



ANDROID BASED VEHICLE RACING GAME

(Neels Wheels 3D)

BY

Neelum Ayub

Registration ID: 04071313035

Supervised By

Dr. Muddassar Azam Sindhu

Session 2013-2017

Department of Computer Science

Quaid-i-Azam University

Islamabad

Revision: 0.3

JULY 18, 2017



Revision History

Revision	Description	Personnel	
0.1	Initially submitted for external demo of Project-I	Created By	Neelum Ayub
		Reviewed By	Dr. Muddassar Azam Sindhu
0.2	Submission for internal demo of Project-II	Modified By	Neelum Ayub
		Reviewed By	Dr. Muddassar Azam Sindhu
0.3	Final Submission for external demo of Project-II	Modified By	Neelum Ayub
		Reviewed By	Dr. Muddassar Azam Sindhu



QUIAD-I-AZAM UNIVERSITY
DAPARTMENT OF COMPUTER SCIENCE

Dated: July 18, 2017

Final Approval

This is to certify that we have read the project report submitted by Neelum Ayub and it is our judgement that this report is of sufficient standard to warrant its acceptance by Quaid-i-Azam University, Islamabad for the degree of Bachelor of Science in Computer Science.

COMMITTEE

1. EXTERNAL EXAMINAR

2. SUPEVISOR

Dr. Muddassar Azam Sindhu
Assistant Professor
Deptt. Of Computer Science
Quaid-i-Azam University
Islamabad

3. CHAIRPERSON

Dr. Onaiza Maqbool
Professor
Deptt. Of Computer Science
Quiad-i-Azam University
Islamabad



Neels Wheels 3D

In the name of Allah Almighty the most merciful and beneficent

In the name of Allah Almighty the most merciful and beneficent

A report submitted to the Department of computer science, Quaid-i-Azam University, Islamabad, as a partial fulfillment of the requirement for the award of the degree of Bachelor of Computer Science.



Acknowledgements

I am grateful to Allah Almighty for the opportunities and His blessings. Many thanks to my parents for their love and motivation to complete this project.

I am using this opportunity to express my gratitude to my project supervisor Dr. Muddassar Azam Sindhu, Department of Computer Science, Quaid-i-Azam University, Islamabad who supported me throughout the course of this project. I am thankful for his aspiring guidance, invaluable constructive criticism and friendly advice during the project work. I am sincerely grateful to him for sharing his truthful and illuminating views on a number of issues related to the project.

Neelum Ayub



Abstract

Game development is a particular challenge within the field of software engineering. Specific knowledge of Physics and Mathematics is necessary to achieve a satisfactory result.

The success of the digital game industry is spawning several opportunities and challenges. It motivates to develop a creative yet enjoying game. The idea behind the development of this game is provide entertainment to Android Users.

Neels Wheels 3D (the name of game) is intended to provide entertainment to users through many interesting features. It is single player game and based on Android will be used for entertainment purposes. The goal of this project is to provide a user with a leisurely but challenging gaming experience that will grab and hold their attention and interest as well as intellectually stimulate them. The game will be three-dimensional having eye catching graphics and environment. Intended audience for this game are Android users.



Table of Contents

Revision History	1
Final Approval	2
In the name of Allah Almighty the most merciful and beneficent.....	3
Acknowledgements	4
Abstract	5
Table of Contents	6
List of Tables	10
List of Figures	11
Chapter 1	12
SOFTWARE PROJECT MANAGAMENT PLAN	12
(SPMP)	12
1 Chapter 1 Software Project Management Plan	13
1.1 Introduction	13
1.2 Project Overview	13
1.3 Scope	13
1.4 Project Deliverables	13
1.5 Project Organization	13
1.5.1 Software Process Model	13
1.5.2 Roles and Responsibilities	14
1.5.3 Tools and Techniques	14
1.6 Project Management Plan	14
1.6.1 Tasks	14
1.6.2 Description	14
1.6.3 Resources Needed	18
1.6.4 Deliverables and Milestones	20
1.6.5 Dependencies and Constraints	20
1.6.6 Risks and Contingencies	20
1.6.7 Assignments	20
1.6.8 Timetable	20
1.7 Additional Material	20
Chapter 2	21
2 Chapter 2 Software Requirement Specification	22
2.1 Introduction	22
2.2 Product Overview	22



2.3	Product Scope	22
2.4	SPECIFIC REQUIREMENTS	22
2.4.1	External Interface Requirements	22
2.4.2	User Interfaces	22
2.4.3	Hardware Interface	22
2.4.4	Software Interface.....	23
2.4.5	Communications Protocols	23
2.5	Software Product Features	23
2.6	Use Case Diagram	24
2.7	Use Cases.....	25
2.8	Domain Model	34
2.9	Software System Attributes	35
2.9.1	Reliability.....	35
2.9.2	Availability	35
2.9.3	Security	35
2.9.4	Maintainability.....	35
2.9.5	Portability.....	35
2.9.6	Performance	35
2.10	Database Requirements	36
2.11	Additional Requirements.....	36
Chapter 3	37
3	Chapter 3 Software Design Description	38
3.1	Introduction	38
3.2	Purpose.....	38
3.3	Project Overview and Scope.....	38
3.4	Requirements Traceability Matrix	38
3.5	SYSTEM ARCHITECTURAL DESIGN	39
3.5.1	Chosen System Architecture	40
3.5.2	Discussion of Alternative Designs	40
3.5.3	System Interface Description	40
3.6	DETAILED DESCRIPTION OF COMPONENTS	41
3.6.1	Player	41
3.6.2	Environment.....	42
3.6.3	Obstacles.....	42
3.6.4	Levels	42
3.7	USER INTERFACE DESIGN	42
3.7.1	Description of the User Interface.....	42



3.7.2	Screen Images.....	42
3.8	Sequence Diagram.....	44
3.8.1	Start Game	45
3.8.2	Set difficulty level.....	45
3.8.3	Select racing environment.....	46
3.8.4	Select vehicle.....	46
3.8.5	Pause game	47
3.8.6	Resume game.....	47
3.8.7	Play game.....	48
3.8.8	Game sound.....	48
	48
3.8.9	Restart Game	49
3.8.10	Exit game.....	49
3.9	Class Diagram	50
3.10	Activity Diagram	51
Chapter 4.....		52
4	Chapter 4 Implementation Details	53
4.1	Introduction.....	53
4.2	Framework selection	53
4.3	Operating System selection	53
4.4	Language.....	53
4.5	Screen Shots.....	53
Chapter 5.....		56
5	Chapter 5 Software Test Documentation	57
5.1	INTRODUCTION.....	57
5.2	System Overview	57
5.3	Test Approach	57
5.4	Testing Objectives	57
5.5	TEST PLAN.....	57
5.5.1	Features to be Tested.....	57
5.5.2	Features not to be Tested	58
5.5.3	Testing Tools and Environment	58
5.6	TEST CASES.....	58
Chapter 6.....		65
6	Chapter 6 Conclusion and Future Work.....	66
6.1	Conclusions and Summary.....	66
6.2	Future Enhancements.....	66



Appendix	67
7 ADDITIONAL MATERIAL	68
7.1 References	68



List of Tables

Table 1-1: Tools and Techniques	14
Table 2-1: Use Case 1	25
Table 2-2: Use Case 2	26
Table 2-3: Use Case 3	27
Table 2-4: Use Case 4	28
Table 2-5: Use Case 5	29
Table 2-6: Use Case 6	30
Table 2-7: Use Case 7	31
Table 2-8: Use Case 8	32
Table 2-9: Use Case 9	33
Table 3-1: Requirement Traceability matrix	39



List of Figures

<i>Figure 1-1.1: Requirements and Analysis</i>	15
<i>Figure 1-2: Design Phase Tasks</i>	16
<i>Figure 1-3: Design Phase Tasks</i>	17
<i>Figure 1-4: Resources</i>	18
<i>Figure 1-5: Giant Chart of Project Plan</i>	19
<i>Figure 2-1: Use Case Diagram</i>	24
<i>Figure 2-2: Domain Model</i>	34
<i>Figure 3-1: System Architecture Design</i>	40
<i>Figure 3-2: System Interface</i>	41
<i>Figure 3-3: Start Screen Image</i>	42
<i>Figure 3-4: Select Vehicle</i>	43
<i>Figure 3-5: Select Level</i>	43
<i>Figure 3-6: Desert Environment</i>	43
<i>Figure 3-7: Snow Environment</i>	44
<i>Figure 3-8: Truck Racing</i>	44
<i>Figure 3-9: Sequence diagram of start game</i>	45
<i>Figure 3-10: Sequence diagram of select difficulty level</i>	45
<i>Figure 3-11: Sequence diagram of select environment</i>	46
<i>Figure 3-12: Sequence diagram of select vehicle</i>	46
<i>Figure 3-13: Sequence diagram of pause game</i>	47
<i>Figure 3-14: Sequence diagram of resume game</i>	47
<i>Figure 3-15: Sequence diagram of play game</i>	48
<i>Figure 3-16: Sequence diagram of game sound</i>	48
<i>Figure 3-17: Restart Game</i>	49
<i>Figure 3-18: Sequence diagram of exit game</i>	49
<i>Figure 3-19: Class diagram</i>	50
<i>Figure 3-20: Activity Diagram</i>	51
<i>Figure 4-1: Main Menu</i>	53
<i>Figure 4-2: Select Vehicle</i>	54
<i>Figure 4-3: Select environment</i>	54
<i>Figure 4-4: Select Level</i>	54
<i>Figure 4-5: City Environment</i>	55
<i>Figure 4-6: Hills Environment</i>	55
<i>Figure 4-7: Snow Environment</i>	55



Chapter 1

SOFTWARE PROJECT MANAGAMENT PLAN (SPMP)



Neels Wheels 3D

1.1 Introduction

During Google Nexus Press Event [1] 2015 the company announced that there are now 1.4 billion active Android devices worldwide. World has seen a tremendous increase in Android users. To provide leisure to the Android users there are almost 2.2 million apps available on Google Play Store [2]. Day-by-day, Android gaming world is expanding as more and more people are opting for Android platform and enjoying numerous games in their devices. This project is a continuity in this race of game development. This chapter explains project management plan of Android based vehicle racing game.

1.2 Project Overview

This is a 3D vehicle racing game. In this project, the user can have the option of selecting a vehicle for racing, user can select from among a car, a motorbike, a truck or a bus. It's totally user's choice. (S)he can have a choice of selecting different configurations of a vehicle before starting the race. It will have different difficulty stages and levels. Player can unlock new vehicles after completing level for different category/types of vehicles. Each race will have different goals and time to complete a race. Vehicles will be fully functional as automated car games, including real physics of car. Racing environment is three dimensional and gives realistic stimulation.

1.3 Scope

Neels Wheels (the name of game) is intended to provide entertainment to users through many interesting features. It is single player game and based on Android will be used for entertainment purposes. The goal of this project is to provide a leisurely but challenging gaming experience that will grab and hold their attention and interest as well as intellectually stimulate them. The scope of this chapter is to provide detailed description of project management plan for Android based vehicle racing game.

1.4 Project Deliverables

Project deliverables for this game are Software requirement Specifications (SRS), Software Project Management Plan (SPMP), Software Design Description (SDD) and Software Test Documentation (STD).

1.5 Project Organization

1.5.1 Software Process Model

For the development of this project, waterfall process model will be used. Waterfall process model is simple and easy to understand. It is easy to manage due to rigidity of the model – each phase has specific deliverables and a review process. As the requirements are well understood so waterfall model is easiest to do. It is most suitable for small projects like Android-based racing games. So I preferred waterfall process model.



1.5.2 Roles and Responsibilities

I am single developer of this project so there is no division of roles and responsibilities.

1.5.3 Tools and Techniques

Following are the tools and techniques used for this project.

Table 1-1: Tools and Techniques

Sr.	Tools and Techniques	
1	MS Word 2012	MS words is used for documentation purposes.
2	ArgoUML v.33.4	It is used for making models and class diagram.
3	Unity 3D 5.0.2	It is 3D game development platform. Used for the development of vehicle racing game.
4	MS Paint (windows 10)	Used for the editing of pictures.
5	Photoshop 18.0	It is used for making graphics of game.
6	ProjectLibre 1.6.2	Used to make project plan.
7	MonoDevelop 6.1	Used for coding of game.
8	Browser (Google Chrome)	Used to browse different models for game.

1.6 Project Management Plan

Following is the description of project management plan for this project (Android based vehicle racing game). It explains how time and resources are managed throughout the life cycle of this game.

1.6.1 Tasks

There are two phases of project plan. First is the requirement and analysis and second is the design phase of this game. In requirements and analysis phase, the major tasks are to identify requirements, define use cases, develop analysis model, develop SRS and review SRS.

In the second phase, the major tasks are develop a design using Object Oriented Approach, design mode of user input, validate input, develop models and evaluate design.

Following figure 1.1 and figure 1.2 shows tasks.

1.6.2 Description

Following is the description of major tasks of both analysis and design phases.

Requirement and Analysis

- **Identify requirements**
The main goal is to review case study and define requirements by meeting stake holders.
- **Define use cases**
Define use cases and make a use case diagram.



		Name	Duration	Start	Finish	Predecessors
1		Analysis and Requirements	28 days?	10/4/16 8:00 AM	11/10/16 5:00 PM	Hardware;People;Software
2		Identify Requirements	27 days?	10/4/16 8:00 AM	11/9/16 5:00 PM	
3		problem Definition	1 day?	10/4/16 8:00 AM	10/4/16 5:00 PM	
4		Review CaseStudy	2 days	10/4/16 8:00 AM	10/5/16 5:00 PM	Neelum Ayub
5		Define Requirements	1 day	10/6/16 8:00 AM	10/6/16 5:00 PM	Neelum Ayub
6		Meet Stakeholders	1 day	10/7/16 8:00 AM	10/7/16 5:00 PM	Neelum Ayub;Supervisor
7		Define Usecase	4 days	10/10/16 8:00 AM	10/13/16 5:00 PM	Hardware;People;Software
8		Write Usecase	2 days	10/10/16 8:00 AM	10/11/16 5:00 PM	Hardware;Ms Word;PC;Software
9		Draw Usecase Diagram	1 day	10/12/16 8:00 AM	10/12/16 5:00 PM	Hardware;PC;Software;Usecase Tool
10		Review Requirement and Usecase	1 day	10/13/16 8:00 AM	10/13/16 5:00 PM	Neelum Ayub;Supervisor
11		Develop SRS	9 days?	10/26/16 8:00 AM	11/7/16 5:00 PM	Neelum Ayub;Supervisor
12		Define Functional and Non-Functional Requirements	3 days	10/26/16 8:00 AM	10/28/16 5:00 PM	5
13		Write Use case description	1 day?	10/26/16 8:00 AM	10/26/16 5:00 PM	8
14		Review Requirements	1 day	10/31/16 8:00 AM	10/31/16 5:00 PM	Neelum Ayub;Supervisor
15		Finalize SRS	5 days	11/1/16 8:00 AM	11/7/16 5:00 PM	Neelum Ayub
16		Review SRS	2 days	11/8/16 8:00 AM	11/9/16 5:00 PM	Neelum Ayub;Supervisor
17		Meet Stakeholder	2 days	11/8/16 8:00 AM	11/9/16 5:00 PM	Neelum Ayub;Supervisor

Figure 1.1: Requirements and Analysis



		Name	Duration	Start	Finish	Predecessors
19		Design Software	41 days	11/14/16 8:00 AM	1/9/17 5:00 PM	
20		Develop Design using OOP	18 days	11/14/16 8:00 AM	12/7/16 5:00 PM	Hardware;People;Software
21		Develop Architectural Design	2 days	11/14/16 8:00 AM	11/15/16 5:00 PM	18 Designing Tool;Neelum Ayub;PC
22		Review Architectural Design	1 day	11/16/16 8:00 AM	11/16/16 5:00 PM	Neelum Ayub;Supervisor
23		Develop Interface Design	2 days	11/17/16 8:00 AM	11/18/16 5:00 PM	21 Designing Tool;Neelum Ayub;PC
24		Review Interface Design	1 day	11/21/16 8:00 AM	11/21/16 5:00 PM	Neelum Ayub;Supervisor
25		Develop Data Design	2 days	11/22/16 8:00 AM	11/23/16 5:00 PM	23 Designing Tool;Neelum Ayub;PC
26		Review Data Design	1 day	11/24/16 8:00 AM	11/24/16 5:00 PM	Neelum Ayub;Supervisor
27		Design mode of user Input	4 days	12/2/16 8:00 AM	12/7/16 5:00 PM	21 Hardware;People;Software
28		Create Input Layout	2 days	12/2/16 8:00 AM	12/5/16 5:00 PM	21 Designing Tool;Neelum Ayub
29		Avoid Errors	1 day	12/6/16 8:00 AM	12/6/16 5:00 PM	25 PC
30		Avoid Extra Steps	1 day	12/7/16 8:00 AM	12/7/16 5:00 PM	PC
31		Validate Input	1 day	12/12/16 8:00 AM	12/12/16 5:00 PM	27 Hardware;People;Software
33		Develop Analysis Models	18 days	11/15/16 8:00 AM	12/8/16 5:00 PM	2 Designing Tool;Neelum Ayub;PC
34		Create Sysytem Sequence Diagram	3.5 days	11/15/16 8:00 AM	11/18/16 1:00 PM	Designing Tool;Neelum Ayub;PC
35		Review System Sequence Diagram	1.5 days	11/23/16 8:00 AM	11/24/16 1:00 PM	34 Designing Tool;Neelum Ayub;PC
36		Create Design Class Diagram	7 days	11/25/16 8:00 AM	12/5/16 5:00 PM	34 Designing Tool;Neelum Ayub;PC
37		Review Design Class Diagram	3 days	12/6/16 8:00 AM	12/8/16 5:00 PM	36
38		Evaluate Design	14 days	12/21/16 8:00 AM	1/9/17 5:00 PM	20 Hardware;People
39		Validate Design	3 days	12/21/16 8:00 AM	12/23/16 5:00 PM	20 PC
40		Verify Design	2 days	1/5/17 8:00 AM	1/6/17 5:00 PM	20 PC
41		Review and Refine design	1 day	1/9/17 8:00 AM	1/9/17 5:00 PM	Neelum Ayub;Supervisor
42		Design Phase Completed	1 day	1/10/17 8:00 AM	1/10/17 5:00 PM	20

Figure 1.2: Design Phase Tasks



		Name	Duration	Start	Finish
1		<input type="checkbox"/> Implementation	94 days?	21/02/17 08:00	30/06/17 17:00
2		<input type="checkbox"/> 3D Environment	29 days?	21/02/17 08:00	31/03/17 17:00
3		City Environment	7 days?	21/02/17 08:00	01/03/17 17:00
4		Hills Environment	8 days?	02/03/17 08:00	13/03/17 17:00
5		Snow Environment	14 days?	14/03/17 08:00	31/03/17 17:00
6	<input type="checkbox"/>	Vehicle physics	41 days?	04/04/17 08:00	30/05/17 17:00
7		Cars Physics	5 days?	04/04/17 08:00	10/04/17 17:00
8		Bus Physics	23 days?	11/04/17 08:00	11/05/17 17:00
9		Truck Physics	16 days?	05/05/17 08:00	26/05/17 17:00
10		Bike Physics	4 days?	23/05/17 08:00	26/05/17 17:00
11		Computer Cars	2 days?	27/05/17 08:00	30/05/17 17:00
12	<input type="checkbox"/>	Levels Design	17 days?	31/05/17 08:00	22/06/17 17:00
13		Basic Level	4 days?	31/05/17 08:00	05/06/17 17:00
14		Normal Level	4 days?	06/06/17 08:00	09/06/17 17:00
15		Advance Level	2 days?	10/06/17 08:00	13/06/17 17:00
16		Game Sounds	3 days?	14/06/17 08:00	16/06/17 17:00
17		Camera View	4 days?	17/06/17 08:00	22/06/17 17:00
18		Implementation Document	2 days?	23/06/17 08:00	26/06/17 17:00
19		Integration	6 days?	23/06/17 08:00	30/06/17 17:00

Figure 1.3: Design Phase Tasks



- **Develop SRS**
Define functional and nonfunctional requirements and develop software requirement specification document. It includes all other details of product like scope, purpose and introduction.
- **Review SRS**
Review software requirement specification document.

Design Phase

- **Develop Design**
Develop architectural design and interface design using Object Oriented Approach.
- **Design mode of user input**
Create mode of input layouts for this game. Identify different inputs.
- **Validate Input**
Validate input by checking existence
- **Develop Designs**
Develop system sequence diagram and class diagram.
- **Evaluate design**
Evaluate and verify design.

Implementation Phase

- **3D Environment**
The first phase of implementation is to design 3D environment.
- **Vehicle Physics**
Apply physics to all vehicles of game.
- **Level Design**
Design difficulty levels of racing game.

1.6.3 Resources Needed

Here is the list of resources needed up to design phase. I used Unity 3D for development tool, for usecase tool I used Argo UML. For designing interfaces and GUI I used MS paint and Photoshop.

Name
<input type="checkbox"/> People
Neelum Ayub
Supervisor
<input type="checkbox"/> Software
Ms Word
Development Tool
Usecase Tool
Designing Tool
<input type="checkbox"/> Hardware
PC

Figure 1-4: Resources

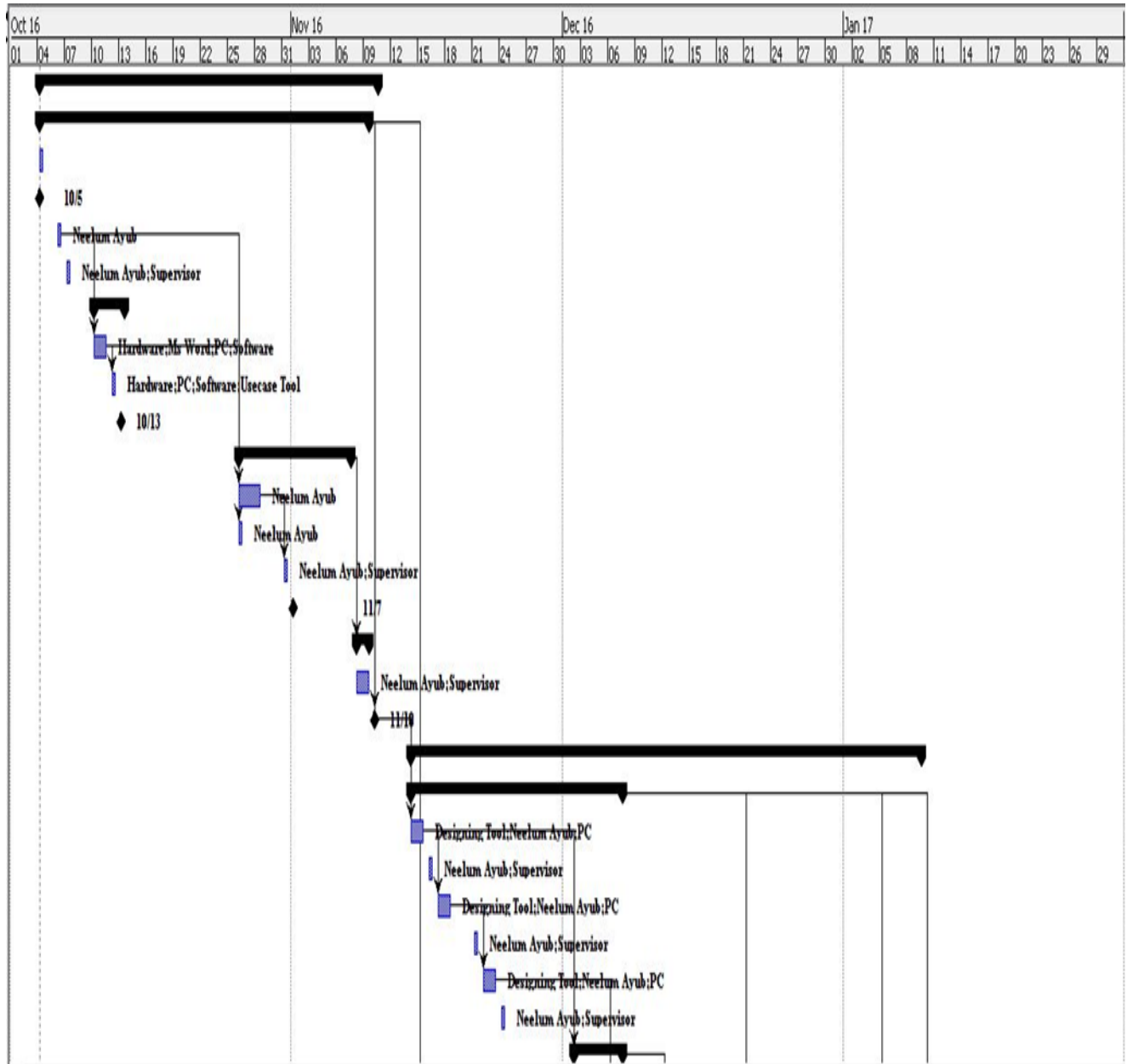


Figure 1-5: Giant Chart of Project Plan



1.6.4 Deliverables and Milestones

Deliverable and milestones are shown in figure 1.1 and figure 1.2.

1.6.5 Dependencies and Constraints

Dependencies and constraints are shown in figure 1.1 and figure 1.2.

1.6.6 Risks and Contingencies

There are no risks and contingencies.

1.6.7 Assignments

Assignments are shown in figure 1.1 and figure 1.2.

1.6.8 Timetable

All time and dates are mentioned in above figure 1.1 and figure 1.2.

1.7 Additional Material

There are no additional material.



Chapter 2

SOFTWARE REQUIRMENT SPECIFICATION (SRS)



2.1 Introduction

This chapter covers the software requirement specification of Android based vehicle racing game. World has seen a tremendous increase in Android users. To provide leisure to the Android users there are almost 2.2 million apps available on Google Play Store [2]. Day-by-day, Android gaming world is expanding as more and more people are opting for Android platform and enjoying numerous games in their devices. This project is a continuity in this race of game development.

2.2 Product Overview

This is a 3D vehicle racing game. In this project, the user can have the option of selecting a vehicle for racing, user can select from among a car, a motorbike, a truck or a bus. It's totally user's choice. (S)He can have a choice of selecting different configurations of a vehicle before starting the race. It will have different difficulty stages and levels. Player can buy new vehicles after completing certain levels. Each race will have different goals and time to complete a race. Vehicles will be fully functional as automated car games, including real physics of car. Racing environment is three dimensional and gives realistic stimulation. This is single player game, user race with computer cars in the game.

2.3 Product Scope

Neels Wheels (the name of game) is intended to provide entertainment to users through many interesting features. It is single player game and based on Android will be used for entertainment purposes. The goal of this project is to provide a user with a leisurely but challenging gaming experience that will grab and hold their attention and interest as well as intellectually stimulate them. The game will be three-dimensional having eye catching graphics and environment. Intended audience for this game are Android users.

2.4 SPECIFIC REQUIREMENTS

Following are specific requirements for interfaces like hardware or user or software.

2.4.1 External Interface Requirements

This product is an application for Android platform with a controller interface. Depending on the chosen game mode GUI items vary. Followings are the requirements for external interface.

2.4.2 User Interfaces

All interaction with user will be via the touch screen interface of Android device. As we know touch interface is most suitable to Android bases apps and games. Standard buttons and GUI texture will appear as needed in game. Touch interface is most suitable to Android apps and games. It provides ease of use to users.

2.4.3 Hardware Interface

As the game is based on Android platform, so there will be only one interface, the touch screen interface of Android device. It must be physically robust and immune to liquid damage and stains. The devices must also have good industrial design aesthetics. The devices should be fully capable android systems that can display UI elements and take input.



The touch screen input will be used to select options for the game such as level of game and mode of game. A Wi-Fi connection is also required to download and install this game on Android systems from Google Play store.

2.4.4 Software Interface

There is no need of any database in this racing game. Game is based on Android operating system. For development of this game I will use Unity 3D 5.3.1 (game development platform). To save level state or vehicle state like either it is lock or unlock, Unity provides us PlayerPrefs by using PlayerPrefs we can store our data in files. Other than this no database is used in this game. This is the first release of this game so no question to discuss previous versions. Unity 3D will be used for development of this game.

2.4.5 Communications Protocols

As the game is standalone so no communication protocol is required. But for downloading this game from Android Play Store, HTTPS protocol is required. Other than downloading from Google play store no other communication protocol is required.

2.5 Software Product Features

Here is the list of the major functions the game must perform or must let the player/user perform.

1. Start the game
User clicks on the game icon to start it.
2. Select racing vehicle of user's choice
User selects racing vehicle of his/her own choice.
3. Select racing environment/track
User selects racing environment of his/her own choice.
4. Select difficulty level
User selects a difficulty level as basic, normal or advance.
5. Press play game
After choosing desired vehicle and environment user then press play to start the race.
6. Move vehicle to avoid obstacles
User moves vehicle by using given interfaces and avoid obstacles.
7. Complete the goal for a level to unlock next level
User completes the race and unlock next level to play.
8. Unlock a new vehicle on the basis of level completed
User can unlock new level if he/she has achieved previous level.
9. Exit game
User can exit game by pressing exit button.

There will be two more functions, that is rate and about. User can rate the game on play store and user can see about page of game and developer.



2.6 Use Case Diagram

Use case diagram describes the actors, use cases and their relationship among them. It represents the user interaction and visualization of use cases. It is a graphic depiction of the interaction among the elements of system.

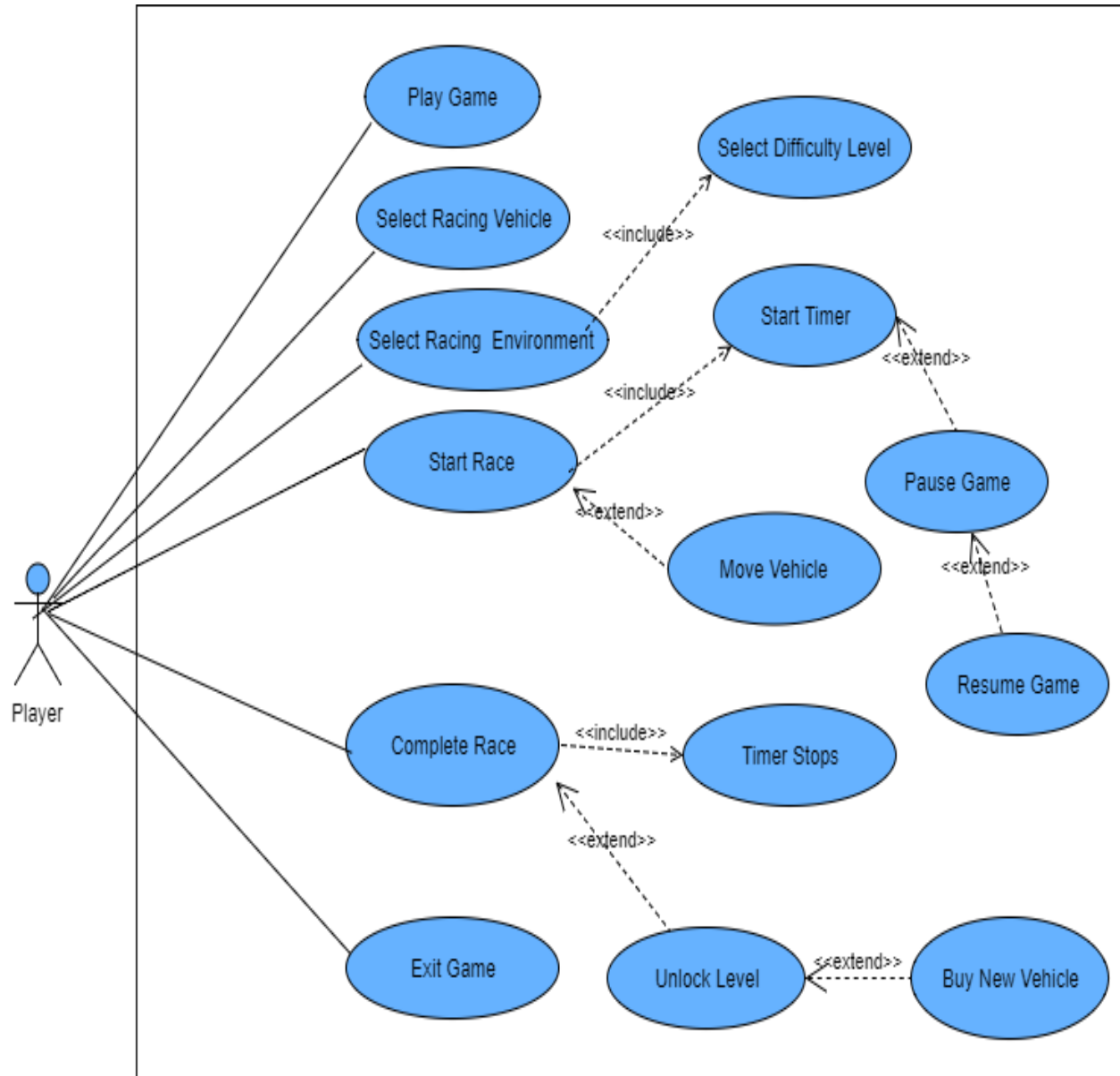


Figure 2-1: Use Case Diagram



2.7 Use Cases

Table 2-1: Use Case 1

UC-1 name: Start Game	
Primary actor	User
Stakeholder & Interests	To initialize the game to provide entertainment to users and to improve racing experience of user.
Pre-condition	<ol style="list-style-type: none">1. Make sure that the game is installed in the system correctly.2. Make sure that Android operating system is not crashed.3. Make sure that Android system is charged enough to run this game.
Post-condition	User has started the game from the given menu.
Main Success Scenario	<ol style="list-style-type: none">1. The game is started.2. Next GUI screen is displayed.
Alternate Flows	<ol style="list-style-type: none">1a. The game is not started.2a. Android operating system is crashed2b. Android system is powered off.
Special Hardware Requirements	<ol style="list-style-type: none">1. Android system2. 200MHz processor3. 32 MB of RAM4. 32 MB of storage
Technology	None
Frequency	Depends on user's choice



Table 2-2: Use Case 2

UC-2 name: Select Racing Vehicle			
Primary actor	User		
Stakeholder & Interests	To select vehicle of user's own choice for race so that game becomes more user addictive.		
Pre-condition	<ol style="list-style-type: none"> 1. Make sure that the game is installed in the system correctly. 2. Make sure that Android operating system is not crashed. 3. All racing vehicle should be displayed. 		
Post-condition	User has chosen racing vehicle of his/her own choice and interest.		
Main Success Scenario	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> User: <ol style="list-style-type: none"> 1. User selected a racing vehicle. 2. Vehicle is ready for a race. </td> <td style="width: 50%; vertical-align: top;"> System: <ol style="list-style-type: none"> 1a. System responded. 2a. System starts race. </td> </tr> </table>	User: <ol style="list-style-type: none"> 1. User selected a racing vehicle. 2. Vehicle is ready for a race. 	System: <ol style="list-style-type: none"> 1a. System responded. 2a. System starts race.
User: <ol style="list-style-type: none"> 1. User selected a racing vehicle. 2. Vehicle is ready for a race. 	System: <ol style="list-style-type: none"> 1a. System responded. 2a. System starts race. 		
Alternate Flows	<ol style="list-style-type: none"> 1a. Option is not working to select desired vehicle. 2a. Vehicles are not displaying on screen. 3a. Android operating system is crashed 		
Special Hardware Requirements	<ol style="list-style-type: none"> 1. Android system 2. 200MHz processor 3. 32 MB of RAM 4. 32 MB of storage 		
Technology	None		
Frequency	Every time user starts to play game.		



Table 2-3: Use Case 3

UC-3 name: Select Racing Environment/track					
Primary actor	User				
Stakeholder & Interests	To select racing track of user's choice to provide entertainment to users. User will have option of selecting a city track or desert etc.				
Pre-condition	<ol style="list-style-type: none"> 1. Game is started. 2. User has selected racing vehicle of his/her own choice. 				
Post-condition	Chosen racing environment/track is selected.				
Main Scenario	<table border="1"> <tr> <td>User</td> <td>System</td> </tr> <tr> <td> <ol style="list-style-type: none"> 1. User has selected a racing track. </td> <td> <ol style="list-style-type: none"> 2. System confirms. </td> </tr> </table>	User	System	<ol style="list-style-type: none"> 1. User has selected a racing track. 	<ol style="list-style-type: none"> 2. System confirms.
User	System				
<ol style="list-style-type: none"> 1. User has selected a racing track. 	<ol style="list-style-type: none"> 2. System confirms. 				
Alternate Flows	<ol style="list-style-type: none"> 1a. Option is not working to select desired racing track. 1b. The Android operating system is crashed 2a. Android system is powered off. 2b. The game is not started. 				
Special Hardware Requirements	<ol style="list-style-type: none"> 1. Android system 2. 200MHz processor 3. 32 MB of RAM 4. 32 MB of storage 				
Technology	None				
Frequency	Every time user start to play game.				



Table 2-4: Use Case 4

UC-4 name: Select Difficulty Level			
Primary actor	User		
Stakeholder & Interests	To select the difficulty and hurdle level, at initial levels user will face very small hurdles in race.		
Pre-condition	<ol style="list-style-type: none"> 1. Make sure that game is started. 2. Make sure that user has selected racing vehicle of his/her own choice. 3. Make sure user has selected racing track. 		
Post-condition	User has selected the hindrance level of game according to his/her racing experience.		
Main Success Scenario	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top;"> User <ol style="list-style-type: none"> 1. User has selected a difficulty level. </td> <td style="width: 50%; vertical-align: top;"> System <ol style="list-style-type: none"> 2. System displayed difficulty levels. </td> </tr> </table>	User <ol style="list-style-type: none"> 1. User has selected a difficulty level. 	System <ol style="list-style-type: none"> 2. System displayed difficulty levels.
User <ol style="list-style-type: none"> 1. User has selected a difficulty level. 	System <ol style="list-style-type: none"> 2. System displayed difficulty levels. 		
Alternate Flows	<ol style="list-style-type: none"> 1a. Option is not working to select desired hurdle level 1b. Android operating system is crashed. 2a. Android system is powered off. 2b. The game is not started. 		
Special Hardware Requirements	<ol style="list-style-type: none"> 1. Android system 2. 200MHz processor 3. 32 MB of RAM 4. 32 MB of storage 		
Technology	None		
Frequency	Every time user starts to play game.		



Table 2-5: Use Case 5

UC-5 name: Play Game					
Primary actor	User				
Stakeholder & Interests	To start the race with user selected vehicle and environment.				
Pre-condition	<ol style="list-style-type: none"> 1. Game is installed and running. 2. User has selected vehicle and racing environment of his/her choice. 				
Post-condition	User has pressed to play game and race is started.				
Main Success Scenario	<table border="1"> <tr> <td>User</td> <td>System</td> </tr> <tr> <td> <ol style="list-style-type: none"> 1. User selects play game option. 2. Race has started. </td> <td> <ol style="list-style-type: none"> 1b. System shows the play game option. 2b. System starts race. </td> </tr> </table>	User	System	<ol style="list-style-type: none"> 1. User selects play game option. 2. Race has started. 	<ol style="list-style-type: none"> 1b. System shows the play game option. 2b. System starts race.
User	System				
<ol style="list-style-type: none"> 1. User selects play game option. 2. Race has started. 	<ol style="list-style-type: none"> 1b. System shows the play game option. 2b. System starts race. 				
Alternate Flows	<ol style="list-style-type: none"> 1a. Option is not working to play game. 1b. Race is not started. 2a. Android operating system is crashed 2b. Android system is powered off. 				
Special Hardware Requirements	<ol style="list-style-type: none"> 5. Android system 6. 200MHz processor 7. 32 MB of RAM 8. 32 MB of storage 				
Technology	None				
Frequency	Depends on user.				



Table 2-6: Use Case 6

UC-6 name: Avoid Obstacles			
Primary actor	User		
Stakeholder & Interests	To avoid obstacles and complete race, use left, right buttons to move vehicle.		
Pre-condition	<ol style="list-style-type: none"> 1. Race has started. 2. Left right Options are working to move vehicle. 		
Post-condition	User is moving vehicle to complete the goal and to avoid hurdles.		
Main Success Scenario	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> User <ol style="list-style-type: none"> 1. User is avoiding obstacles. 2. User is turning vehicle right and left as needed. </td> <td style="width: 50%; vertical-align: top;"> System <ol style="list-style-type: none"> 1b. System is showing obstacles. 2b. System is running game. </td> </tr> </table>	User <ol style="list-style-type: none"> 1. User is avoiding obstacles. 2. User is turning vehicle right and left as needed. 	System <ol style="list-style-type: none"> 1b. System is showing obstacles. 2b. System is running game.
User <ol style="list-style-type: none"> 1. User is avoiding obstacles. 2. User is turning vehicle right and left as needed. 	System <ol style="list-style-type: none"> 1b. System is showing obstacles. 2b. System is running game. 		
Alternate Flows	<ol style="list-style-type: none"> 1a. User is unable to avoid obstacles. 1b. Race is over. 2a. Left, right Options are not working. 2b. Android operating system is crashed 2c. Android system is powered off. 		
Special Hardware Requirements	<ol style="list-style-type: none"> 1. Android system 2. 200MHz processor 3. 32 MB of RAM 4. 32 MB of storage 		
Technology	None		
Frequency	Many times		



Table 2-7: Use Case 7

UC-7 name: Unlock Next Level					
Primary actor	User				
Stakeholder & Interests	Avoiding hurdles and completing race, to unlock next level.				
Pre-condition	<ol style="list-style-type: none"> 1. User is playing a race. 2. User has completed goal. 				
Post-condition	User has unlocked next level.				
Main Success Scenario	<table border="0"> <tr> <td style="vertical-align: top;">User</td> <td style="vertical-align: top;">System</td> </tr> <tr> <td> <ol style="list-style-type: none"> 1. User has completed race 2. User unlocked next level. </td> <td> <ol style="list-style-type: none"> 1b. System is showing next levels. 2b. System stores state of level. </td> </tr> </table>	User	System	<ol style="list-style-type: none"> 1. User has completed race 2. User unlocked next level. 	<ol style="list-style-type: none"> 1b. System is showing next levels. 2b. System stores state of level.
User	System				
<ol style="list-style-type: none"> 1. User has completed race 2. User unlocked next level. 	<ol style="list-style-type: none"> 1b. System is showing next levels. 2b. System stores state of level. 				
Alternate Flows	<ol style="list-style-type: none"> 1a. User hasn't completed race. 1b. User is unable to unlock next level. 2b. Option is not working to select next level. 				
Special Hardware Requirements	<ol style="list-style-type: none"> 1. Android system 2. 200MHz processor 3. 32 MB of RAM 4. 32 MB of storage 				
Technology	None				
Frequency	Many times.				



Table 2-8: Use Case 8

UC-8 name: Unlock A New Vehicle	
Primary actor	User
Stakeholder & Interests	User can unlock a new vehicle on the basis of goal achieved.
Pre-condition	<ol style="list-style-type: none">1. User wants to unlock a new vehicle.2. User has unlock certain levels.
Post-condition	User has bought a new vehicle for race.
Main Success Scenario	<ol style="list-style-type: none">1. User has bought new vehicle.2. New vehicle is unlocked now.
Alternate Flows	<ol style="list-style-type: none">1a. User is not able to unlock a new vehicle.2a. User didn't completed level.2b. Race is over.
Special Hardware Requirements	<ol style="list-style-type: none">1. Android system2. 200MHz processor3. 32 MB of RAM4. 32 MB of storage
Technology	None
Frequency	Every time user start to play game.



Table 2-9: Use Case 9

UC-9 name: Exit game							
Primary actor	User						
Stakeholder & Interests	To exit and close game.						
Pre-condition	Game has a working exit button.						
Post-condition	User has exited the game.						
Main Scenario	<table border="1"><tr><td>User</td><td>System</td></tr><tr><td>1. User wants to exit.</td><td>1b. System shows exit option.</td></tr><tr><td>2. User choses exit option.</td><td>2b. System exits the game.</td></tr></table>	User	System	1. User wants to exit.	1b. System shows exit option.	2. User choses exit option.	2b. System exits the game.
User	System						
1. User wants to exit.	1b. System shows exit option.						
2. User choses exit option.	2b. System exits the game.						
Alternate Flows	1a. Option is not working to exit game. 2a. Android operating system is crashed.						
Special Hardware Requirements	<ol style="list-style-type: none">1. Android system2. 200MHz processor3. 32 MB of RAM4. 32 MB of storage						
Technology	None						
Frequency	Every time user want to quit game.						



2.8 Domain Model

Domain model focus on semantics, provides a glossary of items. It is the foundation for workflow modeling. The domain model represents the ‘things’ that exists or events that transpire in the business environment [2].

Here is the domain model of the game.

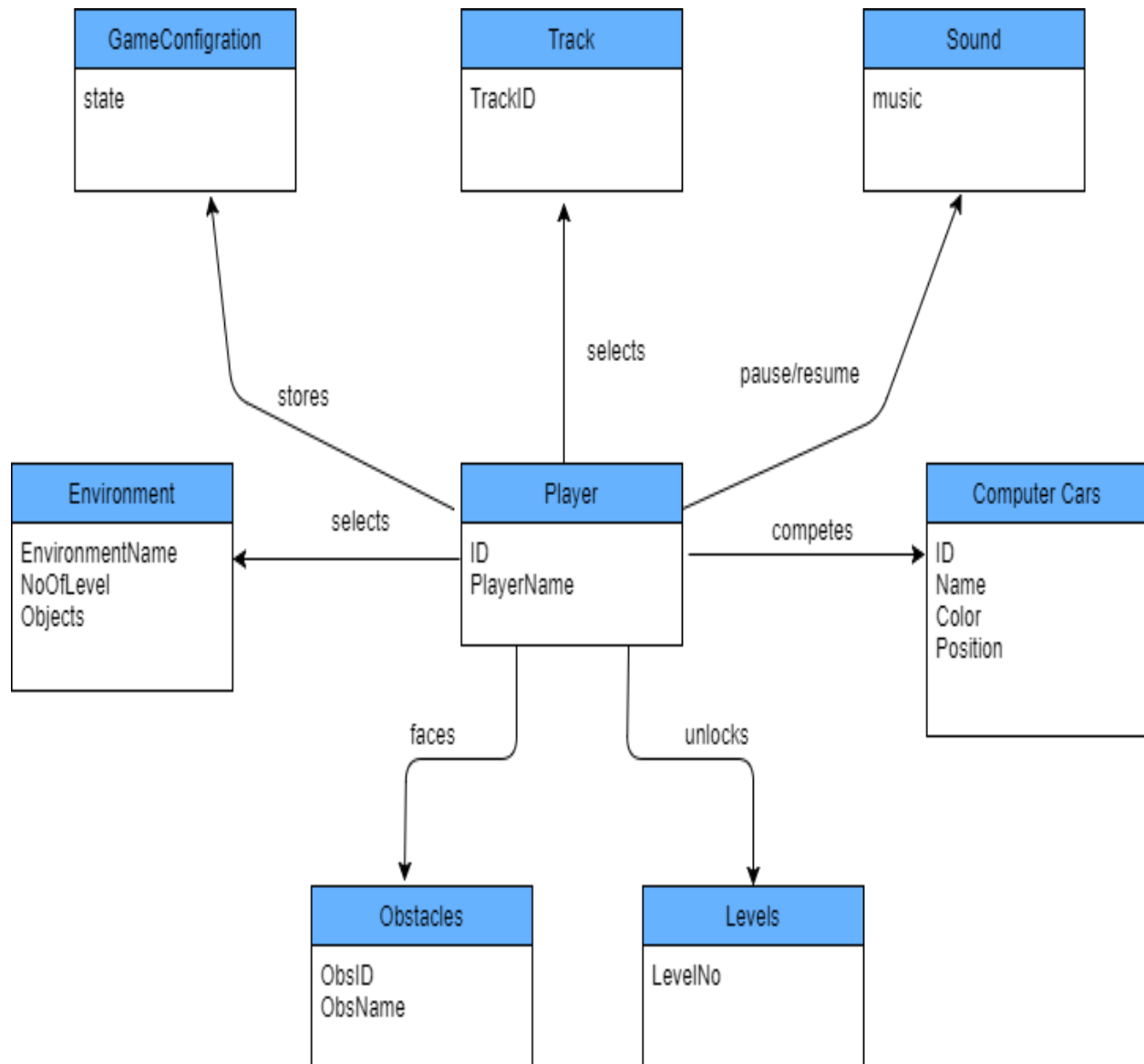


Figure 2-2: Domain Model



2.9 Software System Attributes

Software system attributes define overall factors that affect run-time behavior, game design, and user experience. To develop high quality game, software system attributes are the benchmarks that describe system's intended behavior within the environment for which it was built. Here is detail of some software system attributes.

2.9.1 Reliability

The system should never crash or hang, other than as the result of an operating system error. On the scale of software reliability, Vehicle racing game is 95% reliable.

2.9.2 Availability

Once the game is downloaded and installed, game will not take more than 10 seconds to reach the main menu upon launching. Load time between levels is almost instantaneous.

2.9.3 Security

There is no such security constraints. Because this project is a racing game and in this racing game there is not any confidential data to keep it secure.

2.9.4 Maintainability

All program files shall include comments concerning authorship and date of last change. The code should be modular, to permit future modifications. Anticipated updates include changes to the sets of objects and their descriptions used during the game. These should be stored in a separate data file, rather than embedded in the program code. Game models and assets are fully maintainable and easy to access.

2.9.5 Portability

The system should be portable to any other android system. No other specific portability requirements have been identified.

2.9.6 Performance

Following are the performance requirements of this game

2.9.6.1 Frame rate:

VRG (Vehicle Racing Game) shall execute graphics at least 20 fps (frames per second).

2.9.6.2 Processing Power:

VRG will only require a 20 MHz processor to run smoothly.

2.9.6.3 Executable Size:

The size of VRG executable will be less than one gigabyte.

2.9.6.4 In-game Load Times:

VRG's load time between levels will be near instantaneous.

2.9.6.5 Launch Time:

The time to reach the main menu upon launching the game will not exceed 10 seconds. And the time to switch between scenes will also not exceed 10 seconds.



2.10 Database Requirements

Vehicle racing game doesn't require any database. To keep the record of number of levels unlock and number of vehicles unlock, unity 3D provides us a mechanism of playerprefs by using playerprefs unity can store data in file which can be loaded upon launching the game.

2.11 Additional Requirements

There are additional requirements needed.



Chapter 3

SOFTWARE DESIGN DESCRIPTION (SDS)



3.1 Introduction

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information. It includes the description of how the software will meet the requirements.

3.2 Purpose

The purpose of the Software Design Document is to provide a description of the design of Vehicle Racing Game system to allow for software development to proceed with an understanding of what is to be built and how it is expected to build. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

3.3 Project Overview and Scope

This is a 3D vehicle racing game. In this project, the user can have the option of selecting vehicle for racing, user can select from among a car, a motorbike, a truck or a bus. It's totally user's choice. Instead of downloading different racing games user can enjoy all at one place. (S)He can have a choice of selecting different configurations of a vehicle before starting the race. By configuration means user can select vehicle of desired speed and engine capacity. It will have different difficulty stages and levels. Player can buy new vehicles after completing level. Each race will have different goals and time to complete a race. Vehicles will be fully functional as automated car games, including real physics of car. Racing environment is three dimensional and gives realistic stimulation. Intended audience are users of all age group as it is a racing game. I named this game as Neels Wheels 3D. Other than providing entertainment to users the purpose of this game is to learn throughout the phase of project development.

3.4 Requirements Traceability Matrix

Requirement Traceability Matrix or RTM captures all requirements proposed by the client or development team and their traceability in a single document delivered at the conclusion of the life-cycle.

In other words, it is a document that maps and traces user requirement with test cases. The main purpose of Requirement Traceability Matrix is to see that all test cases are covered so that no functionality should miss while testing. The Matrix is created at the very beginning of a project as it forms the basis of the project's scope and deliverables that will be produced. The Matrix is bi-directional, as it tracks the requirement forward by examining the output of the deliverables and backward by looking at the business requirement that was specified for a particular feature of the product or app.



It is used to track all the requirements and whether or not they are being met by the current process and design.

Requirements Traceability Matrix

Project Name:		Vehicle Racing Game (VRG)								
Project Description:		It is 3D Android based racing vehicle game.								
Requirements		RQ1	RQ2	RQ3	RQ4	RQ5	RQ6	RQ7	RQ8	RQ9
Use Cases	Total									
TC1	4	X		X	X		X			
TC2	2	X			X					
TC3	3		X					X		X
TC4	2			X			X			
TC5	2				X				X	
TC6	4	X			X	X	X			
TC7	2		X					X		
TC8	4				X		X		X	X
TC9	3			X				X		X
TC10	2	X							X	

Table 3-1: Requirement Traceability matrix

3.5 SYSTEM ARCHITECTURAL DESIGN

System Architecture Diagram is used to represent the components of system and interaction between them. Interaction between components of our system is shown in diagram. Double arrow line represents the interaction from both sides. Similarly single arrow represents one way interaction.

A system architecture is a conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture design of Neels Wheels 3D is shown in figure below. It shows how player interact with environment and obstacles of game.



Dashed arrow shows two way connection while box shows entities.

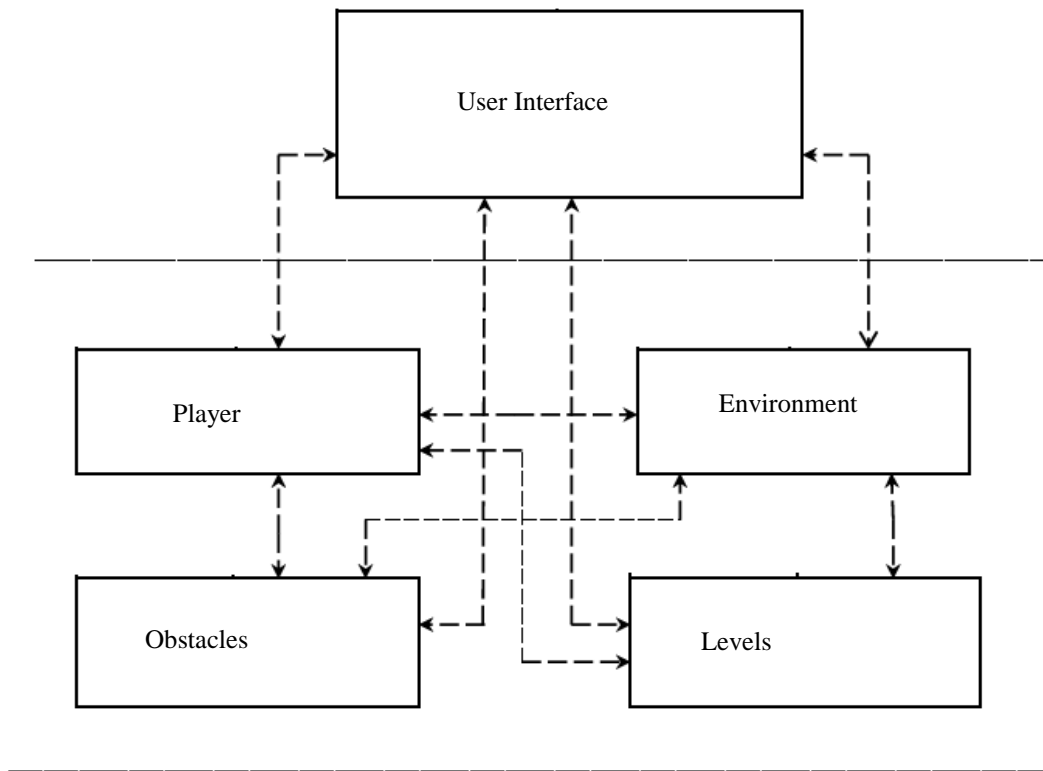


Figure 3-1: System Architecture Design

3.5.1 Chosen System Architecture

Chosen system architecture is 2-tier. Clients/users can download this game from Google server (Google Play store). Client have fully functional game installed in Android system. Game is available on play store for free.

3.5.2 Discussion of Alternative Designs

An alternative design was to add database (for storing unlock levels and vehicles of game) and to make it a 3-tier application. But this is too much waste of resources and time consuming. As unity 3D (the gaming engine) provides us an easier method to save the data of game by using playerprefs mechanism. So I rejected this design due to the time and resources constraints and continued the above one.

3.5.3 System Interface Description

System interface describes the flow of resources. It is the logical characteristics of each interface between the software product and the hardware components of the system. Figure shows the



software interface of vehicle racing game. It clearly shows how different entities of game are interacting with each other.

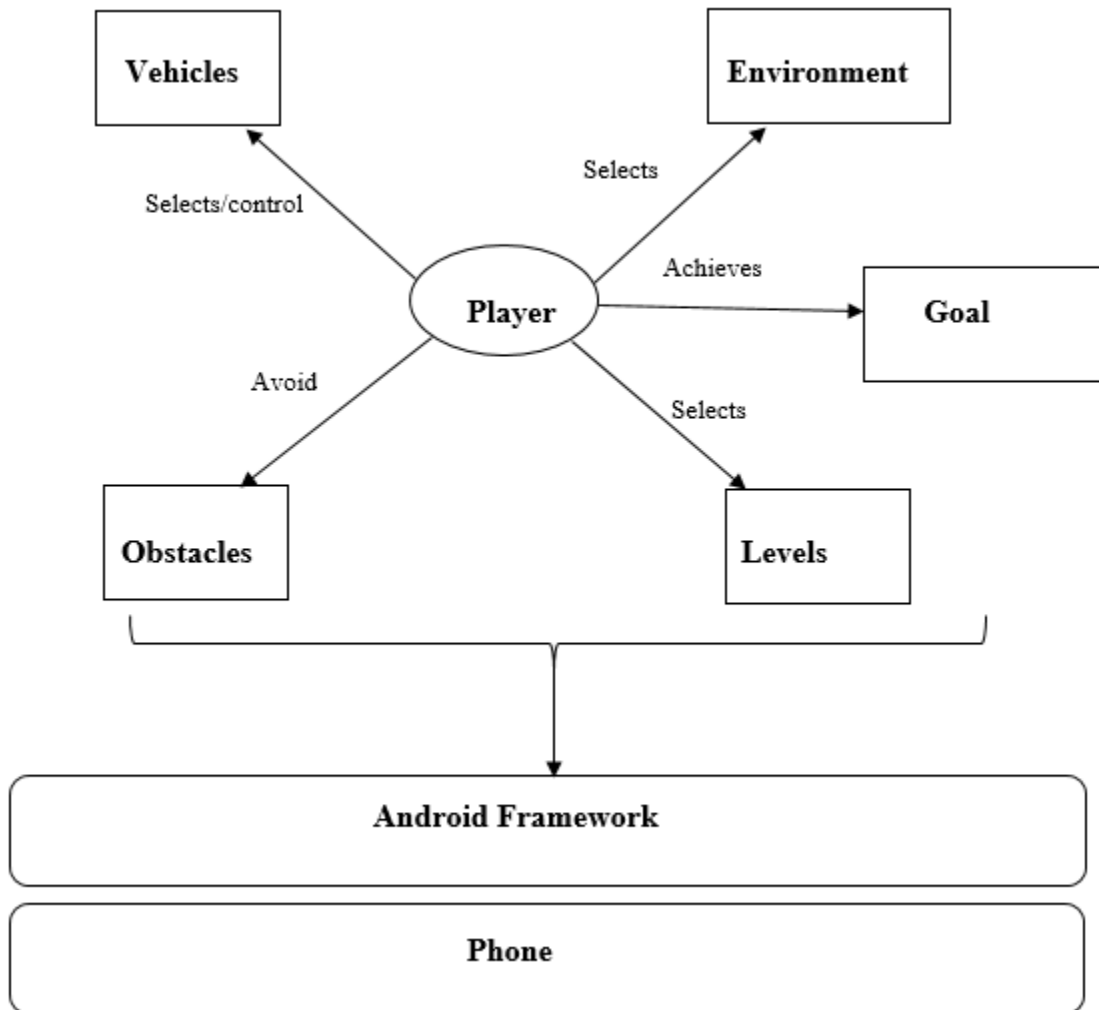


Figure 3-2: System Interface

3.6 DETAILED DESCRIPTION OF COMPONENTS

Here is the detailed description of components of system architecture. Like player, environment, obstacles and etc.

3.6.1 Player

Player or user can simulate his/her vehicle. Player competes the race with traffic cars of the game by avoiding obstacles and achieving milestone within time. Player controls game by using graphical interface and touch interface of game. Player interact with other obstacles of game by using unity physics. Player can select vehicle of his/her own choice. Player can compete his/her vehicle with other computer vehicles in the game.



3.6.2 Environment

Player interacts with environment. Game will have many different environments for example city, snow or hills. Player can select according to his/her interest. Each environment is on the new scene in unity so when user selects it switches between scenes.

3.6.3 Obstacles

The basic idea of the game is to race your vehicle by avoiding different obstacles during race. For example other traffic cars or blocked road or traffic cones and blocks. Each 3D model has collider attached with it so it detect collision by trigger collider.

3.6.4 Levels

Game has different levels, player can select according to his/her efficiency/progress in game. For example basic normal and advance levels

3.7 USER INTERFACE DESIGN

User interface is the logical characteristics of each interface between the software product and its users. In this section user interface of vehicle racing game is discussed.

3.7.1 Description of the User Interface

In vehicle racing game user can interact with game by using touch screen interface of Android device. When user clicks on game icon a splash screen will appear for 5 seconds and then main menu will appear. Main menu has two buttons Play and Exit. When player will click on play a new screen will appear for selection of vehicle. User will select his/her own choice of vehicle and clicks Next button. Now second screen will appear to select racing environment of user's choice. User can select any racing environment like city or desert. Then level screen will appear to choose level and click on "Go" button to start race.

During race user can stop timer and resume timer. If level failed user can reload level or can go to main menu. If level achieved user can go to next level or can go to main menu.

3.7.2 Screen Images

Following are few screen images of game.



Figure 3-3: Start Screen Image



Figure 3-4: Select Vehicle



Figure 3-5: Select Level



Figure 3-6: Desert Environment



Figure 3-7: Snow Environment



Figure 3-8: Truck Racing

3.8 Sequence Diagram

Sequence diagram depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the system [1]. I identify how the function are carried out in the system. So this can help in making the different functionalities in the game. We carried out sequence diagram in each use case.

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.



3.8.1 Start Game

Player clicks on game icon.

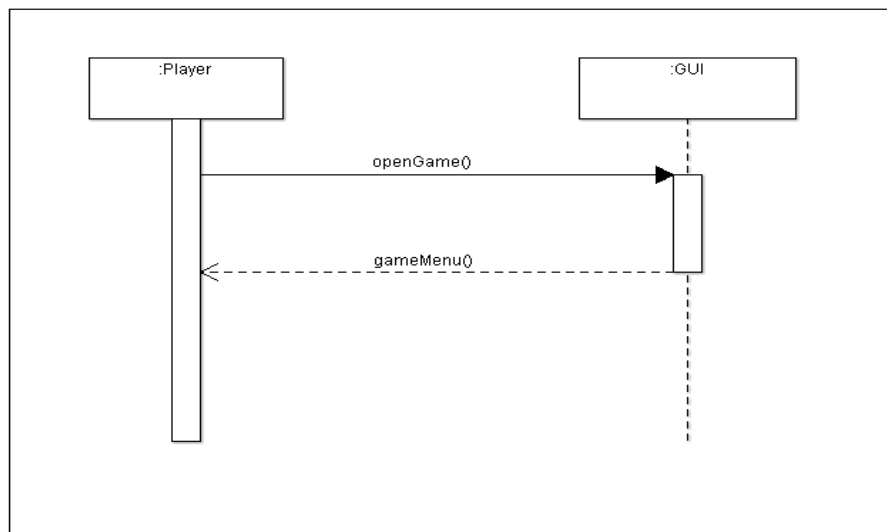


Figure 3-9: Sequence diagram of start game

3.8.2 Set difficulty level

Player selects difficulty level of his/her own choice.

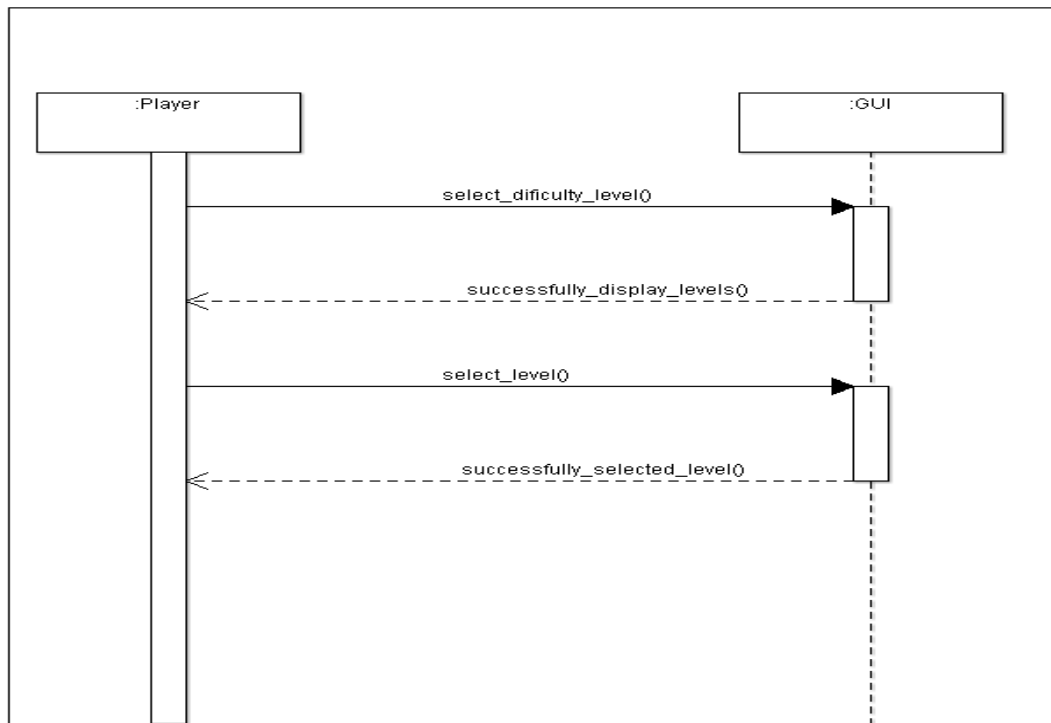


Figure 3-10: Sequence diagram of select difficulty level



3.8.3 Select racing environment

Player selects racing environment of his/her own choice.

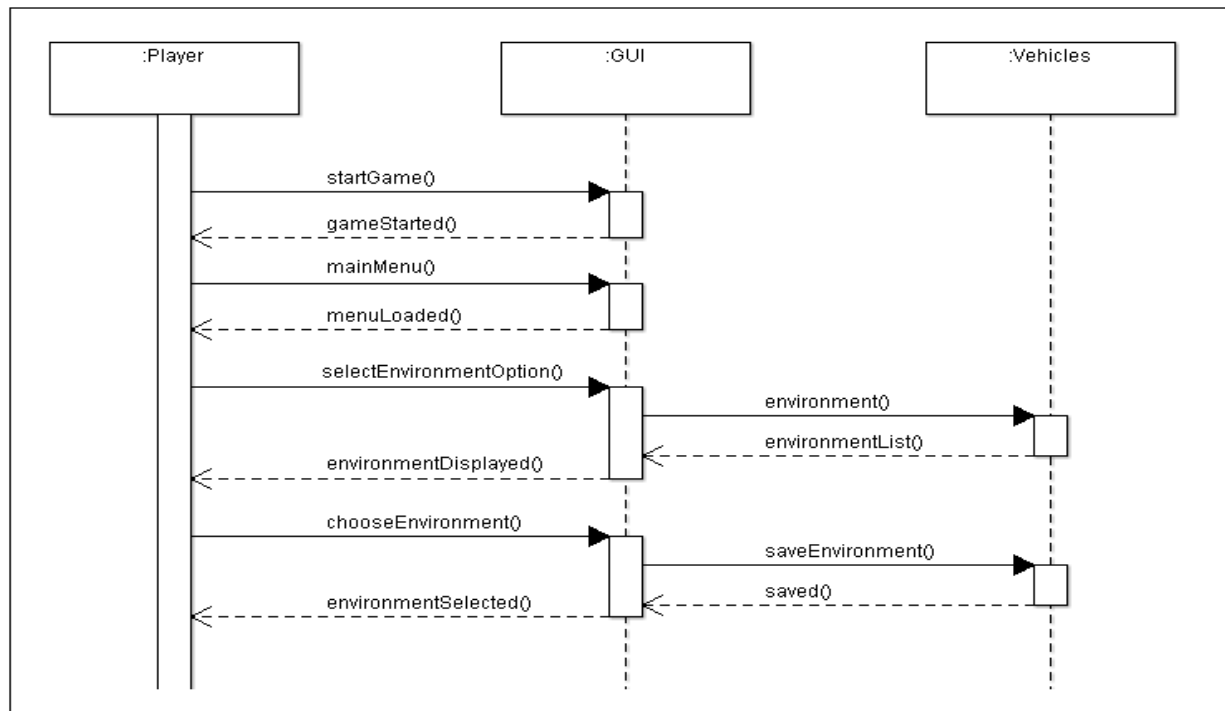


Figure 3-11: Sequence diagram of select environment

3.8.4 Select vehicle

Player selects vehicle of his/her own choice

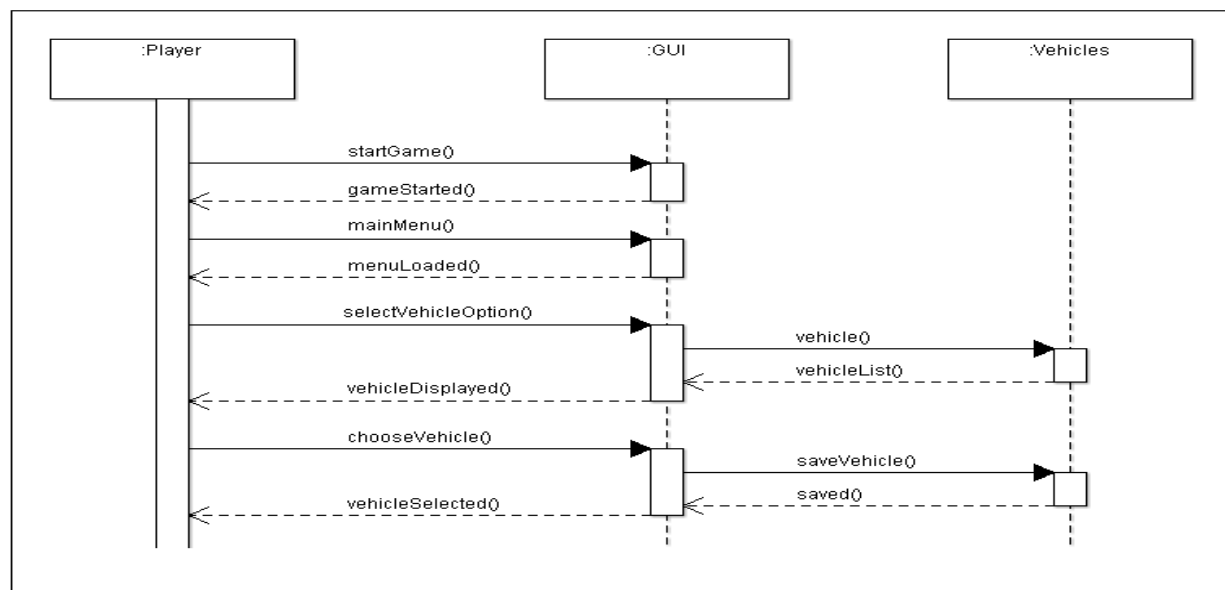


Figure 3-12: Sequence diagram of select vehicle



3.8.5 Pause game

Player pauses game by clicking on pause button.

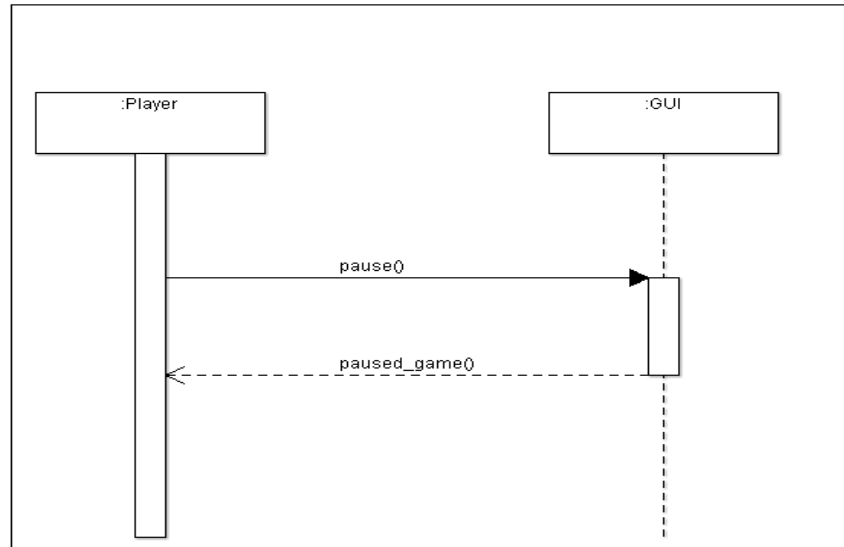


Figure 3-13: Sequence diagram of pause game

3.8.6 Resume game

Player resumes game by clicking on resume button.

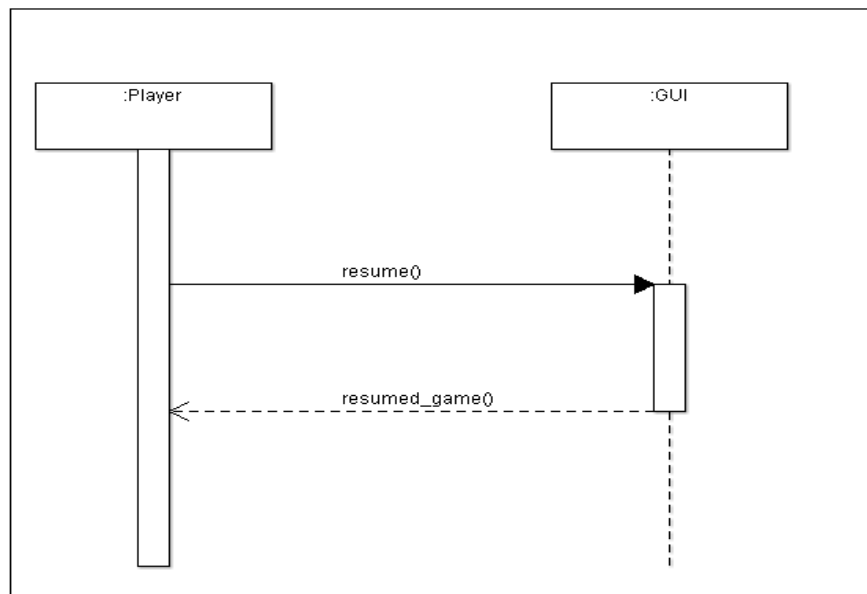


Figure 3-14: Sequence diagram of resume game



3.8.7 Play game

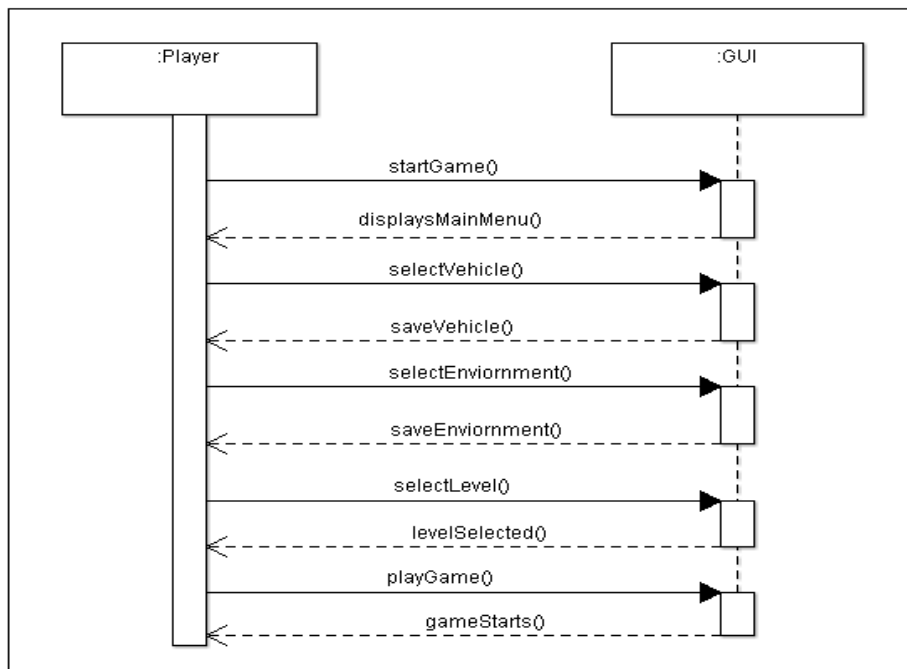


Figure 3-15: Sequence diagram of play game

3.8.8 Game sound

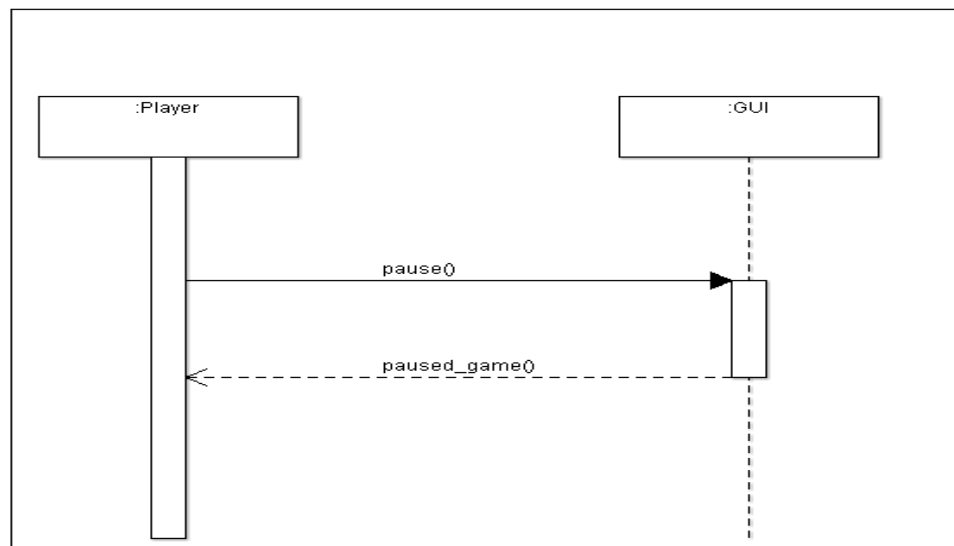


Figure 3-16: Sequence diagram of game sound



3.8.9 Restart Game

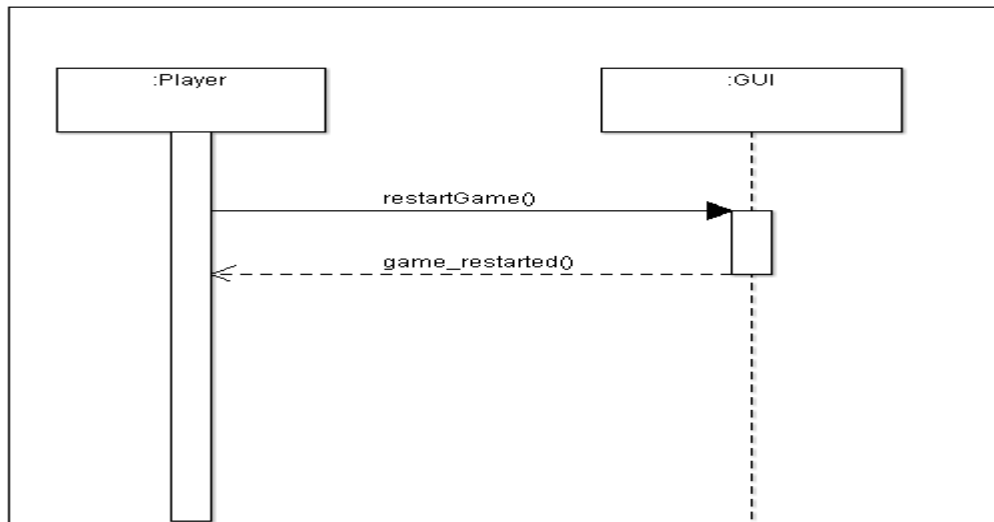


Figure 3-17: Restart Game

3.8.10 Exit game

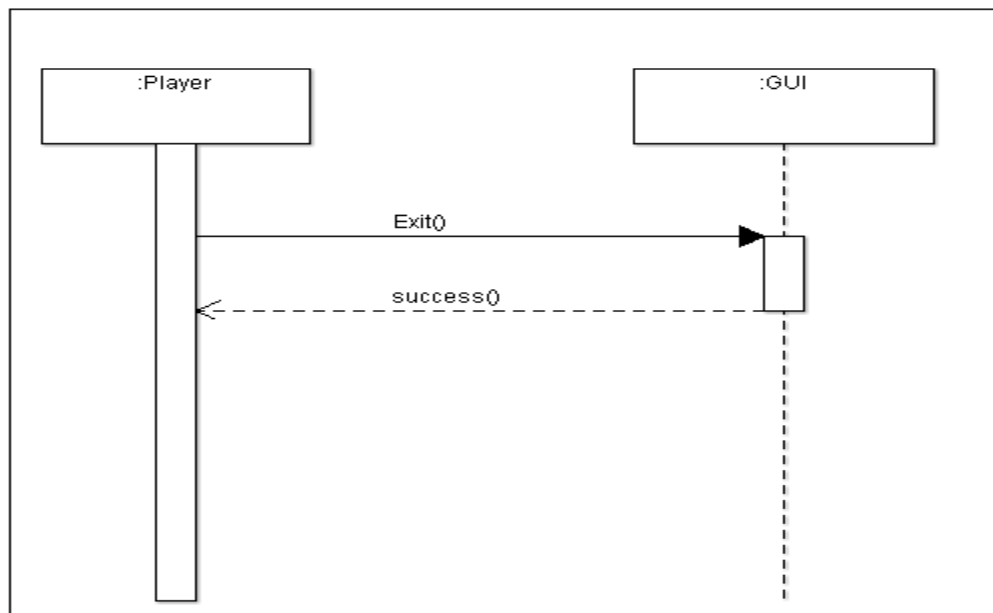


Figure 3-18: Sequence diagram of exit game



3.9 Class Diagram

Class diagram shows the classes of the system, their interrelationships including inheritance, association and aggregation, operations and attributes of the classes.

Here is class diagram of vehicle Racing Game.

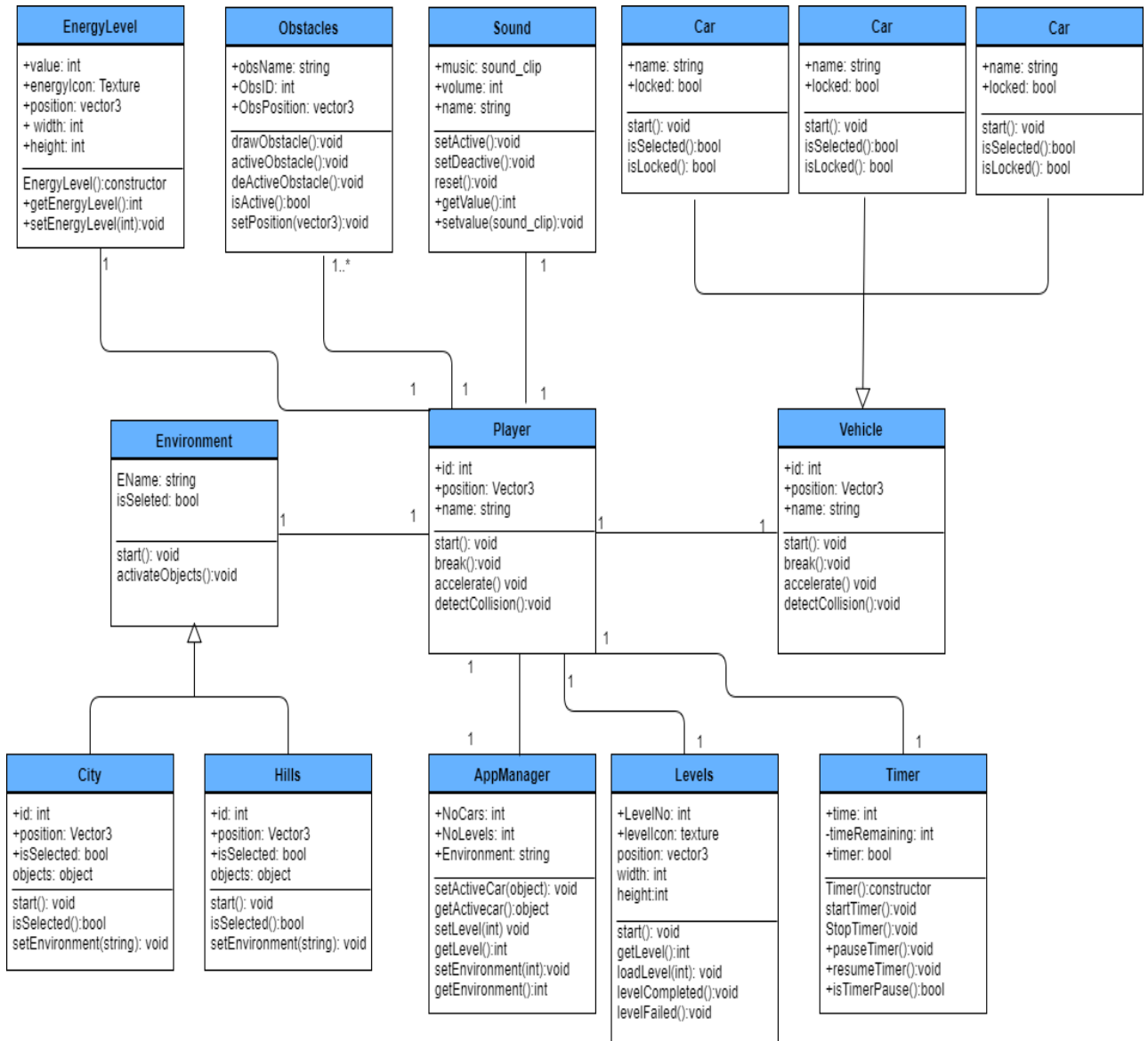


Figure 3-19: Class diagram



3.10 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

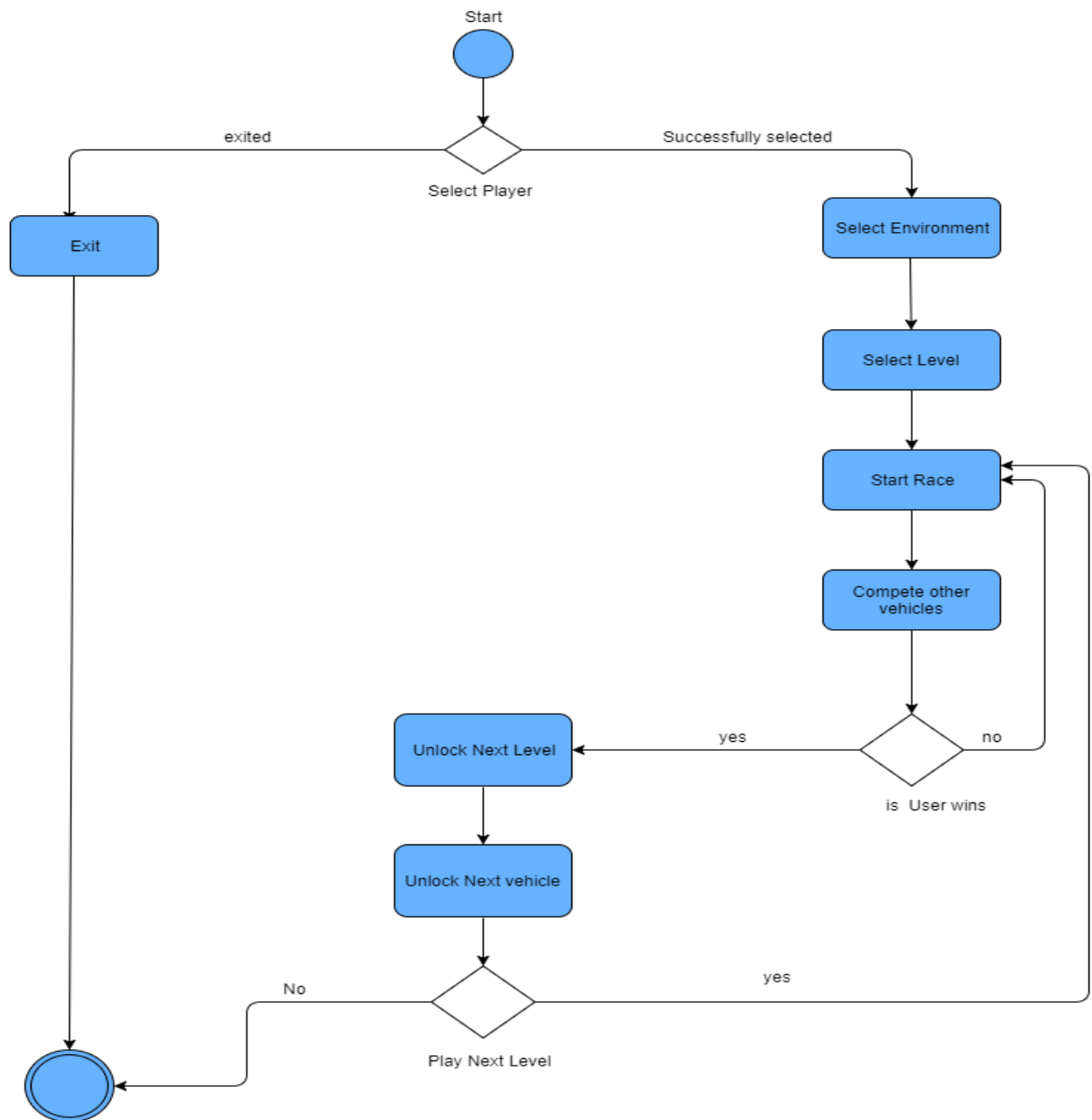


Figure 3-20: Activity Diagram



Chapter 4

Implementation Details



4.1 Introduction

Implementation is realization of technical specification or algorithm as a program, software component or other computer system through computer programming and development. In this chapter we provide you framework and language details of product Neels Wheels 3D. Screen shots are also given below.

4.2 Framework selection

The game is deployed in unity3D v5.6. It supports Android Software Development kit (SDK). This development tool is preferred because it is more flexible than others. It is portable and cross platform. Unity provides us physics and rigid body to handle vehicles. Unity is very flexible in terms of creating GUI's and 3D models.

4.3 Operating System selection

As this is an Android racing game so Operating system preferred is Android. In future it can be for IOS and windows with little changes in code.

4.4 Language

Language used for Neels Wheels Android 3D racing game is C#. While the game physics is done in JavaScript. As unity mono develop supports both C# and JavaScript. Mono develop scripting is easier than JavaScript.

4.5 Screen Shots

Here are screen shots of final product.



Figure 4-1: Main Menu



Figure 4-2: Select Vehicle



Figure 4-3: Select environment



Figure 4-4: Select Level



Figure 4-5: City Environment



Figure 4-6: Hills Environment



Figure 4-7: Snow Environment



Chapter 5

SOFTWARE TEST DOCUMENTATION (STD)



5.1 INTRODUCTION

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

5.2 System Overview

This is a 3D vehicle racing game. In this project, the user can have the option of selecting a vehicle for racing, user can select from among a car, a motorbike, a truck or a bus. It's totally user's choice. (S)He can have a choice of selecting different configurations of a vehicle before starting the race. It will have different difficulty stages and levels. Player can buy new vehicles after completing level. Each race will have different goals and time to complete a race. Vehicles will be fully functional as automated car games, including real physics of car. Racing environment is three dimensional and gives realistic stimulation.

5.3 Test Approach

User acceptance testing (UAT) also called beta testing or end user testing, consist of a process of verifying that a solution works for the user [6]. It is not system testing (ensuring software does no crash and meets documented requirements), but rather is there to ensure that the solution will work for the user i.e. test the user acceptance the solution (software vendors often refer to as Beta testing).

This should be undertaken by a subject-matter expert (SME), preferably the owner or client of the solution under test, and provides a summary of the findings for confirmation to proceed after trail or overview. In software development, UAT as one of the final stages of a project often occurs before a client or customer accepts the new system. Users of the system perform test in line with what occur in real life scenarios [2].

5.4 Testing Objectives

For checking whether the requirement in SRS are fulfilled or not we have to make test on these cases.

UAT has following objectives.

User Acceptance test make test case against the requirements.

User Acceptance test check actual function, input, expected result, actual result, procedure to make test case, pass/fail status against each test case.

5.5 TEST PLAN

5.5.1 Features to be Tested

Features to be tested are all according to client/user/player prospective. For example

- Start game
- Select racing vehicle
- Select racing environment
- Set difficulty level
- Select play



- Pause game
- Resume game
- Move vehicle
- Avoid obstacles
- Lose game by crashing
- Lose game by time over
- Restart game
- Achieve milestone
- Unlock next level
- Exit game

5.5.2 Features not to be Tested

Features not to be tested are from the developer's point of view. For example

- Frame rate of game
- How much power is used by processor
- How much memory is consumed by the game
- Software risk factor
- Maintainability of game

5.5.3 Testing Tools and Environment

As this is beta testing (testing by the user) so no specific tools and environment is required. All a user need is an Android device with game installed.

5.6 TEST CASES

Following are the test cases of Neels Wheels.

Test Case 1:

Purpose	Select racing vehicle
Setup	<ol style="list-style-type: none">1. Open game2. Go to main menu3. Select play game
Instructions	<ol style="list-style-type: none">1. Select racing vehicle option2. Choose a racing vehicle for race
Expected Result	Desired racing vehicle has been selected
Actual Result	Desired racing vehicle has been selected
Verdict	Pass



Test Case 2:

Purpose	Select racing environment
Setup	<ol style="list-style-type: none">1. Open game2. Go to main menu3. Select play game4. Select racing vehicle
Instructions	<ol style="list-style-type: none">1. Select racing environment option2. Choose a racing environment/track for race
Expected Result	Race environment has been selected successfully.
Actual Result	Race environment has been selected successfully.
Verdict	Pass

Test Case 3:

Purpose	Set difficulty level
Setup	<ol style="list-style-type: none">1. Open game2. Go to main menu3. Select play game4. Select racing vehicle5. Select racing environment
Instructions	<ol style="list-style-type: none">1. Select difficulty level option2. Choose a difficulty level for race
Expected Result	Difficulty level has been selected successfully.
Actual Result	Difficulty level has been selected successfully.
Verdict	Pass

Test Case 4:

Purpose	Select Play
Setup	<ol style="list-style-type: none">1. Open game2. Go to main menu



Instructions	1. Select play game option
Expected Result	Game and timer started with selected configuration of vehicle and environment.
Actual Result	Game and timer started with selected configuration of vehicle and environment.
Verdict	Pass

Test Case 5:

Purpose	Pause Game
Setup	<ol style="list-style-type: none">1. Open game2. Go to main menu3. Select configuration of vehicle and environment.4. Start game by selecting play game.
Instructions	2. Select pause game option
Expected Result	Timer has stopped and game is paused.
Actual result	Timer has stopped and game is paused.
Verdict	Pass

Test Case 6:

Purpose	Resume Game
Setup	<ol style="list-style-type: none">1. Open game.2. Go to main menu.3. Select configuration of vehicle and environment.4. Start game by selecting play game.5. Pause game.
Instructions	1. Select resume game option.
Expected Result	Timer has started and game has resumed.
Actual Result	Timer has started and game has resumed.
Verdict	Pass



Test Case 7:

Purpose	Move vehicle
Setup	<ol style="list-style-type: none">1. Open game.2. Go to main menu.3. Select configuration of vehicle and environment.4. Start game by selecting play game.5. Move vehicle
Instructions	<ol style="list-style-type: none">1. Select left, right arrow keys to move vehicle on the path
Expected Result	Vehicle has moved.
Actual Result	Vehicle has moved.
Verdict	Pass

Test Case 8:

Purpose	Avoid obstacles
Setup	<ol style="list-style-type: none">1. Open game.2. Go to main menu.3. Select configuration of vehicle and environment.4. Start game by selecting play game.5. Avoid obstacles
Instructions	<ol style="list-style-type: none">1. Use left, right and brake GUI to avoid obstacles.
Expected Result	Vehicle avoided obstacle.
Actual result	Vehicle avoided obstacle.
Verdict	Pass

Test Case 9:

Purpose	Finish game by crashing
Setup	<ol style="list-style-type: none">1. Open game.2. Go to main menu.3. Select configuration of vehicle and environment.4. Start game by selecting play game.



	5. Obstacle comes
Instructions	1. Hit obstacle
Expected Result	Crashed sound, level failed.
Actual Result	Crashed sound, level failed.
Verdict	Pass

Test Case 10:

Purpose	Finish game by time over
Setup	<ol style="list-style-type: none">1. Open game.2. Go to main menu.3. Select configuration of vehicle and environment.4. Start game by selecting play game.5. Avoid obstacles
Instructions	1. Move vehicle slowly consume all time.
Expected Result	Time over, level failed.
Actual Result	Time over, level failed.
Verdict	Pass

Test Case 11:

Purpose	Restart game
Setup	<ol style="list-style-type: none">1. Open game.2. Go to main menu.3. Select configuration of vehicle and environment.4. Start game by selecting play game.5. Fail level
Instructions	1. Click restart game.
Expected Result	Game restarted.
Actual Result	Game restarted.
Verdict	Pass



Test Case 12:

Purpose	Achieve milestone
Setup	<ol style="list-style-type: none">1. Open game.2. Go to main menu.3. Select configuration of vehicle and environment.4. Start game by selecting play game.5. Avoid obstacles
Instructions	<ol style="list-style-type: none">1. Use left, right and brake GUI to avoid obstacles and reach to the milestone.
Expected Result	Milestone achieved. Next level has unlocked.
Actual Result	Milestone achieved. Next level has unlocked.
Verdict	Pass

Test Case 13:

Purpose	Unlock next level
Setup	<ol style="list-style-type: none">1. Open game.2. Go to main menu.3. Select configuration of vehicle and environment.4. Start game by selecting play game.5. Avoid obstacles and win game by achieving milestone6. complete goal
Instructions	<ol style="list-style-type: none">1. Go to main menu2. Go to levels
Expected Result	New level unlocked.
Actual Result	New level unlocked.
Verdict	Pass



Test Case 14:

Purpose	Buy new vehicle
Setup	<ol style="list-style-type: none">1. Open game.2. Go to main menu.3. Select configuration of vehicle and environment.4. Start game by selecting play game.5. Avoid obstacles and win game by achieving milestone6. Complete level
Instructions	<ol style="list-style-type: none">1. Go to main menu2. Click buy new vehicle
Expected Result	New vehicle unlocked.
Actual Result	New vehicle unlocked.
verdict	Pass.

Test Case 15:

Purpose	Exit game
Setup	Not defined.
Instructions	Select Exit game option.
Expected Result	Confirmation message: Exit game is displayed.
Actual Result	Confirmation message: Exit game is displayed.
Verdict	Pass



Chapter 6

Conclusion and Future Work



6.1 Conclusions and Summary

Neels Wheels 3D is an Android based vehicle racing game. The main idea which makes this game unique is the feature of selecting different vehicles from different category, like player can select car, motor bike, bus or truck. The application is efficient and have quick responses to the actions. The color scheme and graphics are kept captivating. It provides 3D environment to player to move their vehicle and race with other cars, trucks or buses. User can select racing environment of his/her own interest. It provides fun and entertainment to the users to race their desired vehicle. The purpose of this game is to learn.

One of the big difference with all other racing games available on play store is the size of game, I did my best efforts to keep the game size as small as I can so user can enjoy without having any difficulty or problems of memory management. One of my major goal was to provide maximum functionality at the very minimum cost of memory. This game is offline and freely available on Google Play store worldwide.

Android game development is the field of my personal interest so I chose this game as my final year project. I faced many difficulties and challenges but with determinism I passed through all of that and finally the game is in your hands.

6.2 Future Enhancements

- Number of players can be increased.
- Number of levels can be increased.
- Share score on Facebook.
- Multiplayer.



Appendix



7 ADDITIONAL MATERIAL

No additional material is required.

7.1 References

Following is the list of documents and Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.

1.	C. Larman, APPLYING UML AND PATTERNS An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd ed., Massachusetts: Pearson Education, 2005
2.	Roger S. Pressman, Software Engineering - A Practitioner's Approach, McGraw Hill, 7th Edition, 2010
3.	<ul style="list-style-type: none">Google Nexus Press Event¹ http://www.androidcentral.com/google-says-there-are-now-14-billion-active-android-devices-worldwide
4.	<ul style="list-style-type: none">Play store statistics² https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/