411449856)

# Part A - Analysis of the requirements

## Use-Case Diagram



## Brief Overview

|  |
| --- |
| **Use Case:** Start Game |
| **Brief Description:** The player can start the game by selecting the main menu option “start”. This will then load up game and present the player with the select vehicle screen. |

|  |
| --- |
| **Use Case:** Select Vehicle |
| **Brief Description:** The player can choose different vehicles to use from the vehicle selection screen. The user selects the desired vehicle by using left and right, which scrolls through the available vehicles. |

|  |
| --- |
| **Use Case:** Start Race |
| **Brief Description:** The player can start the race after choosing their vehicle by selecting ready. After the player has selected ready the course will load along with the AI opponents and the player’s desired vehicle and the race will begin. |

|  |
| --- |
| **Use Case:** Vehicle Movement |
| **Brief Description:** Using the movement keys WASD and Space, the player can control the vehicle by accelerating and turning. Using W the player can accelerate, is the player lets go then the vehicle slowly decelerates. Pressing A or D turns the vehicle left or right. S moves the vehicle backwards and space is used to brake. |

|  |
| --- |
| **Use Case:** Item Pickups |
| **Brief Description:** Items are found on the course in specific locations. The player can interact with the items by colliding with them. Once a player has collided with an item it is then equipped to the player; the player can only hold one pickup at a time, if another item is picked up before it is used, then the item gets over written. |

|  |
| --- |
| **Use Case:** Use Item |
| **Brief Description:** After obtaining an item the player can then use the item. This can be done by pressing the E key. After using an item it will then trigger the items function e.g. if the item is a shield it will then spawn a shield around the player |

|  |
| --- |
| **Use Case:** End Race |
| **Brief Description:** The race ends once the player has finished the course by reaching the finish line after completing the required amount of laps. Whoever finishes first wins the race. |

|  |
| --- |
| **Use Case:** Exit Game |
| **Brief Description:** The game can be exited via the main menu by selecting exit game, this unloads all the assets and shuts down the game engine. |

## Detailed Use-Cases

### 1

|  |
| --- |
| **Use Case:** Vehicle Movement |
| **Brief Description:** Using the movement keys WASD and Space, the player can control the vehicle by accelerating and turning. Using W the player can accelerate, is the player lets go then the vehicle slowly decelerates. Pressing A or D turns the vehicle left or right. S moves the vehicle backwards and space is used to brake. |
| **Primary Actors:** Game Player |
| **Secondary Actors:** None |
| **Preconditions:** Game is running and you have started the race. |
| **Main Flow:**  1.  2.  3. |
| **Postconditions:** |
| **Alternative flows:** |

### 2

|  |
| --- |
| **Use Case:** Item Pickups |
| **Brief Description:** Items are found on the course in specific locations. The player can interact with the items by colliding with them. Once a player has collided with an item it is then equipped to the player; the player can only hold one pickup at a time, if another item is picked up before it is used, then the item gets over written. |
| **Primary Actors:** |
| **Secondary Actors:** |
| **Preconditions:** |
| **Main Flow:**  1.  2.  3. |
| **Postconditions:** |
| **Alternative flows:** |

### 3

|  |
| --- |
| **Use Case:** |
| **Brief Description:** |
| **Primary Actors:** |
| **Secondary Actors:** |
| **Preconditions:** |
| **Main Flow:**  1.  2.  3. |
| **Postconditions:** |
| **Alternative flows:** |

## Noun Phrase Analysis

### Description

The game we wish to create is a racing game set in space. Within the game you will race against AI opponents around a course. The course is made up of several checkpoints, the player must go through each checkpoint in order and then reach the finish line; the player may have to complete multiple laps of the course in order to finish the race, depending on the laps required for the selected course.

The player can select their vehicle from a menu at the start of the game, the vehicles have different strengths and weaknesses such as faster speed or faster acceleration. The AI opponents are all in control of different vehicles.

Throughout the race power-ups can be picked up by both the player and the AI to help them with the race; these power-up the user in different ways, such as:

* Give the player a weapon so they can attack players and destructible objects
* Give the player an energy shield, so they can absorb incoming damage
* Give the player a temporary boost in speed
* Give the player more health

Both static and moving hazards are also placed within the scenery environment to act as obstacles; this includes hazards such as: asteroids, space stations and aliens. The player and the AI must navigate around the hazards optimally in order to stay at the front of the race.

### Candidate Classes

|  |  |  |
| --- | --- | --- |
| Racing game  AI  Course  Checkpoint  Finish line  Lap  Race  Player | Vehicle  Strength  Weakness  Speed  Acceleration  Item  Weapon  Energy Shield | Damage  Hazard  Obstacle  Asteroid  Space Station  Alien  Scenery |

## Domain Classes

|  |  |
| --- | --- |
| **Class Name** | **Justification** |
| Player | The player class will enable movement. The player class will also hold information such as: current speed, the pickup they currently have equipped, player name, what vehicle the player is using, what lap the player is currently on. |
| AI | The AI class will be store the current pickup, the name of the AI, and the type of vehicle the AI controlled player is using. The AI will communicate with the course class to find out the location of objects and waypoints on the race track. |
| Course | The course will contain the positions of all the objects that make up the course, including obstacles, pick-ups, waypoints, checkpoints. Information about the number of laps needed to complete the course, also the positions the players are within the race. |
| Vehicle | The vehicle class contains the attributes: Max Speed, Reverse Speed, Acceleration Rate, Health and any other general vehicle stats. Vehicle is the base class as their will be many child classes for all of the different type of vehicles that can be used in game. |
| Weapon | The weapon will contain: Rate of fire, Damage, Clip Size, Ammo. This will be used as a base class for multiple weapon types. |
| Power-Up | The Power-Up class is the base of all the different type of power-ups that you can get in the game. |
| Scenery | The Scenery class contains all of the models for the surrounding area of the race course e.g. skybox, planets, background ships etc. |

## Domain Class Diagram

# Part B - Design Class Diagram

# Part C - Sequence Diagrams

## 1

## 2

## 3

# Part D - State Transition Diagrams

## 1

## 2

# Part E - Configuration Management Plan

Something to do with github and stuff here