Software Requirements Specification

for

(Unity 3D Game)

Version 1.1v

Prepared by

Group Name: "CODEX"

Rehan Mumtaz	SESE-19036	mumtaz4203271@cloud.neduet.edu.pk
Kabeer Ahmed	SESE-19028	ahmed4201750@cloud.neduet.edu.pk

Instructor: **Miss Shehnila Zardari**

Date: **20 January,2020**

Roles

Rehan Mumtaz (SE-036) and **Kabeer Ahmed (SE-028)** gathered all the requirements for their project “Car GO” which is a 3D game. To gather the requirements, they observed the gaming trends in the young generation like what type of games gain popularity and what should be the difficulty level of the game and they gathered some beautiful background sceneries to improve the UI and try their best effort to give users a best experience. Rehan Mumtaz presented the idea to associate a theme to the game and implemented the background scenery and gave player opportunity to select different locations across the globe as well as the particle effects for boosters and coins in the game. Kabeer Ahmed suggested and implemented the logic for the selection of different cars for the player ,i.e; each car have different specs based on the collection player has bought it, and also the musical theme of the game. Both the members worked together on the animations of the game. Rehan Mumtaz proposed and organized the performance requirements, external interfaces and design constraints. Kabeer Ahmed proposed the scope and perspective of the product along with all the kinds of interfaces, assumptions and dependencies of the system and the software system quality attributes. Rehan Mumtaz made the class diagram to describe the logical structure of the data while Kabeer Ahmed made the block diagram and the use-case diagram.

Contents

1 Introduction

This software requirement specification (SRS) report includes description of Car Go -3D Game Project, by team “CodeX” of students of Software Engineering Department in NED University of Engineering and Technology, Karachi, Pakistan. All the functionalities, specifications and system requirements about the project will be explained in detail in this document.

1.1 Document Purpose

This document aims to explain the requirements of our system which consists of modelling a 3D car combining it with different 3D environments to create an application that uses this 3D model to implement its functionalities.

1.2 Product Scope

The project name is “CAR GO” 3D game.”CAR GO” is because in this game cars go away from the monster track which wants to smash the car. The software will make the user to be able to play a obstacle avoiding, addictive game. While playing the game, the user’s car goes on a track and collects NOS bottles which increases its score and car will boost for a while. The end-product will run on 64-bit windows PC for the people who wants to access the product through desktop and laptop computers. As the game is addictive, the users especially of young age will definitely like the product and thus the product has a very wide scope.

1.3 Intended Audience and Document Overview

The intended audience of this document is the members of the project group and developers who are willing to implement the application explained in this document. The document will guide the developers through the implementation phase. In addition to this, the document also help the developers to see early misunderstandings, inconsistencies and possible defects of the system.
This SRS document include 3 more parts from here
Section -2 : It contain overall description of the project which includes interface requirements,use case diagrams and user characteristics
Section -3 : It contain specific requirements of the project which includes functional and behaviour Requirements.
Section -4 : It contain other non-functional requirements of the project which includes performance requirements and system quality Attributes.
In the last Appendix A – Data Dictionary

1.4 Definitions, Acronyms and Abbreviations

User : A person who plays the game in single mode as well as in multiple mode.

SRS : Software Requirements Specification

IEEE : The Institute of Electrical and Electronics Engineers

FPS : Fame per second which smooths the movements of the odjects.

Windows : Microsoft Windows, normally alluded to as Windows, is a gathering of a few restrictive graphical working framework families, which are all evolved and showcased by Microsoft.

Game Engine : Programming system intended for the creation and development of computer games.

Unity 3D :A cross-stage game creation framework created by Unity Technologies, including a game engine and incorporated advancement platform.

Scene : In unity 3D every different screen is represented as scene.

Class Diagram : A kind of static construction outline in UML that portrays the design of a framework/system by indicating the framework/system's classes, their characteristics, tasks (or techniques), and the relationship among the classes. s.

Use case Diagram : A kind of graph/figure in UML that speaks to the actor's connection with the framework/system

1.5 Document Conventions

This document follows the IEEE format in which the font used is Arial and the font size used for the paragraph is 11 and for heading is 14. The comments should be italic.

1.6 References and Acknowledgments

The resources listed below are the references that has been used during the requirements analysis; IEEE Standard Documents:

1. IEEE. (1998). IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society.

2. Retrieved January 20, 2020, from http://unity3d.com/unity

3. Retrieved January 20, 2020, from www.tapintonfc.org/tag/unity-3d

2 Overall Description

This section will give information about product perspective, product functions, constraints, assumptions and dependencies, Apportioning of Requirements.

2.1. Product Perspective

Car GO-3D gaming application is totally independent system that is not related to any other system and not a component of a larger system. This program has only one type user, so there is no functionality differences between users which means there exists only one type user interface. Car GO-3D will run on windows operating system. Moreover, it will be implemented making use of Unity3D for visualizing its features. The game seeks advantage of some built-in libraries of unity engine to include some functionalities like the scene management system and the particle effect system inside the unity engine. The game logic also uses the animation system of the unity engine to put into play all the animations of the game. The game logic interacts mainly with sub-systems and other libraries inside the unity engine but no external dependencies are on the game.

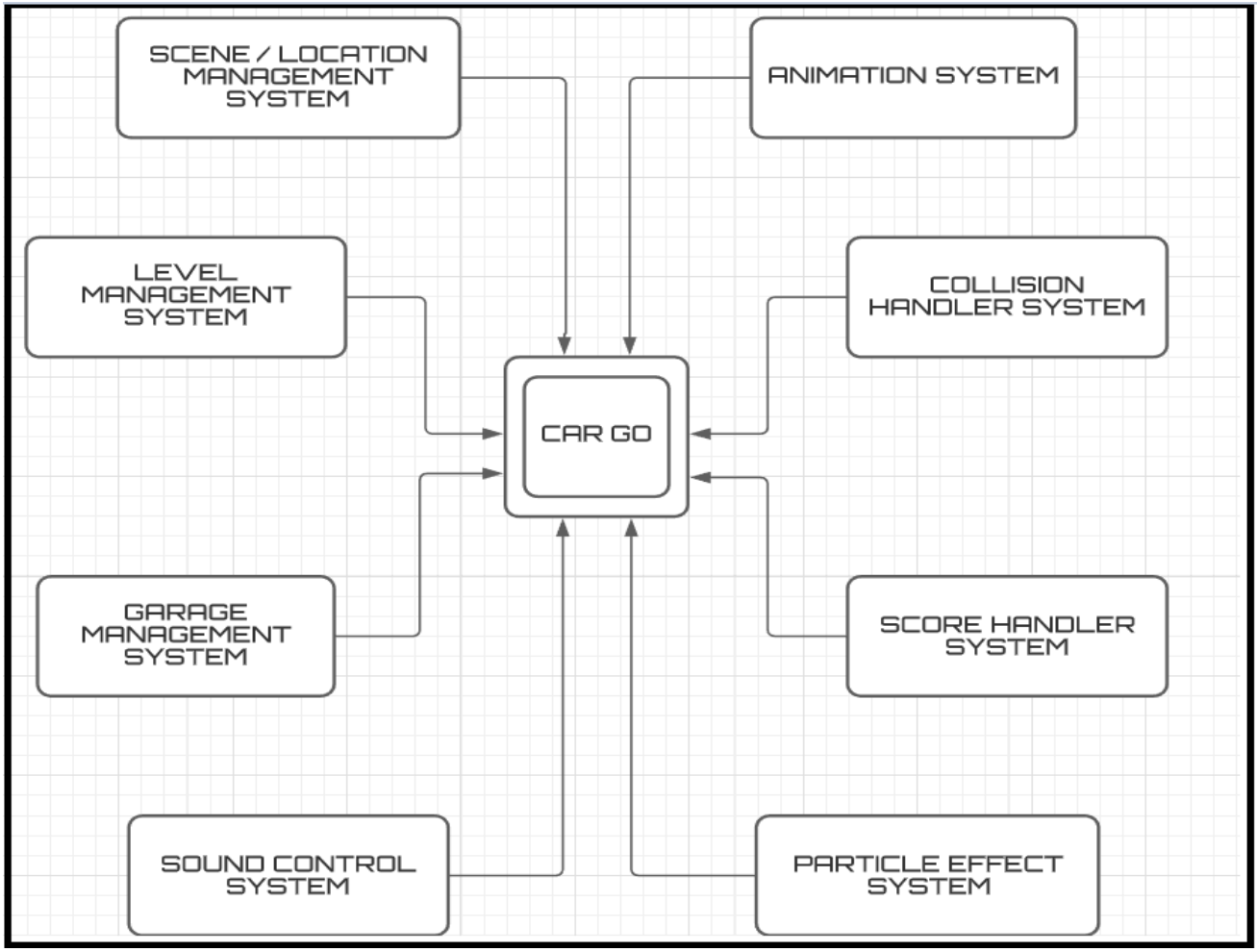


Figure 1- Block Diagram

The diagram above shows how the main system interacts with the other system and the context of the system

2.1.1 System Interface

One system requirement is the required operating system i.e: Windows 7,8.1, 10 are compatible with the game , however, 64 bit is more preferred (32-bit is also workable) The user must have a mouse and a keyboard for great user Experience other wise game wont be operable

2.1.2 User Interface

There will be one type of user. Therefore, there are no differences between users in terms of functionality, visualization and interface
Behind the user's car there is a monster car which will be chasing throughout the game, the user wanna escape and should have to look carefully at the path he is passing through.
At the beginning of the program there will be 5 options;

- START - to enter the game
- SELECT Car – To select the car to play the game with
- SELECT location – To select the city user wants to play at
- HELP - to read instructions / controls on how to play
- CREDITS - to read about the team Eliters and how they developed the game
- EXIT - to quit the game

After Pressing START, the user is required to press "Spacebar" so the car actually starts to move forward. When the car starts to move forward then there are 3 cases:

- When "up arrow key" or "W" button is activated, the car JUMPS upward.
- When "rightward key" or "D" button is activated, the car moves in RIGHT direction.
- When "leftward key" or "A" button is activated, the car moves in LEFT direction.

By moving with the help of these keys, the user can collect the coins scattered all over the path.

2.1.3 Hardware Interface

Only Personal computers, be it desktop or laptop, which have keyboard to use arrow keys or WAD keys will be suitable for the application. These devices should have some limit requirements to make the application run effectively. We expect 1.5 GHz processor, 2GB minimum internal memory & 4GB RAM minimum for computers and NVIDIA 560MX OR AMD RX 560 should be required due to the 3D high end graphics it is needed to run the game effectively without lag and obviously the pixels should not be broken. For external server if found appropriate Parse servers will be used.

2.1.4 Software Interface

Computers or Laptop will be used for the application and they must have Windows to run the application. There will be a graphical user interface for the user to select different cars and find instructions and credits of the game

2.1.5 Communication Interface

There won't be any communication required, the game is a desktop application and will work even when offline.

2.1.6 Memory Constraints

It should be keep in mind that the Game engine which will be using is the Unity 3D , as it has lots of dependencies and packages came with it as it requires some amount of memory. For performance a RAM of 4GB is required and storage of 2GB is required

2.1.7 Operations

There are not so many operational constraints. Most of the operations are visible to user if he/she navigates through the graphical user interface. Regarding the operations, user can find help menus also. After the user and system interactions, some operations will take place at the background implemented through coding, such as the score and coin increments after player collects each coin or nitro boosters.Users can buy coins via the payment method implemented at the backend which can be used to buy new cool cars

2.1.8 Site Adaptation Requirements

There won't be any site adaption requirements because the game is a desktop application and and in case the user wants to buy coins or purchasing then internet is required.

2.2 Product Functions

Use case diagram of the Car Go-3D application is revealed in Figure 2. Steps are gathered in distinct entities, the functions of which are stated in further subsections.

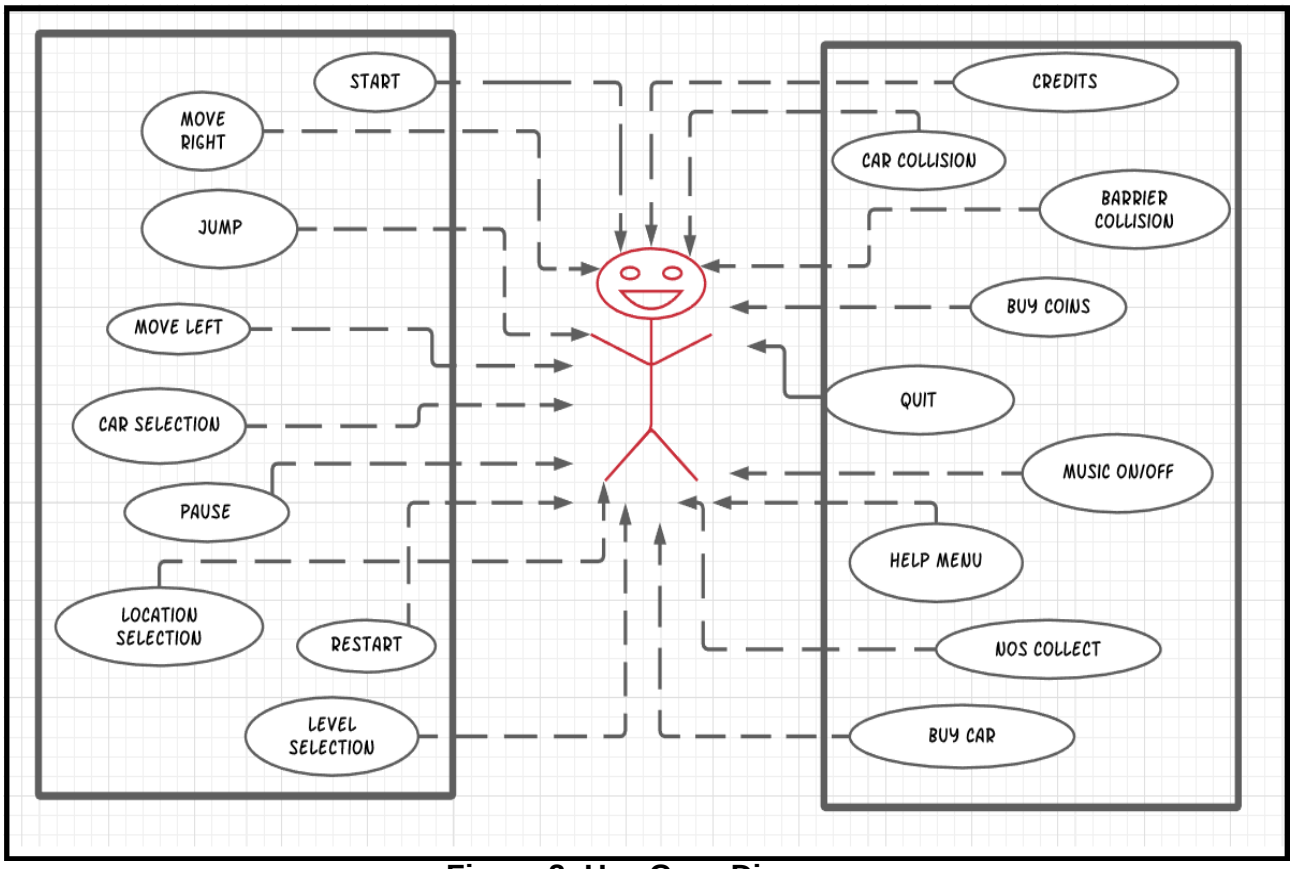
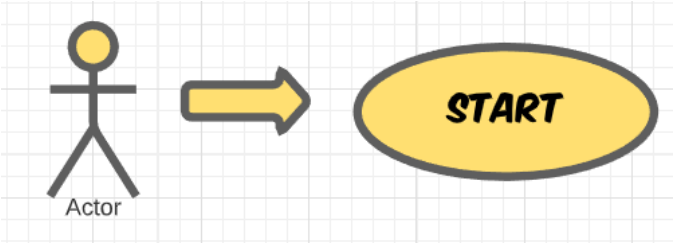


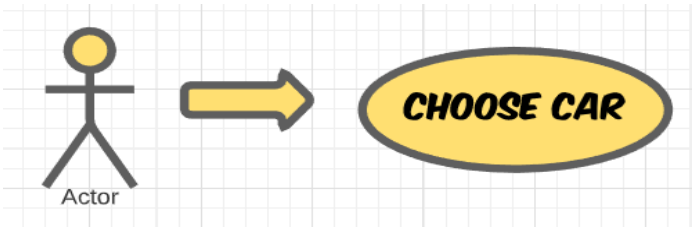
Figure 2- Use Case Diagram

2.2.1.1 Start



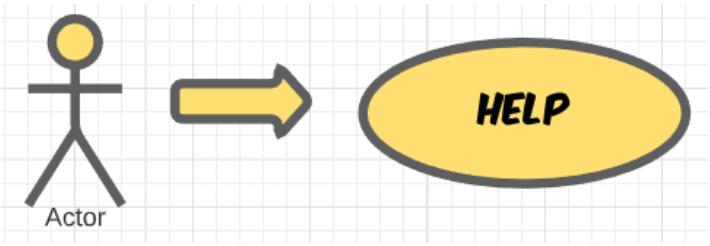
Brief Description:
This functionality becomes active just after clicking on Car Go-3D application icon or when the user enters the application. After starting user will be directed to a introductory screen and to proceed further, the user has to press CONTINUE button. Then the user will be asked to select one of the available options from the main menu. After which the user can start the game based on his/her choice of option

2.2.1.2 Choose Car



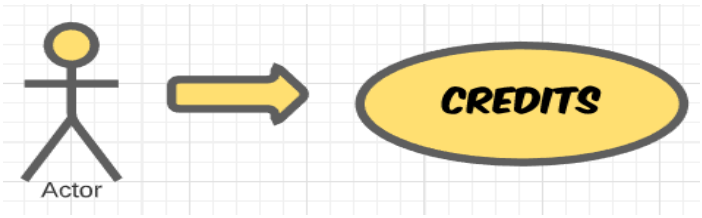
Brief Description:
This functionality becomes activated when the user presses “CHOOSE CAR” button on the Main menu. This function makes the user to choose between different cars to play with.

2.2.1.3 Help



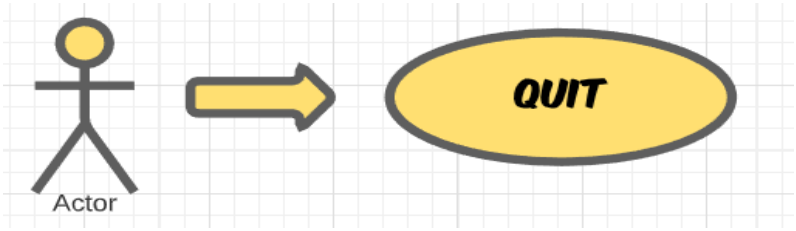
Brief Description:
This functionality becomes activated when the user presses “HELP” button on the Main menu. This function takes the user to HELP menu where the user can read instructions about.

2.2.1.4 Credits



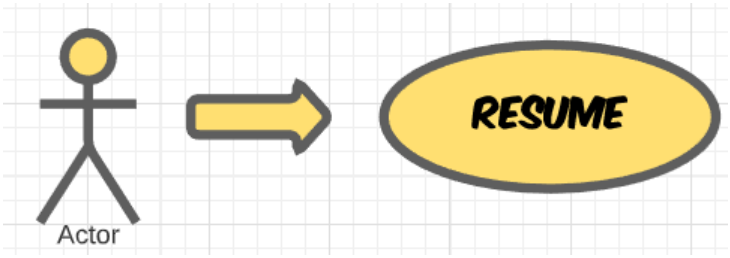
Brief Description:
This functionality becomes activated when the user presses “CREDITS” button on the Main menu. This function takes the user to CREDITS menu where the user can read about the developers of the game.

2.2.1.5 Quit



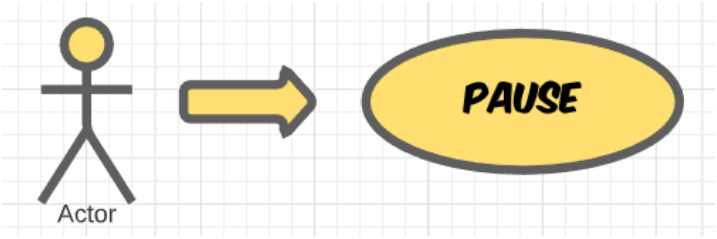
Brief Description:
This functionality becomes activated when the user presses “QUIT” button on the Main menu. This function exits the gaming application

2.2.1.6 Resume



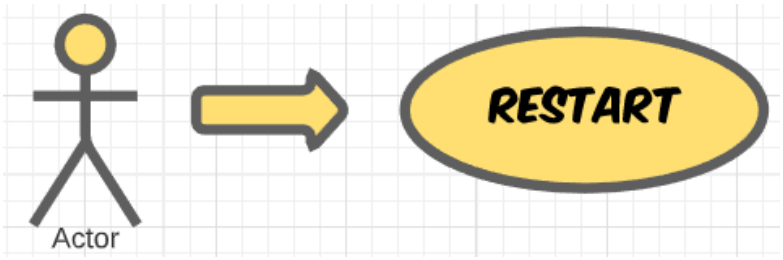
Brief Description:
This functionality becomes activated when the user clicks on “RESUME” button on the PAUSE menu. This function takes the user back to its original play. In other words it resumes the game.

2.2.1.7 Pause



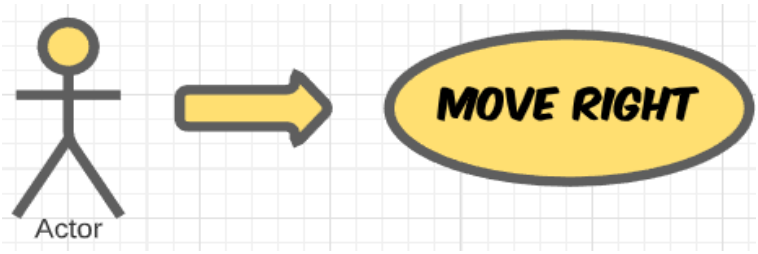
Brief Description:
This functionality becomes activated when the user clicks on the “PAUSE” button on the screen while the car is running. This function pauses the game and displays PAUSE menu.

2.2.1.8 Restart



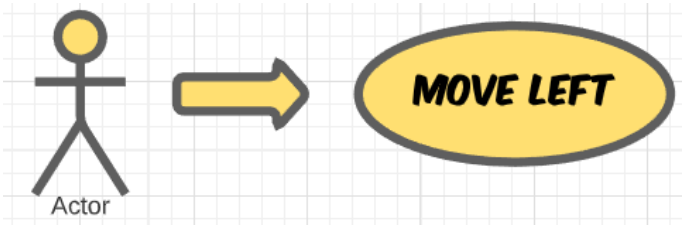
Brief Description:
This functionality becomes activated when the user clicks on “RESTART” button on the PAUSE Menu. This function restarts the level for the user**.**

2.2.1.9 Move Right



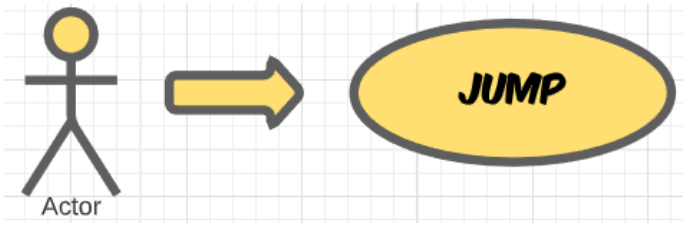
Brief Description:
This functionality becomes activated when the user press ‘rightward key’ or ‘D’ button on the keyboard. This function makes the user move right.

2.2.1.10 Move Left



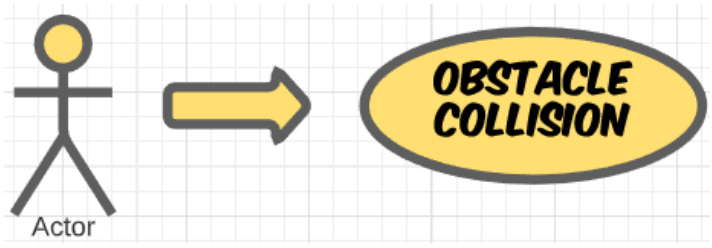
Brief Description:
This functionality becomes activate when the user press “leftward key” or “A” button on the keyboard. This function makes the user move left.

2.2.1.11 Jump



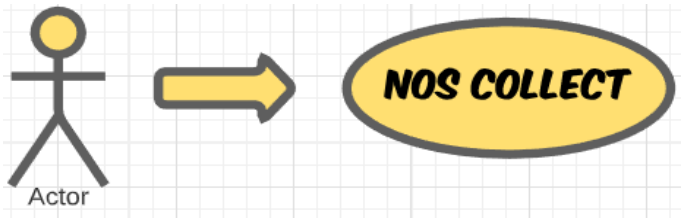
Brief Description:
This functionality becomes activate when the user press “forward key” or “W” button on the keyboard. This function makes the user Jump upward.

2.2.1.12 Obstacle Collision



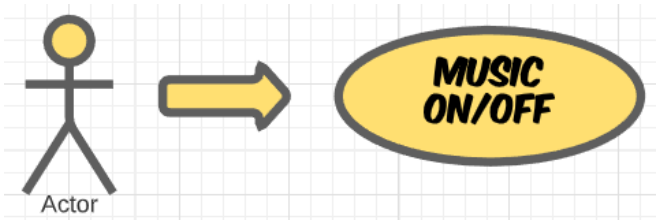
Brief Description:
This functionality becomes activated when the car collides with an OBSTACLE (Moving Cars,and Barriers). This function makes the game over for the user and the user dies thus the level restarts.

2.2.1.13 NOS Collect



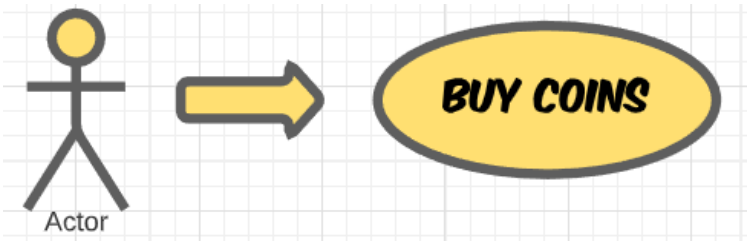
Brief Description:
This functionality becomes activated when the car touches a Nitro BOTTLE. This function makes the Cars speed fast for the user and thus it increases score of the user.

2.2.1.14 Music On/off



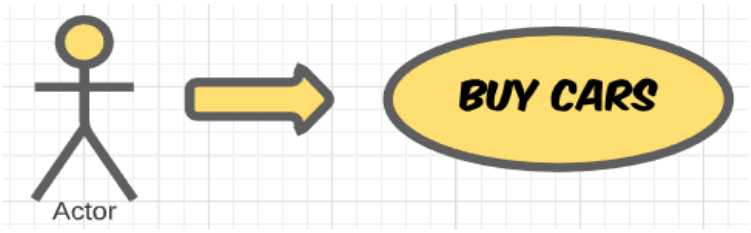
Brief Description:
This functionality becomes activated when the user wants to SWITCH ON or OFF the music

2.2.1.15 Buy Coins



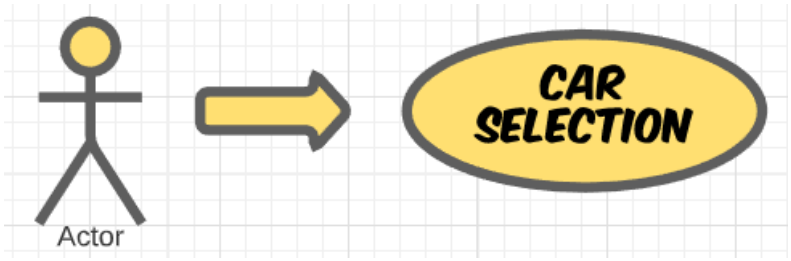
Brief Description:
This functionality becomes activated when the user wants to purchase the coin which the user can use to buy new cars and bought some cool specs for their cars

2.2.1.16 Buy Car



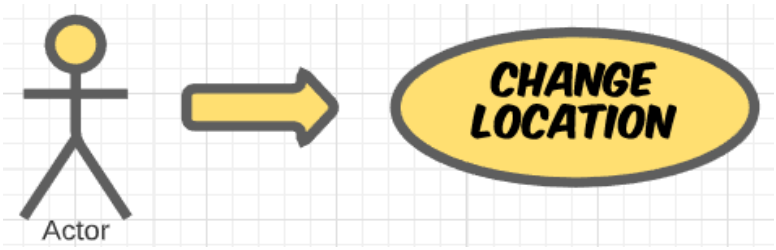
Brief Description:
This functionality becomes activated when the user wants to purchase the cars , user can buy any car of the budget he/she owns

2.2.1.17 Selection of car



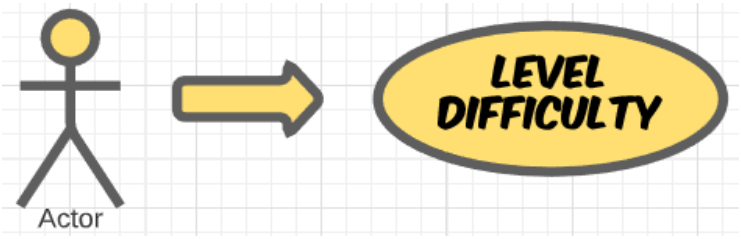
Brief Description:
This functionality offers the user to select any car from the garage he/she owns .Moreover, user can apply differenct colours and change differenct specs of the cars

2.2.1.18 Location Selection



Brief Description:
This functionality offers the user to select different locations across the globe which are unlocked only. The user will have to exprrience each map in order to unlock different locations

2.2.1.19 Level Selection



Brief Description:
This functionality lets the user to change level diffulty in order to complete the game . It offers user three levels in each map i.e: EASY , MEDIUM & DIFFICULT

2.3 Users Characteristics

The user should be familiar to using a computer and windows environment, and how to navigate to and startup an application. Since the language of the application is English, so, the user should be familiar to basic level of English language.

2.4 Constraints

The gaming application requires a computer system (Desktop or Laptop) with decent specifications as described in section 2.1.3 i.e. Hardware Interfaces. The operating System on the computer must be windows

2.5 Assumptions and Dependencies

Apart from the operating system, the game can also run on Unity 3D engine so it will be our dependency. The game runs offline so no servers are required. The users are assumed to be familiar with basic computer skills and English language to navigate to the game and also select and perform different operations inside the game.

2.6 Apportioning of Requirements

The additional requirements for future versions of this gaming application are;

- At least 10 different cars for the player to select.
- More than 10 different locations of the game with different themes
- Endless runner game mode selection option.

3 Specific Requirements

3.1 External Interface Requirements

The detailed user interface, hardware interface and software interface description is given in Section 2.1.

3.2 Functional Requirements

Following are the functional requirements of the game

- The user should be able to choose cars from a number of cars given in the game.
- The system shall provide help functionality to the user which will include instructions of how to play the game.
- The user shall also be able to see developers name using the credit button.
- The user shall be able to enable the forward movement of the car using the spacebar key.
- The user shall be able to make the car jump using the forward key or W key on the keyboard.
- The user shall be able to make the car move right using the rightward key or D key on the keyboard.
- The user shall be able to make the car move left using the leftward key or A key on the keyboard.
- The game environment shall have NoS which will increase the total score by 1000 and Coins are also increase whenever the car strike with it.
- The game environment shall have obstacles like Moving Cars,and Barriers whenever the car strike with it the game should end.
- The user shall be able to pause the game by clicking on the pause button which is always available on the screen while the game is running.
- The user should be able to resume the game, resume button will pop up when the user have paused the game.
- The user should be able to restart the game, restart button will pop up when the user have paused the game.

3.3 Behaviour Requirements

This gamen does not require any special software and platform integration. It is designed in such a way that it provides friendly environment for the users who wishes to play this game It will not affect the overall performance of the system

3.4 Logical Database Requirements

There is no database connected to Car GO but to show all the objects present in the game and the relation between these objects class diagram is used. Following class diagram represents logical structure of the data along with their attributes

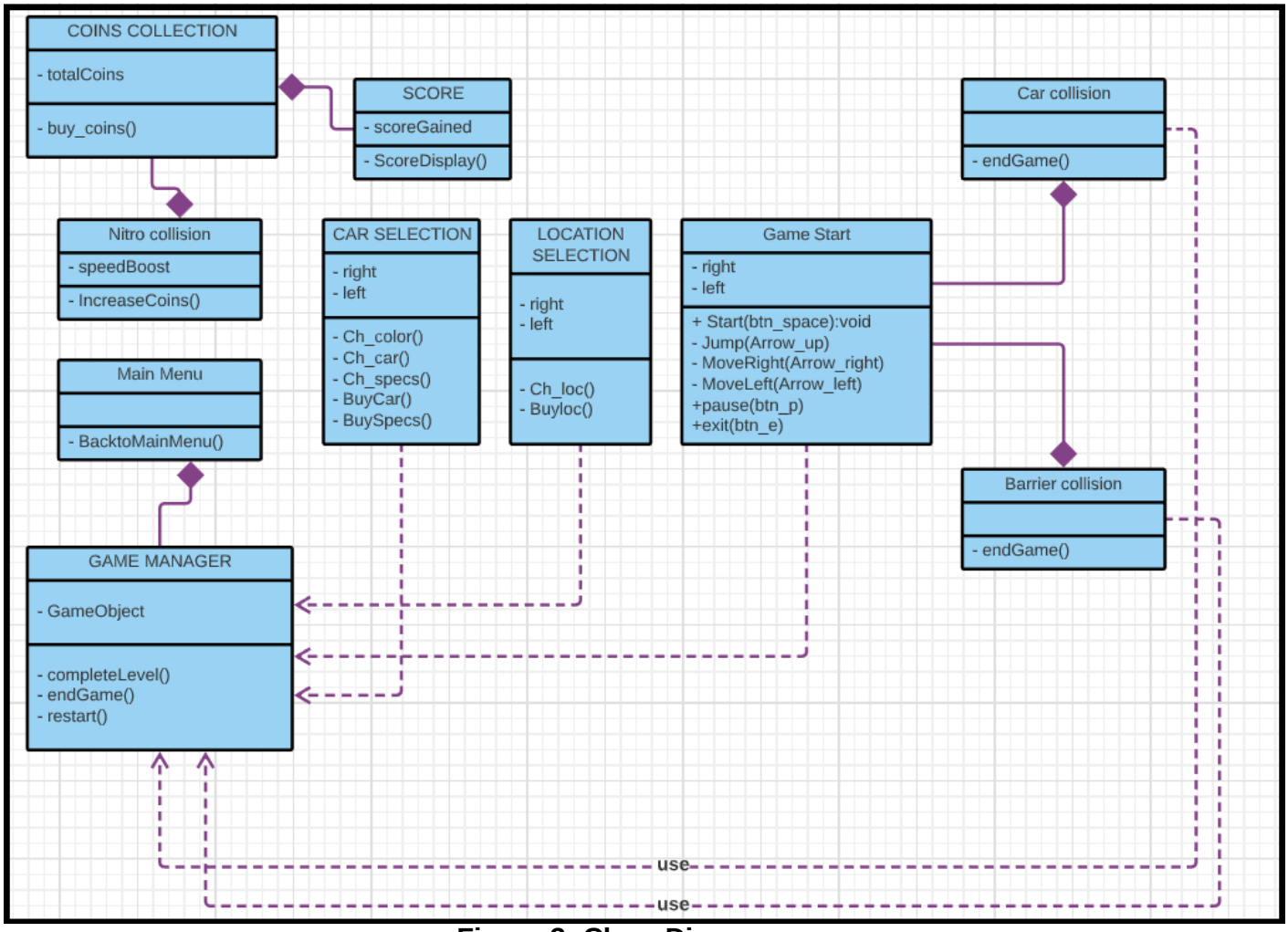


Figure 3- Class Diagram

4 Other Non-functional Requirements

4.1 Performance Requirements

Since there will be only one user playing the game in his/her local environment, the system does not need to handle multiple user case. In single player mode, the game should work at least 30 FPS, so that the game will be fluent. Loading phase of the game should not take more than 40 seconds; in other words the game should start in 40 seconds after the user opens it -this requirement is about the game itself. The game can be run at any computer which has game setup, keyboard and windows. A player who has not played the game before should learn all of its functionalities in 20 minutes. System does not require high performance.

4.2 Safety and Security Requirements

4.2.1 Security and Privacy

The game will not take or use any users data therefore there aren't any particular security and privacy requirements.

4.2.2 Safety

A warning to take a break should generate by the game after three hours of continuous play to prevent the user from eyestrain.

4.3 Software Quality Attributes

4.3.1 Portability

The game requires minimum effort in terms of installation. The game will only be playable on devices supporting windows as operating system and have keyboard to control the car.

4.3.2 Maintainability

All code related to game should be completely documented. Comments of last change and date should be included in the program files. The code shall be in modular form to allow future changes. Objects that will be use in game should not be embedded in code rather they should be stored in a separate file so that the modifications in the set of objects will be easily possible.

4.3.3 Reliability

Apart from the consequence of an operating system error the game shall never stop working, crash or hang at least with in a time of 5 hours of continuous running.

4.3.4 Availability

The game should be available to all the users who have the game installation package.

Appendix A – Data Dictionary

Following are the commonly used words throughout the SRS document

Windows : Microsoft Windows, normally alluded to as Windows, is a gathering of a few restrictive graphical working framework families, which are all evolved and showcased by Microsoft.

Game Engine : Programming system intended for the creation and development of computer games.

Unity 3D :A cross-stage game creation framework created by Unity Technologies, including a game engine and incorporated advancement platform.

Scene : In unity 3D every different screen is represented as scene.

Class Diagram : A kind of static construction outline in UML that portrays the design of a framework/system by indicating the framework/system's classes, their characteristics, tasks (or techniques), and the relationship among the classes. s.

Use case Diagram : A kind of graph/figure in UML that speaks to the actor's connection with the framework/system