

SUPERIOR UNIVERSITY LAHORE



Faculty of Computer Science & IT

Final Year Project PROJECT REPORT

[CAR RACER]

Project ID: [BSSE-S17-003]

Project Team

Student Name	Student ID	Program	Contact Number	Email Address
Yasir Nazir	15138	BSSE	+92-302-4514082	yasir14251@gmail.com
Omair Younas	15139	BSSE	+92-336-0001177	omiirana@gmail.com
Ameer Hamza	15118	BSSE	+92-307-6333961	ameerhamzasuddal@gmail.com
Asad Ullah Sarwar	15113	BSSE	+92-334-5417777	asad.sarwar76@gmail.com

[Project Supervisor]
(Mumtaz Ali)

Project Report

[CAR RACER]

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

Name: _____

Date: _____ Signature: _____

PROJECT MANAGER

Comments: _____

Date: _____ Signature: _____

HEAD OF THE DEPARTMENT

Comments: _____

Date: _____ Signature: _____

Dedication

We dedicate our dissertation work to our families and to God Almighty our creator, the strong pillar, source of inspiration, wisdom, knowledge and understanding. He has been the source of our strength throughout this program. A special feeling of gratitude to our loving parents, their words of encouragement and push for tenacity always ringed in our ears.

We would also like to dedicate this dissertation to our many friends and teachers of Superior who have supported us throughout the program. We will always appreciate all that they have done for us.

Acknowledgements

Alhamdulillah, all praises to Allah Almighty, the most Merciful and the most Gracious, for the strengths and his blessings in completing this Project. Working on this project was indeed a challenging task that demands immense efforts. But we feel proud to express our deepest sense of gratitude and appreciation to our supervisor Mr. Mumtaz Ali for his kind help, advice, inspired guidance, unlimited support, sympathetic attitude and sincere personal involvement throughout the development of project.

We would never have been able to reach this stage but for the prayers and great support of our families. Thanks and best wishes for all those who have made this learning experience so wonderful for us.

Executive Summary

The project “CAR RACER” is developed for a startup of business which is presented in final year project. The idea is to develop a car racing game for people to play. User can download this game from play store. It is for android based smart phones. In Car Racer, player can choose his/her car. By gaining points new cars gets unlocked.

- Download Car Racer game.
- Select any Car
- Chose different Map
- Change Car Settings
- Change Environment

Table of Contents

Contents

Dedication.....	iv
Acknowledgements.....	v
Executive Summary.....	vi
Table of Contents.....	vii
List of Figures.....	ix
List of Tables.....	x
Chapter 1.....	1
Introduction.....	1
1.1. Background.....	2
1.2. Motivations and Challenges.....	2
1.3. Goals and Objectives.....	3
1.4. Gap Analysis.....	3
1.5. Proposed Solution.....	4
1.6. Project Plan.....	4
1.6.1. Work Breakdown Structure.....	4
1.6.2. Roles & Responsibility Matrix.....	5
1.6.3. Gantt Chart.....	7
1.7. Report Outline.....	8
Chapter 2.....	9
Software Requirement Specifications.....	9
2.1. Introduction.....	10
2.1.1. Purpose.....	10
2.1.2. Document Conventions.....	11
2.1.3. Intended Audience and Reading Suggestions.....	11
2.1.4. Product Scope.....	12
2.1.5. References.....	12
2.2. Overall Description.....	12
2.2.1. Product Perspective.....	12
2.2.2. Product Functions.....	13
2.2.3. User Classes and Characteristics.....	13
2.2.4. Operating Environment.....	14
2.2.5. Design and Implementation Constraints.....	14
2.2.6. User Documentation.....	15
2.2.7. Assumptions and Dependencies.....	16
2.3. External Interface Requirements.....	16
2.3.1. User Interfaces.....	16
2.3.2. Hardware Interfaces.....	16
2.3.3. Software Interfaces.....	16
2.3.4. Communications Interfaces.....	17
2.4. System Features.....	17
2.4.1. System Feature 1.....	17

2.4.2.	System Feature 1(for the users).....	17
2.4.2.1.	Description and Priority.....	17
2.4.2.2.	Stimulus/Response Sequences.....	17
2.4.2.3.	Functional Requirements	17
2.4.3.	System Feature (for admin)	18
2.4.3.1.	Description and Priority.....	18
2.4.3.2.	Stimulus/Response Sequences.....	18
2.4.3.3.	Functional Requirements	18
2.5.	Other Nonfunctional Requirements	18
2.5.1.	Performance Requirements.....	18
2.5.2.	Safety Requirements.....	18
2.5.3.	Security Requirements.....	18
2.5.4.	Software Quality Attributes.....	19
2.5.5.	Business Rules	19
2.6.	Other Requirements.....	19
Chapter 3	20
Use Case Analysis	20
3.1.	Use Case Model	22
3.2.	Fully Dressed Use Case.....	23
Chapter 4	24
System Design	24
4.1.	Architecture Diagram.....	26
4.2.	Sequence/Collaboration Diagram	27
	Operation contracts	27
4.3.	Activity Diagram.....	28
4.4.	Data Flow diagram.....	29
Chapter 5	30
Implementation	30
5.1.	Important Flow Control/Pseudo codes.....	31
5.2.	Components, Libraries, Web Services and stubs	32
5.3.	Deployment Environment	33
5.4.	Tools and Techniques.....	33
5.5.	Best Practices / Coding Standards.....	33
Appendix A: Information / Promotional Material	35
Chapter 6	36
Testing and Evaluation	36
Chapter 7	41
Summary, Conclusion and Future Enhancements	41
7.1.	Project Summary	42
7.2.	Achievements and Improvements	42
7.3.	Future Enhancements/Recommendations	42
Reference	43

List of Figures

1.4	Caption of first figure of first chapter	03
1.6.1	Caption of second figure of first chapter	05
2.2.5	Caption of first figure of second chapter	15
3.1	Caption of first figure of third chapter	22
3.2	Caption of second figure of third chapter	23
4.0	Caption of first figure of forth chapter	25
4.0	Caption of second figure (Part A+B) of forth chapter	26
4.1	Caption of third figure of forth chapter	26
4.2	Caption of forth figure of chapter forth	27
4.3	Caption of fifth figure of chapter forth	28
4.4	Caption of Sixth figure of chapter forth	29
5.1	Caption of first figure (Path A+B) of chapter fifth	31
5.5	Caption of Second figure (Part A+B) of chapter fifth	33
8.0	Caption of first figure (Part A+B) Appendix	35

List of Tables

1.6.2.	Label of first table of first chapter	06
6.1	label of first table of six chapter	37
6.1	label of second table of six chapter	38
6.1	label of third table of six chapter	39
6.1	label of forth table of six chapter	40

Chapter 1

Introduction

Chapter 1: Introduction

Car racer is a 3rd person racing game. It is for android based smart phones. In Car Racer, player can choose one car from the given cars. There are 3 tracks. Player can choose any one of them. By gaining points new cars gets unlocked. The requirements for the game to contain 3D graphics introduced an interesting challenge for the project. Since all had none or little experience in 3D modeling. Spending time to learning that how to model proper 3D models for our game was therefore necessary. During the research to find out what 3D models program to use, we found that we could use different studios to create models that we could later use in our project. The complete game contains, models made in both Blender and 3D studio MAX.

1.1. Background

Car Racer is a 3rd person racing game. In this game, there are certain points where player has to be extra careful. If not they will be thrown a little back. Game will be built using Unity 3D engine which is not a specific platform. But this game is specifically for android user's. We are going to use different 3D libraries. And take full advantage of Unity IDE.

1.2. Motivations and Challenges

- Design a small advanced 3D grade, A worthy game in unity 3D
- Cover topics such as graphics designs ,`3d `graphics ,art animations ,designs
- Support each other as a group members
- Management problems
- Production problem
- Analyze the problems
- Problems solving technique

1.3. Goals and Objectives

In developing this project, our main goal is provide the facilities to user's to 3 racing tracks for racing. Player can choose any one of them. By gaining points new cars gets unlocked. To design android games android studio must available.

- Make a Unity 3D game for android and desktop users
- Marking a game very easy to play without any hesitation
- Efficient gaming system to play
- No payments for playing games

1.4. Gap Analysis

The performance of our game is according to market demands and in future we add some more features in our game and make performance better and better quality assurance make our project important and increase its performance better.

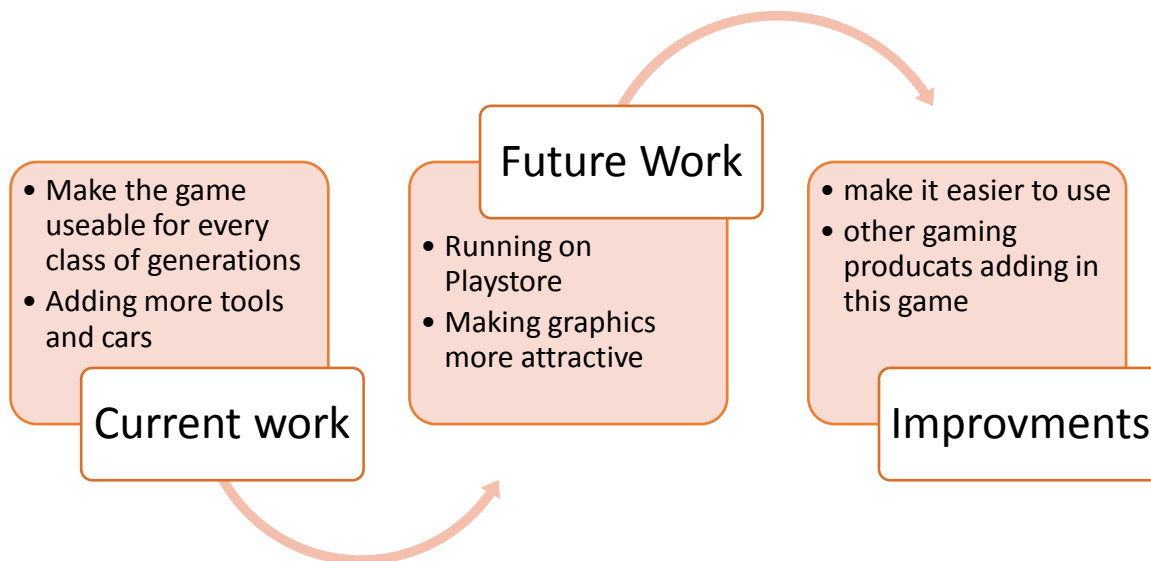


Figure:01 Gap Analysis, Current work to future work.

1.5. Proposed Solution

Developing software applications is a time consuming process, and with time consuming process cost is getting high. During the last years, several software methodologies, often know as agile software development methodologies can be more or less good, depending on the task and application type. The game must contain 3D models, and render these in game 3D environments were never a requirements, and platform, and platform games with 2D envirmant could still open for 3D objects.

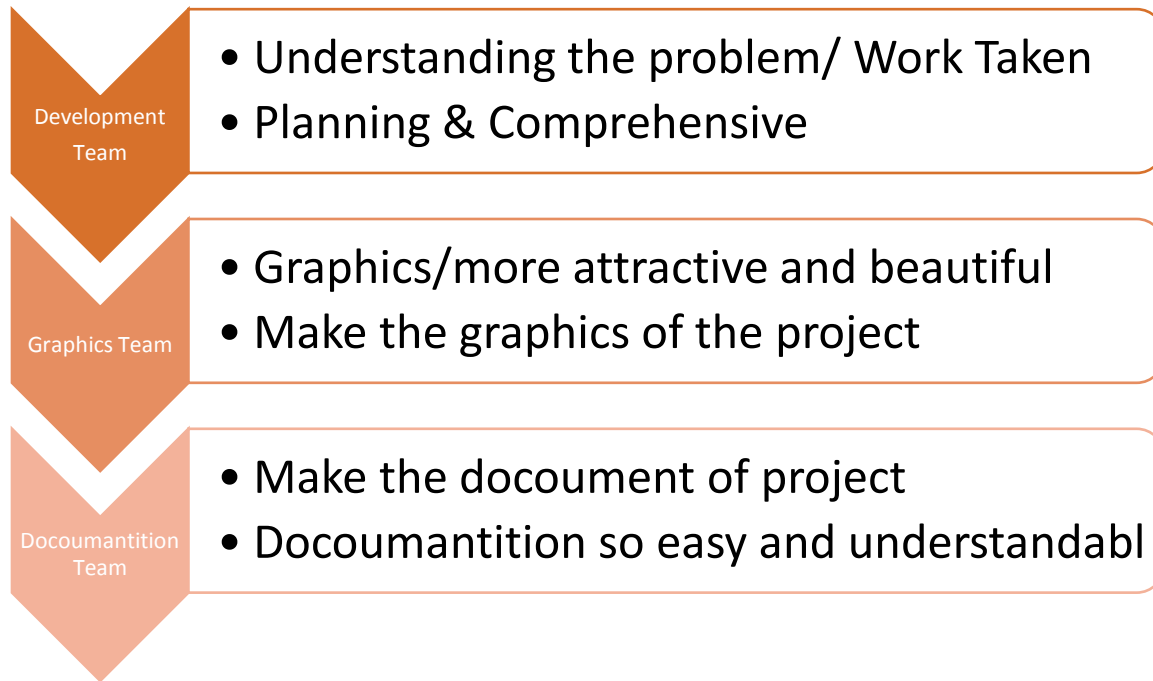
1.6. Project Plan

The important aim of this game is to facilitate the people with better entertainment. In this project, we use different type of techniques by using 3D unity development.

- Development Team
 - Understanding the problem/ work taken
 - Planning & Comprehensive
- Graphics Team
 - Make the graphics of the project
 - Graphics/more attractive and beautiful
 -
- Documentation Team
 - Make the document of project
 - Documentation so easy and understandable

1.6.1. Work Breakdown Structure

All the important figures of game plane are described such as, Blender, 3D Unity and other Android studio figures. Unity support both 2D and 3D unity platform.

**Figure:02**

(Structure of Development team, Graphic team & Documentation team)

1.6.2. Roles & Responsibility Matrix

We all the team members are working as a team, that's why we divided the whole team in different parts and assign the works to them.

- **FYP Coordinator Responsibilities**

1. Combine students name lists.
2. We arranged a class on daily basis and have a talk with our project advisor.
3. To maintained the project plane and process.
4. Compile the data base of students, a supervisor, and project plan.

- **FYP team Responsibilities**

- **Team leader Responsibilities**

The Project Manager has primary responsibility for defining, planning, tracking, and managing the project they are assign. He is responsible

identifying key resources and providing the direction required for meeting the project objectives. He is also responsible for ensuring appropriate management, customer and supplier involvement throughout the life of project. Qualifications and skills of the project manager include an ability to lead and motivated people and encourage team work, an ability to communicate effectively with senior management and a clear vision of what determine successful product for customers and for the component.

➤ **Yasir Nazir**

ASP.net MVC

Data base

Jquery

3D unity

Blender

➤ **Ameer Hamza**

ASP.net

OOP

3D UNITY

Blender

➤ **Umair younas**

3D unity

MVC

Blender

Andriod Studio

➤ **Asad -ullah Sarwar**

3D unity

Blender

Andriod Studio

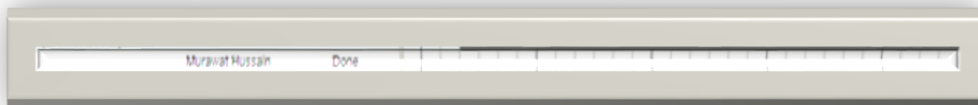
Task Name	Start Time
System analysis	08/11/2017
System planning	17/11/2017
Documentation	03/01/2018
System designing	02/02/2018
Implementation	23/02/2018
Testing	07/03/2018

Table:01
(Roles and Responsibilities of the whole team.)

1.6.3. Gantt Chart

Phase 1: Requirement Analysis

This phase will consists of 30 Days (November 01 to 30)



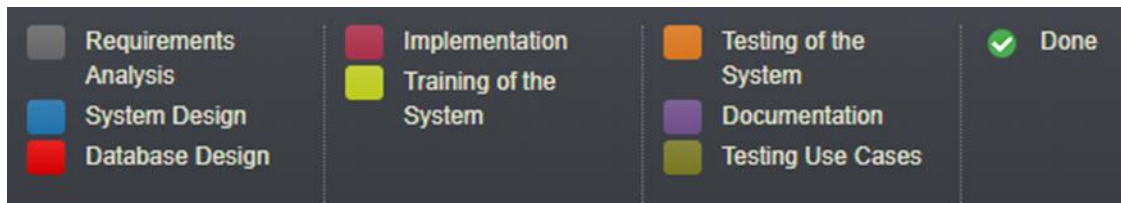
Phase 2: System Design

This phase will consists of 30 Days (December 01 to 30)



Phase 3: Game Design

This phase will consists of 15 Days (Feb 01 to 15)



1.7. Report Outline

In this report, we have written complete details of our project, that how our project works, what type of features we have added in our project, how the user played the game, what coding languages we have used, how we designed the graphics. In short we have listed all the details of our project.

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

- **Requirements definitions**

All the requirements on the system are found by talking to system users.

Examples of requirements are services, constraints and goals.

- **System and Software designs**

In this activity over all architecture of system, is established.

- **Implementation and Unit testing**

The software implemented unit which is already tested. Further testing is required.

- **Integration and system testing**

The units are merged to make a complete system. Further testing is required to merge the system.

- **Operation and maintained**

2.1.1. Purpose

Developing software applications is a time consuming process and with time consuming process comes high cost. During the last years, several software methodologies, often known as Agile Development, have become widely used by software developers to address this issue. Many different development methodologies can be more or less good, depending of the task and applications. One of software development methodologies(UML) Unified Modeling Language, which as the name hints, takes on an evolutionary approach to the problem, and allows the project to evolve through different stages of project. When we chose to develop a 3D graphics computer games. Some requirements for the computer game were given from the beginning.

- **3D Graphics**

The game must contain 3D models, and render these in same 3D environments were never requirements, and platform games with 2D. Environments could still open for 3D objects.

- **Impressive Results**

The game result must impress whoever plays the game. It should last long, and make the player come back and play it over and over again.

- **Graphical Effects**

To achieve an impressive result, we would need to add modern graphical effects, such as real-time rendered soft-shadows, motion, and ambient, occasion.

2.1.2. Document Conventions

Our document consist of following important conventions

- UML (unified modeling language)
- Data base
- ERD (Entity relation diagram)
- AD(Architure diagram)
- Sequence Diagramed
- Activity diagram
- STD

2.1.3. Intended Audience and Reading Suggestions

Our project has been done under the guidance of our professors of SUPERIOR UNIVERSTY LAHORE and at the moment it has been restricted to only educational institutes as it is very helpful in term of making a android game more and more much faster as compare to manual process and also helps improve the quality of environment for the android along with unity as well. This project can make change in terms of latest making a game android and unity as well as compared to technologies to making the project for the android as well as unity users.

2.1.4. Product Scope

- We made development of the game more easier because of implementations tools like unity 3D games.
- We support effective development of the game system of racer with its high level abstractions programming tools.
- We use 3D models, an render these in the game.
- We also used USK model in the game
- We target the main market
- The scope of game is wide and loud
- We made it user friendly and good 3D features
- We polish our game according to programming skills
- In future we should play it on GOOGLE PLAYSTORE

2.1.5. References

- <http://unity3d.com/unity/features/iphone-publishing>
- http://www.nasa.gov/mission_pages/station/main/index.html
- <http://3dbuzz.com/forum.html>
- <http://unity3d.com/support/documentation/Manual/index.html>

2.2. Overall Description

2.2.1. Product Perspective

For our user we defend the major prospects of our project that are very helpful according to users prospects.

- 3rd Generation game
- For Android phone users
- Users have 5 different ways
- 3 tracks for racing
- Player can choose according to its wish

- Android users can have options to use coins
- Specific Game for Android user's

2.2.2. Product Functions

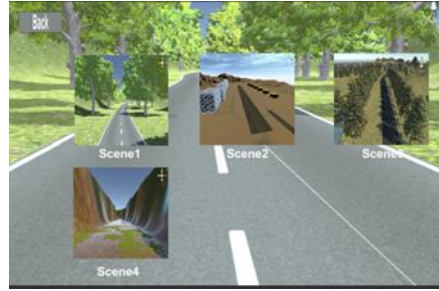
- Start Button
- User can start the game ,and all handles gave to users to start this game
- Pause Button
- The User can easily pause the cars on differ ways according to his own wish
- Turn/Move button
- User can easily turn or move the car
- Bit Coins offer
- After achieving the goals or requirements some achievement or goals are given to user then he found them

2.2.3. User Classes and Characteristics

We have made a simple Main Menu with very few buttons. First screen that pop-ups when we start the game is main menu scene. It has two buttons one is start game and other is exit game. Start game buttons takes us to the next scene in which we have to options to play. Both options are connected with scenes.



We added the following features in our game for the user's requirements.



2.2.4. Operating Environment

- Android Mobile phones
- Windows servers /8/10
- Operating systems

2.2.5. Design and Implementation Constraints

This chapter includes the design and architecture of proposed system

- SDLC Model

This incremental model is used for the development of project.

- The Agile Development Model

Agile software development is a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.



Figure:03
(Agile Development)

Agile methods break tasks into small increments with minimal planning, and do not directly involve long-term planning. Iterations are short time frames (time boxes) that typically last from one to four weeks. Each iteration involves a team working through a full software development cycle including planning, requirements analysis, design, coding, unit testing, and acceptance testing when a working product is demonstrated to stakeholders. This minimizes overall risk and allows the project to adapt to changes quickly. Stakeholders produce documentation as required. An iteration may not add enough functionality to warrant a market release, but the goal is to have an available release (with minimal bugs) at the end of each iteration. Multiple iterations may be required to release a product or new features.

2.2.6. User Documentation

User make a very important role to make a our game more attractive and give us kind suggestions .So for user help I tell that about our gaming functions .In our game a user can choose his game in 10 different ways. There are more 3 tracks or racing .Player can choose any one of them. By gaining points the new cars unlocked for users play. So the points can be added In this game.

2.2.7. Assumptions and Dependencies

Assumptions and dependencies are the factors on which the project depends on or the scenarios that can affect our project. For example, if the Android user cannot select the proper path and car then what will be second option for users. If the hardware machine is not according to requirements to run the project, then it will be a very high possibility of errors and slow process this gaming. Or if the user has not applied the security principles on the project according to his requirements then it will act randomly and the issues of security will start to raises.

2.3. External Interface Requirements

2.3.1. User Interfaces

The general user interface is the most important part of any software. Android studio makes the use of the software much more efficient, easy to understand and use if it is clear and well Designed. Our project is almost android based so we have designed graphics of our software according to the general desktop application standards.

2.3.2. Hardware Interfaces

Car Racer is a mobile gaming application design for android users but we keep a function which a game can play on mobile phones and tablets. Gaming application data is stored on its gaming engine. This game is developed for initially for only android users and we little changing some new ideas in the game and become more attractive for our others users .In future we add more graphics in this game. We want to add some hardware ordinations like, scaling, pixel formatting, and accelerated 3D graphics.

2.3.3. Software Interfaces

The general users interface is very important phase of any project. Our software consist of different parts of project. The screen short of all major parts are attached in the system.

2.3.4. Communications Interfaces

It is used to make a commutations between users and admin panel of the game.

2.4. System Features

This system has designed for game development in a secure way and much faster way. It has easiest way to play without any difficult stages or phases.

2.4.1. System Feature 1

This system has designed for game development in a secure way and much faster way. It has easiest way to play without any difficult stages or phases.

2.4.2. System Feature 1(for the users)

2.4.2.1. Description and Priority

We made this game, for all types of users, like android so main priority of this game is to development to make new techniques and graphics.

2.4.2.2. Stimulus/Response Sequences

The responds of our game is very much fast in term of sequences, means that this game. Is quick response for the player? Users have not face any difficulties by using it.

2.4.2.3. Functional Requirements

- The game developed in Unity3d for Android users as well as tablet users
- Display, Control and Audio
- The game must be played on Mobile and computer.

2.4.3. System Feature (for admin)

2.4.3.1. Description and Priority

Admin has taken access to the player to menu button and other features of our project.

2.4.3.2. Stimulus/Response Sequences

The features include in the project have very quick response, from the menu button.

2.4.3.3. Functional Requirements

- Quick response from the menu button
- Choose the path according to users wish

2.5. Other Nonfunctional Requirements

2.5.1. Performance Requirements

The system should perform in accuracy and precision to avoid errors and problems. It should be fast so it doesn't take time more than expectation because the main purpose of using this game to improve the quality of Android Studio users and in future more attributes are included in this game as compare to other games.

2.5.2. Safety Requirements

The game must have safety features like the game should be easy to modify and it will be easy for the user to correct any wrong on the time to avoid errors as much as possible.

2.5.3. Security Requirements

The game should be secure as it should save the users privacy and doesn't leak out any important information about the our game features it will be a big security flaw that can't be filled if the game is not secured and in that way it will be a failure.

2.5.4. Software Quality Attributes

The game should be flexible, means it should be able to expand as much as possible according to the requirements of android users and must have the capacity to modify in a way that the performance of the systems doesn't affected by the process of modification. The game must be easy to use and user friendly and also easy to the maintenance of the system should be quite easy as it plays a very much important role of success in product success.

2.5.5. Business Rules

According to business rule we working on this game and in future we uploaded it on play store for the business purpose.

2.6. Other Requirements

We have covered almost all the necessary requirements for the SRS and development phase. There are no more any other requirements which are worth mentioning separately.

Chapter 3

Use Case Analysis

Chapter 3: System Analysis

The main objective of this project is to learn about gaming technologies and Gaming engines. Unity has provided us with the basics of how to make a game. This game is a first step towards better understanding of games.

- **Functional Requirements**
 - The game developed in Unity3d
 - Display, Control and Audio
 - The game must be played on Mobile and computer
 - Game must be played on Android Operating system
- **None Functional**
 - Audio control
 - Performance
 - Development User interface should be clean and compatible with mobile screens
 - Tools

3.1. Use Case Model

It shows the static view of a software. The class diagram is an only diagram that show the directly mapped with object oriented languages. It also show the collection of classes, interfaces, and applications constrains.

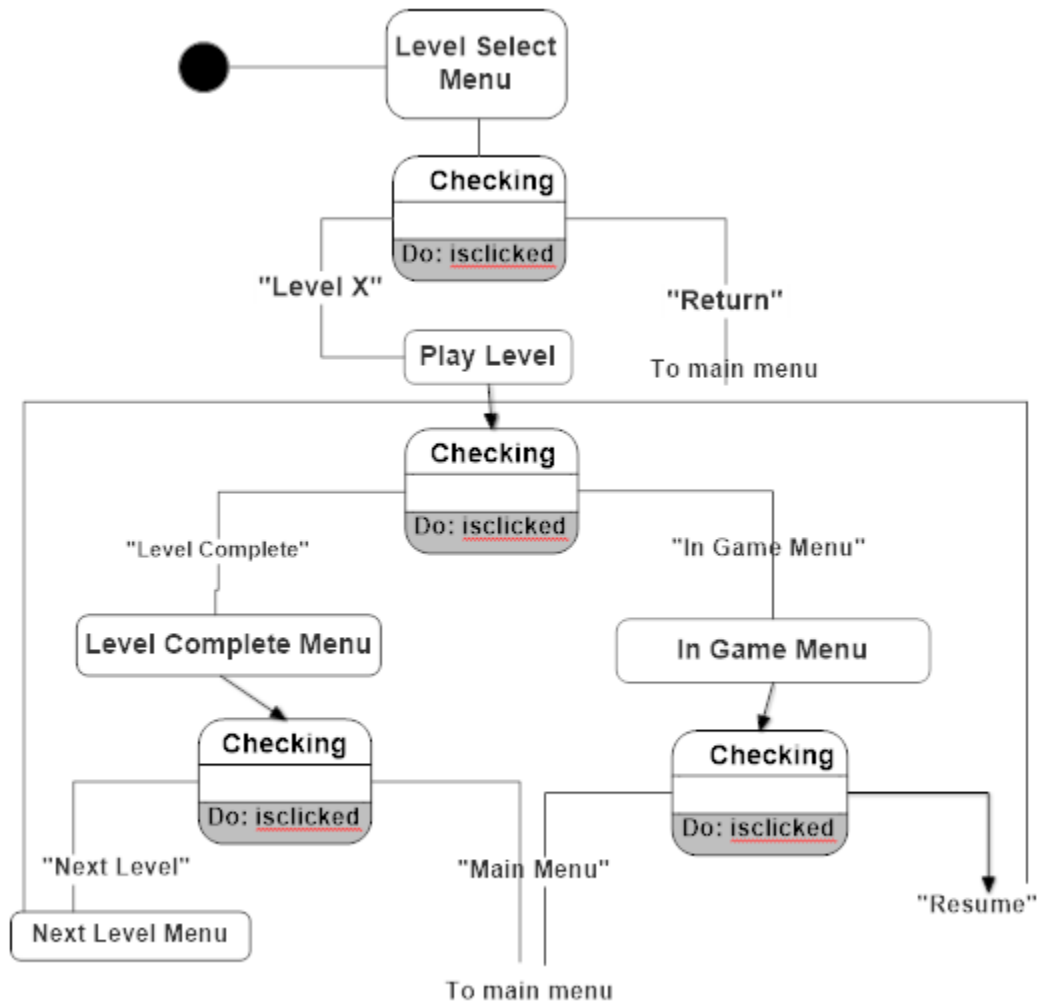


Figure:04

(Use Case Model of what happened when user enters in the game.)

3.2. Fully Dressed Use Case

This use is a graphical representation of an information that show the relationship between the exit game,, scene 1, scene2, scene3, start game within the system. The use case diagram is a technique that help to define any business process.

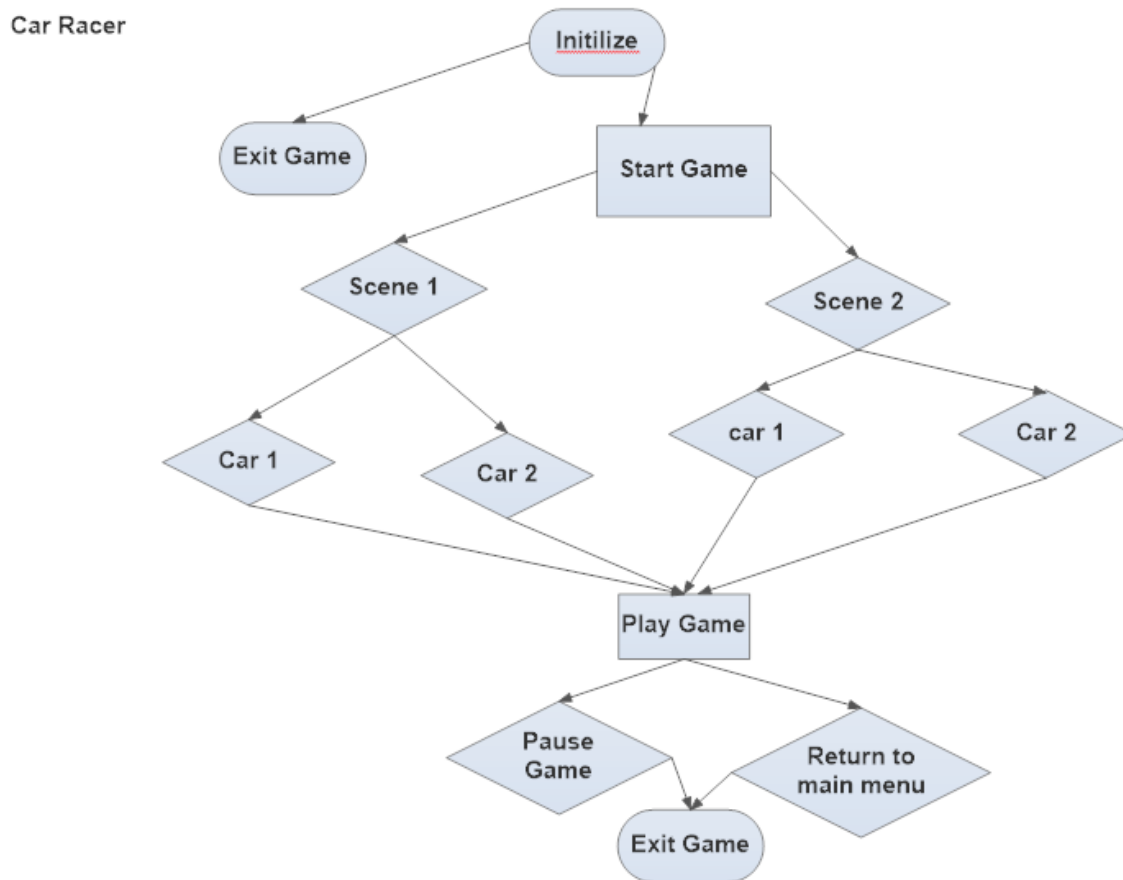


Figure : 05

(Fully Dressed Use Case, Complete details of Initialize state to Exit Game.)

Chapter 4

System Design

Chapter 4: System Design

- Scenes

Scene 1 is a mix of mountains, forests and Buildings. Most of the terrain is covered with grass and sand texture. The scene looks something like this.

- Terrain

The terrain includes trees, mountains, lake and buildings. The terrain width and length is set to 2000. Detail Resolution is 1024. Wind speed for grass is 0.5. All other stuff is default.

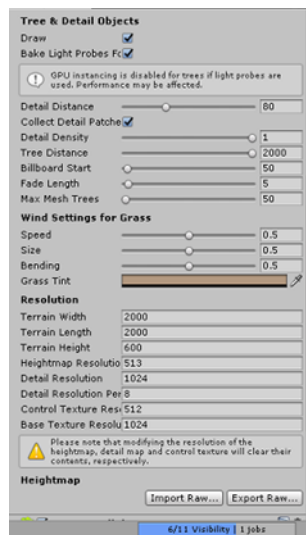


Figure:06

(System Design)

- Trigger Points

In scene1 we have made many trigger points. Trigger points are triggers that send message to the AI car to slow down or hit break. With each trigger point there is different script attached. Most of the Trigger points are along sharp turns that send message to the car to slow down and speedup. Following are the two scripts attached to them.

```

public class OnTriggerSpeedSlow : MonoBehaviour {

    // Use this for initialization
    void OnTriggerEnter(Collider other)
    {
        Debug.Log ("trigger called");
        other.gameObject.SendMessage
("SlowDown");
    }
}

```

Figure:07.1

(Coding of Trigger Points)

```

public class OnTriggerSpeedUp : MonoBehaviour {

    // Use this for initialization
    void OnTriggerEnter(Collider other)
    {
        other.gameObject.SendMessage
("SpeedUp");
    }
}

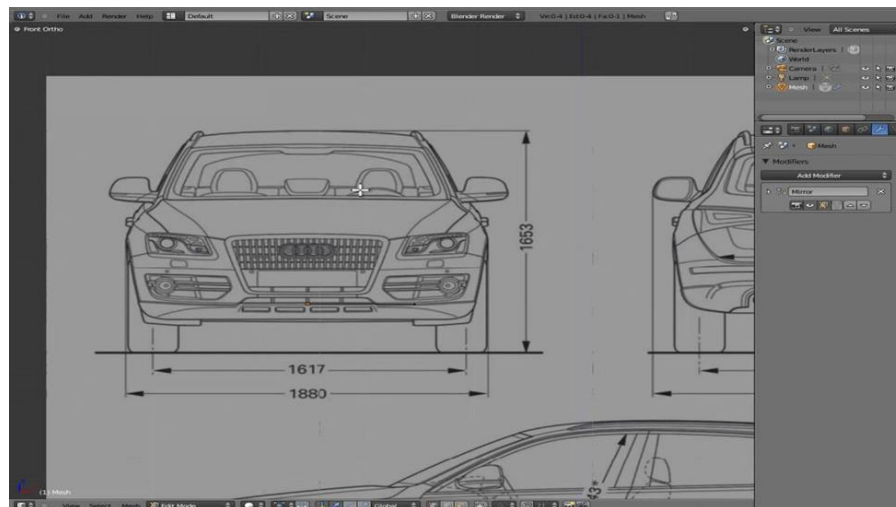
```

Figure:07.2

(Coding of Trigger Points)

4.1. Architecture Diagram

We make the game in android so the architectural diagram of our project be displayed as fellow.

**Figure:08**

(Architecture Design of Game)

4.2. Sequence/Collaboration Diagram

The sequence diagram is useful because it show the interaction logic between the objects in the system in time order that the collaborations happen. It demonstrate the exchanging of paths between object.

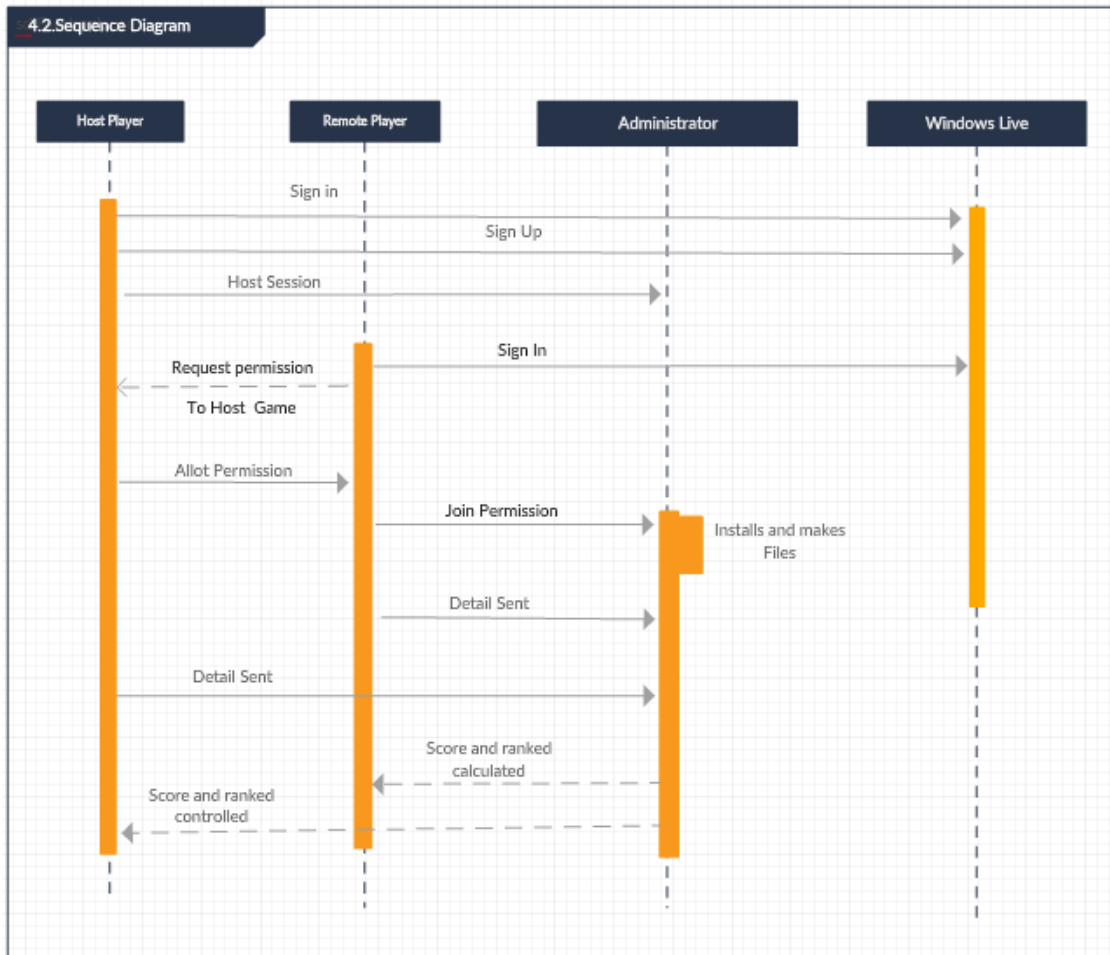


Figure:09

(Sequence of functions going on)

Operation contracts:-

Our project type doesn't requires any specific operational contracts. We does not requires any out sourced contracts for the development, implementation and working of our project.

4.3. Activity Diagram

The activity diagram is use to provide the view of flows and what is happening inside the game. It show the physical view of a system or architecture of a system as the deployment of software/system artifacts to deployment targets. The artifacts show the solid elements in physical world that is the development process.

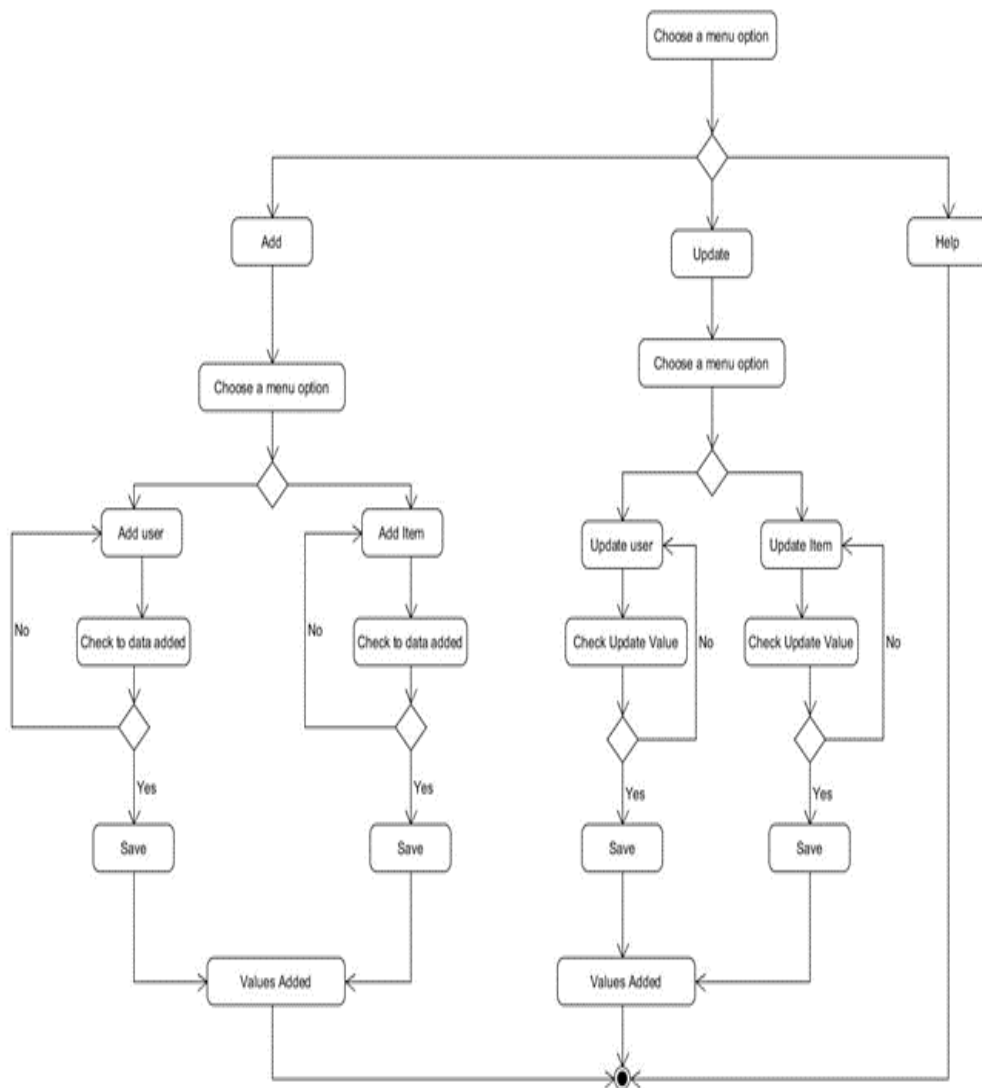


Figure:10

(Activity diagram of game.)

4.4. Data Flow diagram

It does not describe the functionality of the system but it describes the components used to make those functionalities and it help to understand The Data flow diagram is useful because it show the interaction logic between the objects in the system in time order that the collaborations happen. It demonstrate the exchanging of messages between objects that required command to complete the functionality of the system.

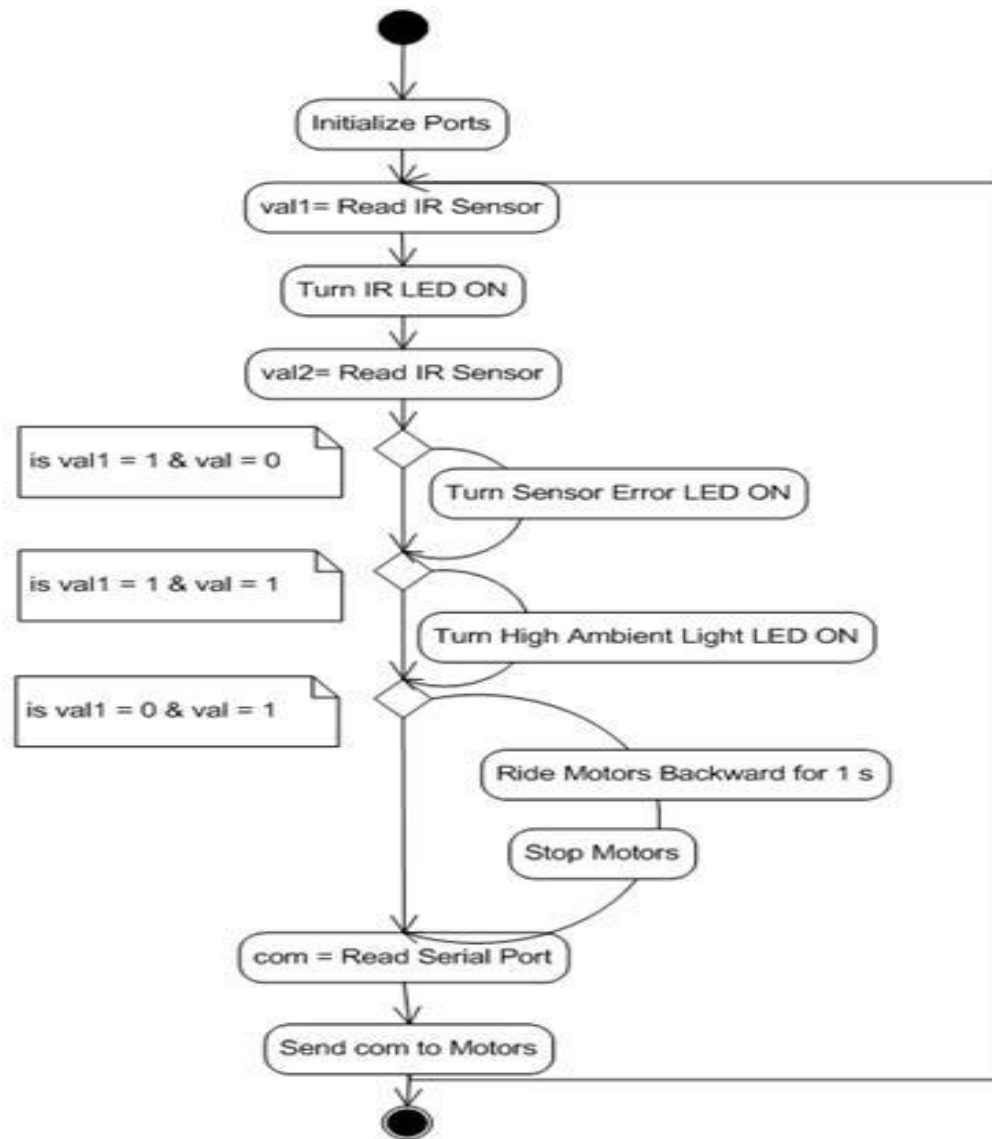


Figure:11

(Data Flow Diagram.)

Chapter 5

Implementation

Chapter 5: Implementation

The Implementation of our project consists of the following different phases. Some of the most important phase comprised of the main Flow of the Software, tools used for the development environment, languages and scripts on which our project is based upon and rest of the components of implementation.

5.1. Important Flow Control/Pseudo codes

The main Flow chart of our project, explaining the main components, their flow from one to another and all the main functionalities are as below..

- **Main Menu**

We made a simple menu system with a few buttons .First screen that pops up when we start the game in main menu system. It has two buttons one is start button and second is exit button. Start button takes us on next scene in which have to option to play.



Figure:12.1

- **UI Buttons**

Other buttons include Home button and pause buttons. Home button takes us straight to Main Menu and pause button pause the game. The Red Buttons are to control the movement of the player car. We have added a timer on scenes. When the scene start timer starts until Player hits the Finish line collider. When Player hits the finish line collider the timer turns yellow and all other buttons disappear. After race is finished we have option of replay.

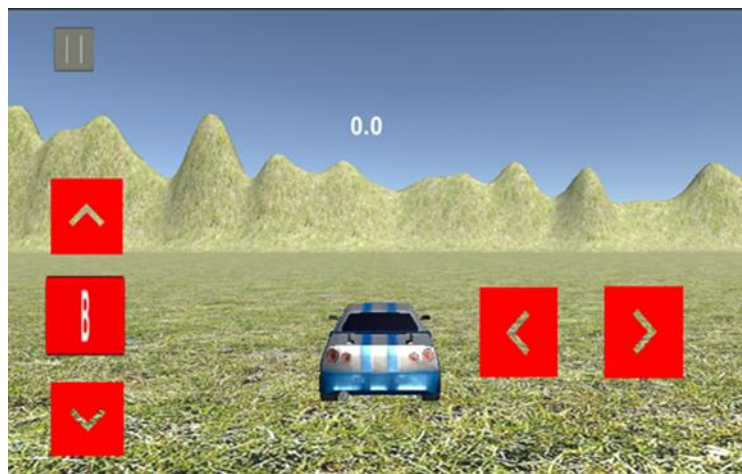


Figure:12.2

5.2. Components, Libraries, Web Services and stubs

- Main menu
- Ui Buttons
- Feature Extraction
- Feature Classification
- Libraries
- Android studio
- 3D unity

5.3. Deployment Environment

In our software product development, the development environment is a set of programming tools and process used to create the program. It includes testing, staging, QA and production, however there is no limit to create useful deployment environments.

5.4. Tools and Techniques

- **C#**
C# Language is used for programming in Unity. All the programming is done in c#. The IDE used for programming is Mono Develop.
- **Unity 3D engine**
We used Unity gaming engine to do all the designing. Creating terrains, textures, graphics and programming.
- **Blender**
Blender is a professional, free and open-source 3D computer graphics software tool set used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games. Blender features include modeling texturing rigging animating, graphics and compositing.

5.5. Best Practices / Coding Standards

We are following android and C# in this project in this project as it offers an easy way to recognize over its simplicity plus suppleness of its language. C# is also very interesting language which have modification as well. It is object oriented and functional language.

```

private void ApplySteer()
{
    Vector3 relativeVector =
transform.InverseTransformPoint(nodes [currentNode].position);
float newsteer = (relativeVector.x /
relativeVector.magnitude)*maxSteerAngle;
wheelFR.steerAngle = newsteer;
wheelFL.steerAngle = newsteer;
}

private void Drive()
{
    currentSpeed = 2 * Mathf.PI * wheelFL.radius * wheelFL.rpm *
60 / 1000;
    if (currentSpeed < maxSpeed) {
        wheelFL.motorTorque = maxMotorTorque;
        wheelFR.motorTorque = maxMotorTorque;
    } else {
        wheelFL.motorTorque = 0;
        wheelFR.motorTorque = 0;
    }
}

private void CheckWayPointDistance()
{
    if (Vector3.Distance (transform.position, nodes
[currentNode].position) < 10.5f) {
        if (currentNode == nodes.Count - 1) {
            currentNode = 0;
        } else {
            currentNode++;
        }
    }
}
}

```

Figure:13.1

```

private void LerpToSteerAngle()
{
    wheelFL.steerAngle = Mathf.Lerp (wheelFL.steerAngle,
targetSteerAngle, Time.deltaTime * turnspeed);
    wheelFR.steerAngle = Mathf.Lerp (wheelFL.steerAngle,
targetSteerAngle, Time.deltaTime * turnspeed);
}

public void SlowDown()
{
    Debug.Log ("SlowDown called");
    wheelFL.brakeTorque = 300f;
    wheelFR.brakeTorque = 300f;
    this.maxSpeed = 50f;
    wheelFL.motorTorque = maxMotorTorque;
    wheelFR.motorTorque = maxMotorTorque;
}

public void SpeedUp()
{
    this.maxSpeed = 100f;
}
}

```

Figure:13.2

Appendix A: Information / Promotional Material

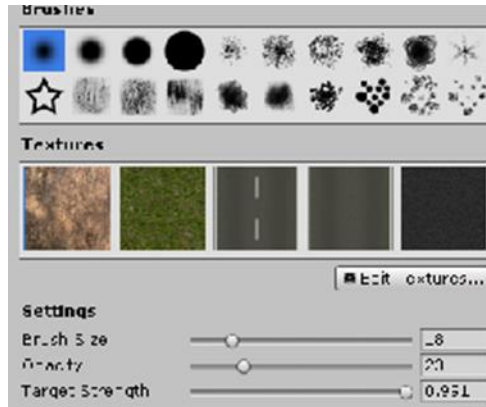


Figure:14.1



Figure:14.2



Figure:14.3

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product.

Meets the business and technical requirements that guided its design and development.

Testing was carried out on a weekly basis to gauge which aspects of the gamewere most important to users, what issue were and which part the most enjoyable.

6.1 Game Testing Steps:

- Check whether Game starts correctly or not.
- Check whether Play button is working correctly or not.
- Whether level selection is working correctly or not.

Game start Test Case:

Test Case Name: Test Game start			
Pre-Condition		Game is installed	
Sr.	Action	Expected Result	Status
1	Press the game icon "Car game"	Game starts after 4,5 seconds	OK

Table:02

Test Game Start (Test Case)

Test Case Play button:

Test Case Name: Test play button			
Pre-Condition	Main menu is open		
Sr.	Action	Expected Result	Status
1	Press the play button	Level selection open	OK

Table:03

Test Play Button (Test Case)

Test case Pause:

Test Case Name: Test Pause button			
Pre-Condition	Game play is open		
Sr.	Action	Expected Result	Status
1	Press the pause button	Pause panel is open	OK

Table:04
Test Pause (Test Case)

Test case Level Complete:

Test Case Name: Test level complete Selection			
Pre-Condition		Target achieved.	
Sr.	Action	Expected Result	Status
1	Target achieved.	Level complete panel is open.	OK

Table:05

Test Level Complete (Test Case)

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

We have developed a game you can use to play. This game is specially for kids but elders can also use to play this game.

7.2. Achievements and Improvements

By building this project, we have learned a lot of things during developing. Like different languages, different software to design graphics and a lot of other things specially how to do documentation of project. In start we have do a wrong things which we changed by time.

7.3. Future Enhancements/Recommendations

In future, we added a lot of other cars in our project. Also we have add a lot of other tracks in our game. And we launched it later when we think that now this is abled to launched on world wide level.

Reference

Reference

1. Riccitiello, John (October 23, 2014). "John Riccitiello sets out to identify the engine of growth for Unity Technologies (interview)". *Venture Beat (Interview)*. Interview with Dean Takahashi. Retrieved January 18, 2015.s
2. "Unity - does indie gaming's biggest engine have an image problem?". *The Guardian*. Retrieved August 14, 2015.
3. "Mobile game developer survey leans heavily toward iOS, Unity". *Gamasutra*. May 24, 2012.
4. Smykill, Jeff (August 9, 2006). "Apple Design Award winners announced". *ArsTechnica*. Retrieved January 18, 2015.
5. McElroy, Griffin (August 20, 2013). "Unity for Wii U opens up GamePad hardware and more to developers". *Polygon*. Retrieved August 4, 2014.
6. Davidson, A. (2013) 'The Saudi Marathon Man', *The New Yorker*, 16 April. Available at: <http://www.newyorker.com/news/daily-comment/the-saudi-marathon-man> (Accessed: 22 June 2015).
7. Guy, J. (2001) *The view across the river: Harriette Colenso and the Zulu struggle against imperialism*. Charlottesville, Virginia: University Press of Virginia.
8. Turner, J.M.W. 1812, *Snow storm: Hannibal and army crossing the alps*, Oil on canvas, 145x2360 mm, Turner Collection, Tate Gallery, Britain.
9. Ubisoft Entertainment 2012, *Assassins creed III*, computer program, Ubisoft Australia.
10. Homer (1997) *The Iliad*. Translated by J. Davies. Introduction and notes by D. Wright. London: Dover Publications.
11. Lucas, G. (2004) *The wonders of the Universe*. 2nd edn. Edited by Frederick Jones, James Smith and Tony Bradley. London: Smiths.
12. *Medicine in old age* (1985) 2nd edn. London: British Medical Association.
13. Jones, B., (1997) *Methods in tumour research*. National Agency for Tumour Research, volume. 7.
14. Peart, N. (1976) *Something for Nothing*. Toronto: Toronto Sound Studios.
15. Rush (2015) [Bishopthorpe Social Club. 29 March].
16. Di Trocchio, P 2011, *Manstyle : men + fashion*, exhibition catalogue, 11 March-27 November, National Gallery of Victoria, Melbourne, Vic.
17. Eales, S (ed.) 2003, *The Jacaranda atlas*, 3rd edn, John Wiley & Sons, Brisbane.
18. Fujimoto, Tetsuo 1998, *Work 98-1, Fibre art : Hemp cloth, polyester threads, Pellon backing and natural plant dyes* Exhibited at: *Textural Space exhibition*, Surrey Institute of Art & Design, 27 Apr - 23 Jun 2001.
19. Gorman, P 2001, *The look: adventures in pop & rock fashion*, Sanctuary Publishing Ltd, London.
20. Le Corbusier, c. 1950-1954, *Notre-Dame du Haut, exterior*, Ronchamp, France, viewed 12 January 2012, *Oxford Art Online database*.

21. *Liebovitz, A 2002, Dancers: photographs by Annie Liebovitz, Smithsonian Institution Press, Washington.*
22. *Mojang 2009, Minecraft, video game, Xbox 360, Mojang, Sweden.*
23. *Turner, JMW 1812, Snow storm: Hannibal and army crossing the alps, Oil on canvas, 145×2360 mm, Turner Collection, Tate Gallery, Britain.*
24. *Ubisoft Entertainment 2012, Assassins creed III, computer program, Ubisoft Australia.*