**COMSATS University Islamabad,**

**Attock Campus Pakistan**

**P For Programming (Video Game)**

##### ***By***

**Hammad Ahmad Khan CIIT/FA16-BCS-010/ATK**

**Waqar Younas CIIT/FA16-BCS-061/ATK**

***Supervisor***

**Mr. Shehzad Rizwan**

***Bachelor of Science in Computer Science (2016-2020)***

**The candidate confirms that the work submitted is their own and appropriate**

**Credit has been given where reference has been made to the work of other**

**COMSATS University Islamabad,**

**Attock Campus, Pakistan**

**P For Programming (Video Game)**

***A project presented to***

**COMSATS Institute of Information Technology, Islamabad**

**In partial fulfillment**

**Of the requirement for the degree of**

***Bachelor of Science in Computer Science (2016-2020)***

***By***

**Hammad Ahmad Khan CIIT/FA16-BCS-010/ATK**

**Waqar Younas CIIT/FA16-BCS-061/ATK**

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Hammad Ahmad Khan Waqar Younas

--------------------------- -------------------------

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS) “P for Programming” was developed by **Hammad Ahmad Khan (CIIT/FA16-BCS/010)** and **Waqar Younas (CIIT/FA16-BCS/061)** under the supervision of “Mr. Shehzad Rizwan” and that in (their) opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

---------------------------------------

**Supervisor**

---------------------------------------

**External Examiner**

-------------------------------------------

**Head of Department**

**(Department of Computer Science)**

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Mr. Shehzad Rizwan”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Hammad Ahmad Khan Waqar Younas

--------------------------- ------------------------

**Abbreviations**

|  |  |
| --- | --- |
| **SRS** | Software Require Specification |

**Table of Contents**

[**1** **Introduction 8**](#_Toc46620564)

[1.1 Introduction: 9](#_Toc46620565)

[1.2 Brief: 9](#_Toc46620566)

[1.3 Relevance to Course Modules: 10](#_Toc46620567)

[1.4 Project background: 10](#_Toc46620568)

[1.5 Literature Review: 10](#_Toc46620569)

[1.6 Analysis from Literature Review: 11](#_Toc46620570)

[1.7 Methodology and Software lifecycle for this project: 11](#_Toc46620571)

[1.7.1 Rationale behind selected methodology: 11](#_Toc46620572)

[**2 Problem Definition** 14](#_Toc46620573)

[2.1 Problem Definition 15](#_Toc46620574)

[2.2 Problem Statement: 15](#_Toc46620575)

[2.2.1 Problem Solution for Proposed System: 15](#_Toc46620576)

[2.3 Deliverables and Development Requirements: 15](#_Toc46620577)

[2.3.1 Deliverables Milestone: 15](#_Toc46620578)

[2.3.2 Development Requirements: 16](#_Toc46620579)

[2.3.3 System requirements: 16](#_Toc46620580)

[2.4 Current System: 16](#_Toc46620581)

[2.4.1 Coding Quiz: 16](#_Toc46620582)

[2.4.2 Coding Planets: 17](#_Toc46620583)

[2.5 Presented solution: 17](#_Toc46620584)

[2.5.1 Comparison between Presented solution And Other Applications: 18](#_Toc46620585)

[**3 Requirement Analysis** 19](#_Toc46620586)

[3.1 Use Case Diagram 20](#_Toc46620587)

[Signup Use Case: 21](#_Toc46620588)

[3.2 Detailed Use Case: 26](#_Toc46620589)

[3.3 Functional Requirements: 27](#_Toc46620590)

[3.3.1 FR01 Display 27](#_Toc46620591)

[3.3.2 FR02 Control 27](#_Toc46620592)

[3.3.3 FR03 Device 27](#_Toc46620593)

[3.4 Non-Functional Requirements: 28](#_Toc46620594)

[3.4.1 User Interface: 28](#_Toc46620595)

[3.4.2 Performance 28](#_Toc46620596)

[3.4.3 Quality Graphics 28](#_Toc46620597)

[**4 Design and Architecture** 29](#_Toc46620598)

[4.1 System Architecture: 30](#_Toc46620599)

[4.2 Data Representation: Level 0 Diagram: 31](#_Toc46620600)

[4.3 Level 1 Diagram: 31](#_Toc46620601)

[4.4 Activity Diagram: 32](#_Toc46620602)

[4.5 Login-in Activity Diagram: 33](#_Toc46620603)

[4.6 Sign-up Activity Diagram: 34](#_Toc46620604)

[4.7 Process Flow/representation: 35](#_Toc46620605)

[4.8 Design Models: Sequence Diagrams: 36](#_Toc46620606)

[4.8.1 Main Menu Sequence Diagram: 36](#_Toc46620607)

[4.8.2 Selection Panel Sequence Diagram: 37](#_Toc46620608)

[4.8.3 Data-type Sequence Diagram: 37](#_Toc46620609)

[4.8.4 Loops Sequence Diagram: 38](#_Toc46620610)

[4.8.5 Arrays Sequence Diagram: 38](#_Toc46620611)

[4.8.6 Conditions Sequence Diagram: 39](#_Toc46620612)

[4.9 Class Diagram: 40](#_Toc46620613)

[**5**  **Implementation**  41](#_Toc46620614)

[5.1 External APIs: 42](#_Toc46620615)

[5.1.1 Firebase: 42](#_Toc46620616)

[5.2 User Interface: 42](#_Toc46620620)

[5.1.2 Login/SignUp: 42](#_Toc46620621)

[5.1.3 Main Menu 43](#_Toc46620622)

[5.1.4 Data Type Level: 44](#_Toc46620623)

[5.1.5 Loop Level: 45](#_Toc46620624)

[5.1.6 Condition Level: 47](#_Toc46620625)

[5.1.7 Arrays Level: 49](#_Toc46620626)

[5.1.8 Methods Level: 50](#_Toc46620627)

[**6 Testing and Evaluation** 51](#_Toc46620628)

[Testing and Evaluation 52](#_Toc46620629)

[6.1 Manual Testing: 52](#_Toc46620630)

[6.1.1 Unit Testing 52](#_Toc46620631)

[6.2 Functional Testing: 52](#_Toc46620632)

[6.3 Integration Testing: 53](#_Toc46620633)

[6.4 Automated Testing: 53](#_Toc46620634)

[**7 Conclusion and Future Work** 54](#_Toc46620635)

[7.1 Conclusion: 55](#_Toc46620636)

[7.2 Future Work: 55](#_Toc46620637)

[8 References 56](#_Toc46620638)

**List of figures**

[**Figure 1 1 Scrum Methodology 9**](#_Toc46596199)

[**Figure 2 1 Coding Quiz 13**](#_Toc46596099)

[**Figure 2 2 Coding Planets 14**](#_Toc46596100)

[**Figure 3 1 Login Use Case 17**](#_Toc46596109)

[**Figure 3 2 Signup Use case 18**](#_Toc46596110)

[**Figure 3 3 Main Menu Use case 19**](#_Toc46596111)

[**Figure 3 4 Level Panel Use Case 4**](#_Toc46596112)

[**Figure 3 5 Use Case Datatype 5**](#_Toc46596113)

[**Figure 3 6 Use Case Loop 6**](#_Toc46596114)

[**Figure 3 7 Use Case Conditions 23**](#_Toc46596115)

[**Figure 3 8 Use Case arrays 24**](#_Toc46596116)

[**Figure 3 9 Use Case Methods 25**](#_Toc46596117)

[**Figure 3 10 Detailed Use Case 26**](#_Toc46596118)

[**Figure 4 1 Architecture Overview 30**](#_Toc46596119)

[**Figure 4 2 Level 0 Diagram 31**](#_Toc46596120)

[**Figure 4 3 Level 1 Diagram 31**](#_Toc46596121)

[**Figure 4 4 Main Activity Diagram 32**](#_Toc46596122)

[**Figure 4 5 Sign-in Activity Diagram 33**](#_Toc46596123)

[**Figure 4 6 Sign-up Activity Diagram 34**](#_Toc46596124)

[**Figure 4 7 Process Flow/representation 35**](#_Toc46596125)

[**Figure 4 8 Main Menu Sequence Diagram 36**](#_Toc46596126)

[**Figure 4 9 Selection Panel Sequence Diagram 37**](#_Toc46596127)

[**Figure 4 10 Data-type Sequence Diagram 37**](#_Toc46596128)

[**Figure 4 11 Loops Sequence Diagram 38**](#_Toc46596129)

[**Figure 4 12 Array Sequence Diagram 38**](#_Toc46596130)

[**Figure 4 13 Conditions Sequence Diagram 39**](#_Toc46596131)

[**Figure 4 14 General Class Diagram 40**](#_Toc46596132)

[**Figure 5 1 Login Interface 42**](#_Toc46596133)

[**Figure 5 2 Signup Interface 43**](#_Toc46596134)

[**Figure 5 3 Main Menu Interface 44**](#_Toc46596135)

[**Figure 5 4 Data Type Interface 45**](#_Toc46596136)

[**Figure 5 5 Loop Interface 46**](#_Toc46596137)

[**Figure 5 6 Condition Interface 48**](#_Toc46596138)

[**Figure 5 7 Arrays Interface 49**](#_Toc46596139)

[**Figure 5 8 Methods Interface 50**](#_Toc46596140)

**CHAPTER 1**

# Introduction

## Introduction:

As in the beginning, many problems come in studying coding and that is why the programming become so tough and boring for the students so they thought that they can’t do coding. Also, those who do not know how to learn programming.

We are trying to develop an out of the box solution for the students learning through playing the game so that their programming concepts are enhanced. As you know, in our education system there is a lot burden on students. The students and the beginners think that programming is a tough task and so in this way students lose their interest in programming. Therefore, we are looking to overcome this problem through this project. That is why a lot of what is done in Introduction to Programming is not only teaching but practice is also essential.

Programming is about writing the instructions, which a computer follows to enable it to store knowledge, process knowledge, and communicate knowledge with the outside world.

Many people will say yes, it is hard to learn code. Moreover, it is a sad fact that a lot of these people give up and fail. However, what these people do not realize is that there is no such thing as an ‘instant coder’. It takes a lot of time and practice to get good at coding.

Game is a structured form of play, usually undertaken for enjoyment and sometimes used as

an educational tool. Nowadays the benefits of gaming in educational settings has been seen as an effective way to further education.

When people code, they like the fact that they have a challenge to solve. In other words, coding itself is already a game, so making a visual representation of your code that looks like a game made sense.

Our motivation is to build the youth for the better career. As there are lot of opportunities in the field of programming.

Thus, the main target of our project is to make the interest of the beginners in programming and clear the concepts of basic programming through game in an attractive way.

## Brief:

The game which going to design will be an educational purpose game. The purpose is trying to grow the interest of the beginners in the programming by teaching them the fundamentals of programming like data types, conditions, loops, arrays and different methodologies, etc. There would be certain levels in this game, first user will create his account by filling the registration form, and the information of user will be stored in online database.

The user will login and play the game. The user could visit his account and check his rating, and see the leader board. We will use Unity tool for Game Development, and we will use C# language for scripting. We will use the Scrum (agile method) methodology for software development.

## Relevance to Course Modules:

Our project related courses are Introduction to Computer Programming (ITCP), Object Oriented Programming (OOP) and Visual Programming (VP). We used C# language for scripting of the game. Requirement Gathering, Project Documentation, Planning, Software Development Methodologies and Feasibility Assurance etc. we studied in SE-1 course, all are useful when we come to software planning and design. We learnt about Software Engineering Phases on which we will analyze each of the Phases through Iterative Software Development Methodology. We will draw many types of diagram i.e. Context Diagram, Use Case Diagram, Data Flow Diagram and sequential Diagrams etc. that we learnt in SE-1 for the Conceptual Model of our Project and understanding of supervisor.

Along with these courses although all the courses have their own importance in the development of our Project. Communication Skills, Professional Practices in IT, Human Computer Interaction (HCI).

## Project background:

As, the beginners in programming learn by imitation. That’s how their brain works. In a word, they scan the adult behavior and copy it completely or partially during the crucial first year of growth.

That’s why it’s essential to filter the surroundings and create a kind of beautifully intelligent world around your beginners. The use of serious games for educational purposes can have a very positive effect, since they provide an effective way to engage students in learning activities and have the ability to stimulate cognitive processes like acknowledging displayed information, deductive and inductive reasoning, and also problem-solving. In a serious game the fun factor must be maintained in the first place. Therefore, to provide a rich and engaging environment that presents educational goals the best practices of pedagogical psychology is needed. But the best pedagogic practices are those that are interpreted as fun.

## Literature Review:

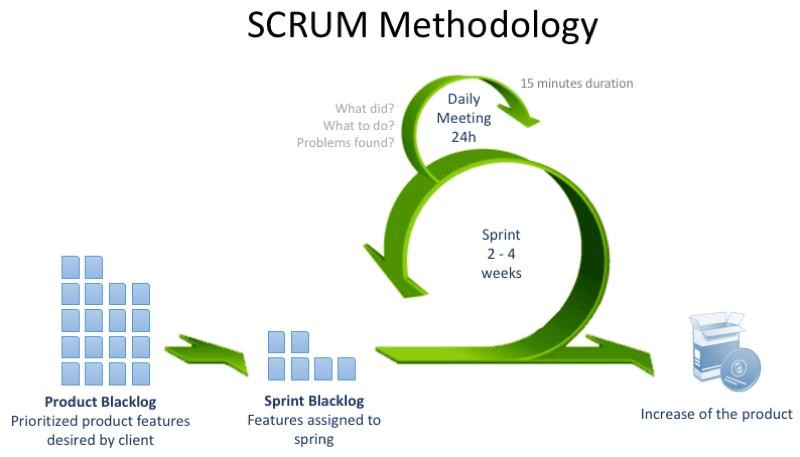
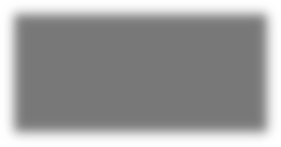
There are some games like Programming hero, Algorithm city and Coding game for Kids uploaded on Google Play store, but these games are either 2D games or tutorial-based games and none of them is using the concept of a 3D game.

## Analysis from Literature Review:

The major drawback of these games is that they nullified the idea of game in these applications. Our main purpose would be to create an environment where the user spends their time in effective way, there would be fun as well as they would gain concepts of programming, we will try to use the 3D game-based approach to deliver programming concepts.

## Methodology and Software lifecycle for this project:

For the improvement of our game, we have adopted the Scrum (agile method) methodology.



**Figure 1 1 Scrum Methodology**

The game is divided into several modules (Sprint) and every module have a different functionality. When work would done on any module it will delivered to the supervisor in increments.

### Rationale behind selected methodology:

We have selected this life cycle because we are developing our project in an incremental form and showing the result to supervisor If any changes are require then the supervisor will tell us and these changes will be accommodated to our project.

This project is divided into several Modules:

* + - **Data types module**

* + - **Loop module**
    - **Conditions module**
    - **Arrays module**
    - **Methodologies module**

These all modules are accommodated in 5 sprints:

**Sprint 1:**

In first sprint, the quiz’s module and data type module will developed. There are some activities in this module. This section consists of MCQ’s which would be related to the current level (finished by the user). It would be in 2D game.

**Sprint 2:**

In second sprint, the loop module will be developed. There will be a game, which is based on the concept of loops. It is a 3D game.

*For Example:*

If the person has to collect 20 apples in the bucket then, he will climb up the tree and pick an apple and put it in the bucket. Again, he will climb up the tree and pick 2nd apple and put in the bucket. He will repeat this process ‘n’ times and continue this iteration until the loop finish and 20 apples will store in the bucket.

**Sprint 3:**

In the third sprint, we will develop a 3D racing game which clears the concept of conditions. It would be 3D game.

*For Example:*

If the player wants to start a race. So, before the game starts the player will fulfill some conditions like the player will choose cars, tyres and environment etc. Let’s suppose if the player starts a game firstly, he will choose the environment. If he will choose the rocky environment, then he will have to decide the type of car used for this area else the tyres used in rocky areas. After fulfilling all the conditions, the game will start.

**Sprint 4:**

In the fourth sprint, we will develop a game in which the idea of arrays is included. It is a 3D game.

*For Example:*

The person creates an array of 10 garages and he has to park the 1 car in each garage. Now, if the user wants to park more than 10 cars the array will overflow and the error occurs. If the garage 5 and 6 are free and user wants to park the 6th car, so, the user parks this car in 6th garage not in the 5th garage.

**Sprint 5:**

In the fifth sprint, we will develop the game is included methodologies. It would be 3D game.

*For Example:*

There would be a car and we have to update it..

**CHAPTER 2**

# Problem Definition

## Problem Definition

As in the beginning, a lot of problems come in studying coding and that’s why the programming become so tough and boring for the students so they thought that they can’t do coding. Also, those who do not know how to learn programming.

## Problem Statement:

To facilitate the students in overcoming their fear of learning new difficult coding structures by providing them an interactive, fun filled and activity-oriented game.

### Problem Solution for Proposed System:

We are thinking for an out of the box solution, as far as our educational system is concerned, we see that there is a lot burden on students. Students and the beginners thought programming as a tough task, so in this way most of the students lose their interest in programming. So, we are looking to overcome this problem and we took a game as a tool to teach the basics of programming. I believe we can develop these skills through practice. That is why a lot of what is done in Introduction to Programming is not teaching but practice.

## Deliverables and Development Requirements:

### Deliverables Milestone:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Deliverable Name** | **Description** | **Stake Holders** | **Month** | **Status** |
| 1. | Project proposal | Determination of idea. | Group members  And supervisor. | September  2019 | Completed |
| **2.** | Scope and Document | Planning the milestone, timeline of project, feasibility check, defining modules and  Scope of project. | Group members | September 2019 | Completed |
| **3** | Preparation of first 4 chapters of documentation | Introduction, Problem Definition, Requirement Analysis, Design and  Architecture. | Group members and supervisor. | November 2019 | Completed |
| **4** | 30%  implementation of work | Implementation of interfaces and 1st two levels of project | Group members and supervisor | Nov-Dec 2019 | Completed |

### Development Requirements:

* Unity 2d and 3d
* Visual Studio

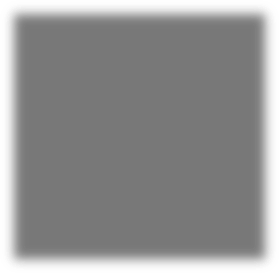
### System requirements:

* + - Operating System: Window 7/8/10
    - CPU: Intel core i3
    - RAM: 250MB of RAM required
    - Hard Disk Required, 500MB of free space required.

## Current System:

### Coding Quiz:

**Figure 2 1 Coding Quiz**



Using this Coding Quiz game learn, spend some free time with it and have fun playing it. It has 3 types which is simple quiz, time quiz and hard quiz. In Coding Quiz Simple you can spend time and think about the answer. In Coding Quiz Time, you must answer the questions within seconds. In Coding Quiz Hard there is no multiple option to choose there is shuffled characters to answer them.

### Coding Planets:

**Figure 2 2 Coding Planets**

Coding Planets is Educational, Brain preparing and confuse amusement. In this amusement, you need to give directions for a robot to explain the riddles. Each age can play the amusement. These days, Programming is exceptionally fundamental. What's more, every nation building up the applications for getting the hang of programming so we built up this for neighborhood.

## Presented solution:

This project is a 3D video game which main aim is to teach the basic concept of programming through a game. The user will learn the concept of programming as well as they will enjoy at a same time. Concept of programming cannot learn in pressure situation. That is why our main purpose is to ease the burden of students and learn in effective way.

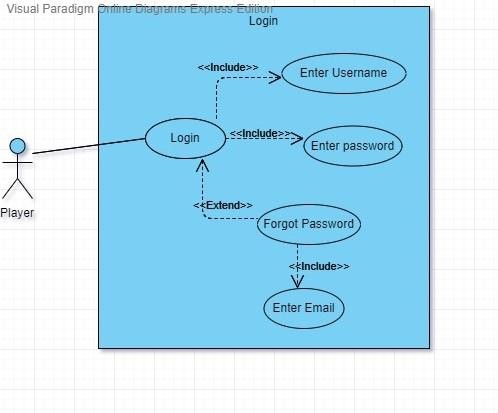
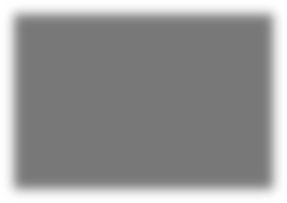
### Comparison between Presented solution And Other Applications:

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Coding Quiz** | **Coding Planets** | **P for Programming** |
| 3D video Game |  |  |  |
| Cover All basics of  Programming |  |  |  |
| Entertainment |  |  |  |
| Player Profile |  |  |  |

**CHAPTER 3**

# Requirement Analysis

## Use Case Diagram



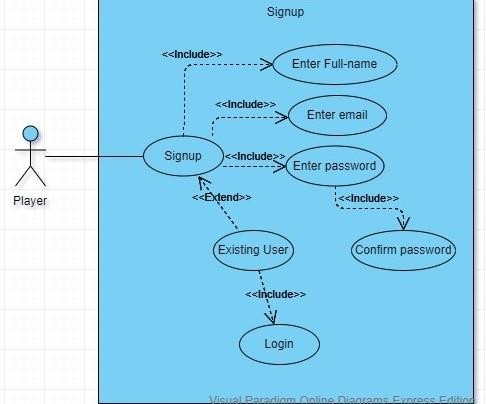
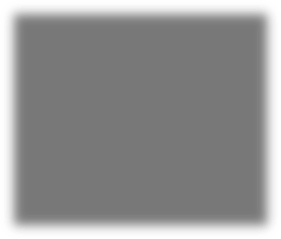
**Login Use Case:**

**Figure 3 1 Login Use Case**

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-1** |
| **Use Case Name:** | Login use case |
| **Actors:** | Player |
| **Description:** | Player will login to their account by entering email id and password. |
| **Preconditions:** | Player must have to open the application |
| **Post conditions:** | Player will successfully login to their account. |
| **Normal Flow** | Player clicks Play Button to open the selection Panel |
| **Alternative Flow** | If player enter wrong entries the account will not login |

***Table 3.1 UC-***

## Signup Use Case:

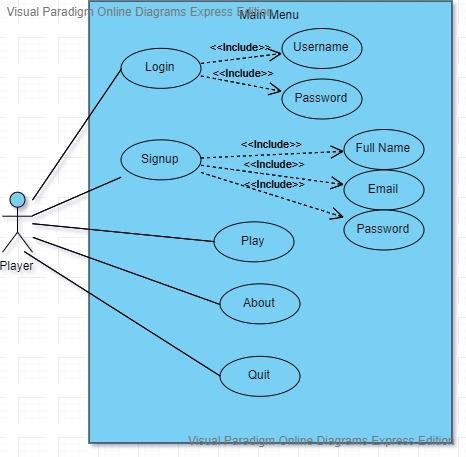
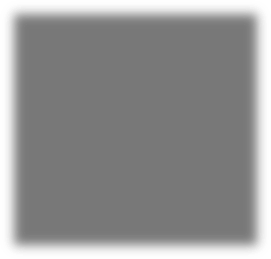


**Figure 3 2 Signup Use case**

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-2** |
| **Use Case Name:** | Sign up use case |
| **Actors:** | Player |
| **Description:** | Player will create an account by entering email id, full name, and password and confirm password. |
| **Preconditions:** | Player must have to open the application |
| **Post conditions:** | Player will successfully create their account. |
| **Normal Flow** | Player clicks Play Button to open the selection Panel |
| **Alternative Flow** | If player enter wrong entries the account will not be created or if the email already exists in the database the account will not be created. |

***Table 3.2 UC-2***

***Main Menu Use Case:***

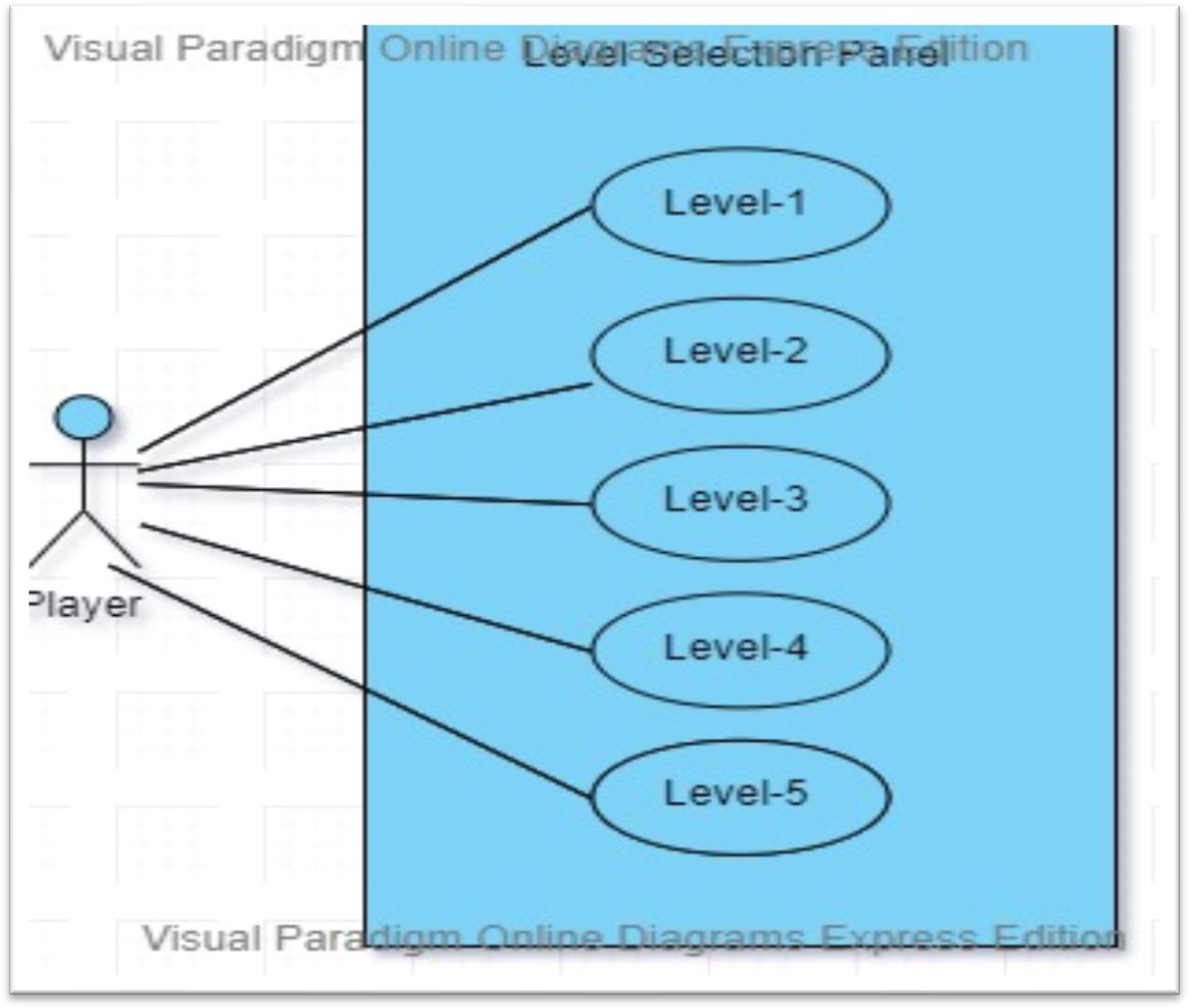


**Figure 3 3 Main Menu Use case**

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-3** |
| **Use Case Name:** | Main Menu use case |
| **Actors:** | Player |
| **Description:** | Player wants to select one of the modules. |
| **Preconditions:** | Player must have to open the application |
| **Post conditions:** | Player successfully open one of the modules. |
| **Normal Flow** | Player clicks Play Button to open the selection Panel |

***Table 3.3 UC-3***

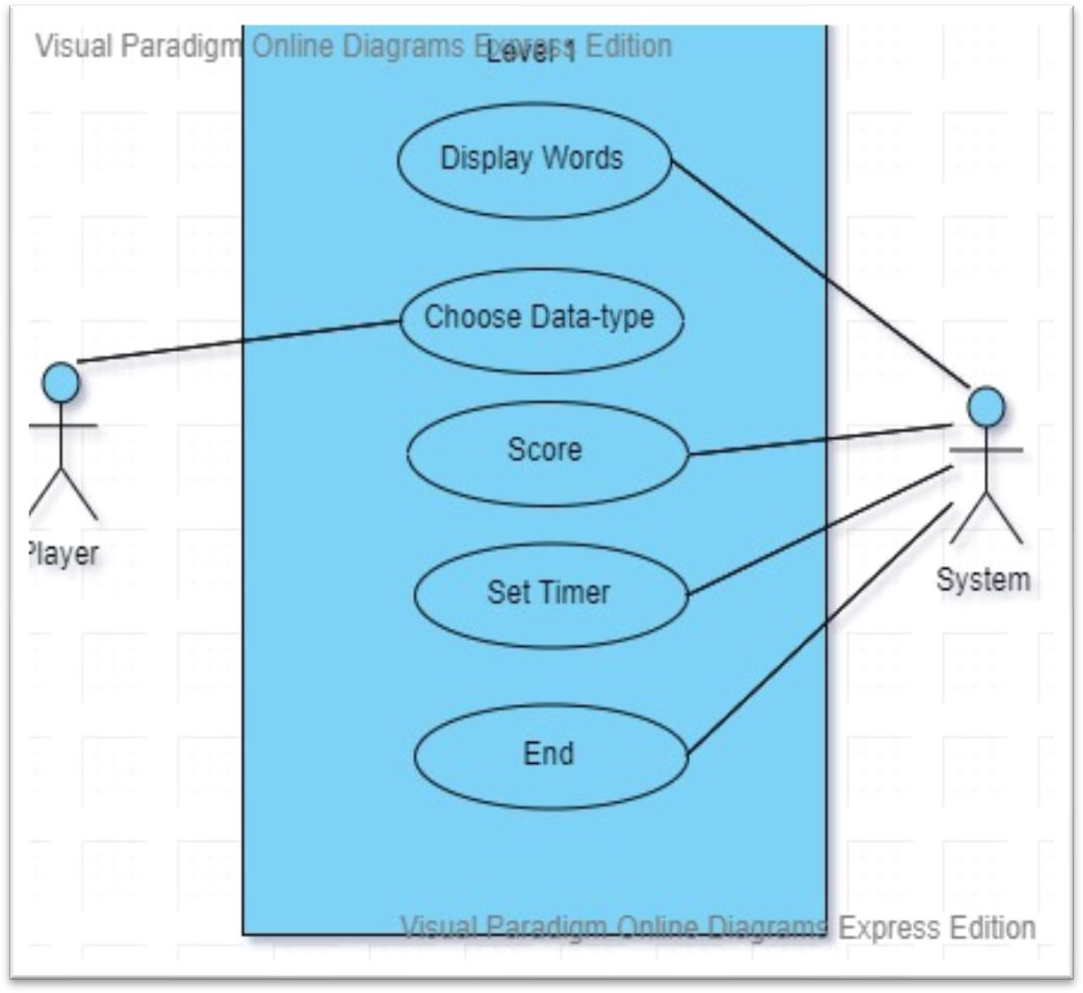
**Level Panel Use Case:**



**Figure 3 4 Level Panel Use Case**

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-4** |
| **Use Case Name:** | Level Panel use case |
| **Actors:** | Player |
| **Description:** | Player wants to open the one of the levels to play the game. |
| **Preconditions:** | Player must be opened the home screen |
| **Post conditions:** | Player successfully selects the one of the levels |
| **Normal Flow** | Player opens the level of their own desire. |

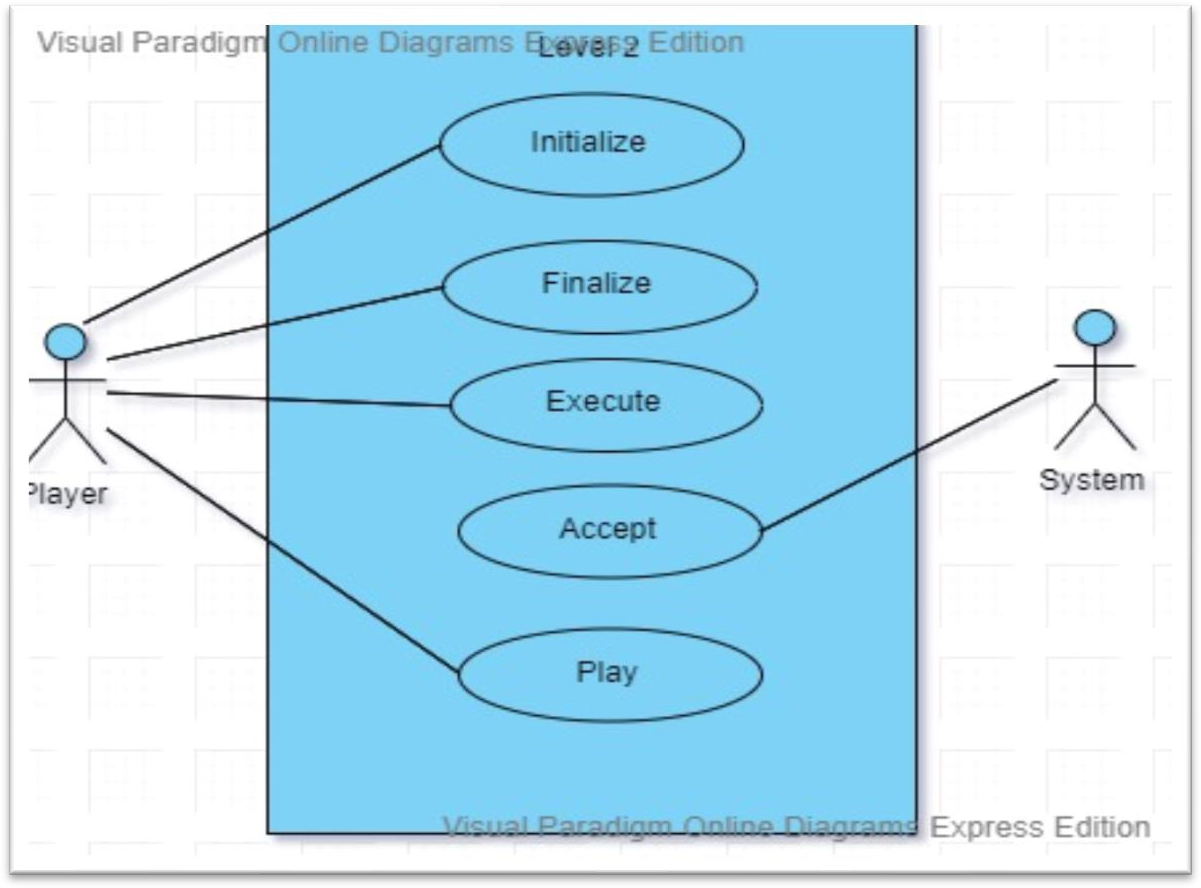
**Use Case Datatype level:**



**Figure 3 5 Use Case Datatype**

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-5** |
| **Use Case Name:** | Level 1 use case |
| **Actors:** | Player, System |
| **Description:** | Player must have to understand the concept/basics of data type |
| **Preconditions:** | Player must have opened the data type level |
| **Post conditions:** | Scores should be added by selecting the right data type |
| **Normal Flow** | Scores must be added |

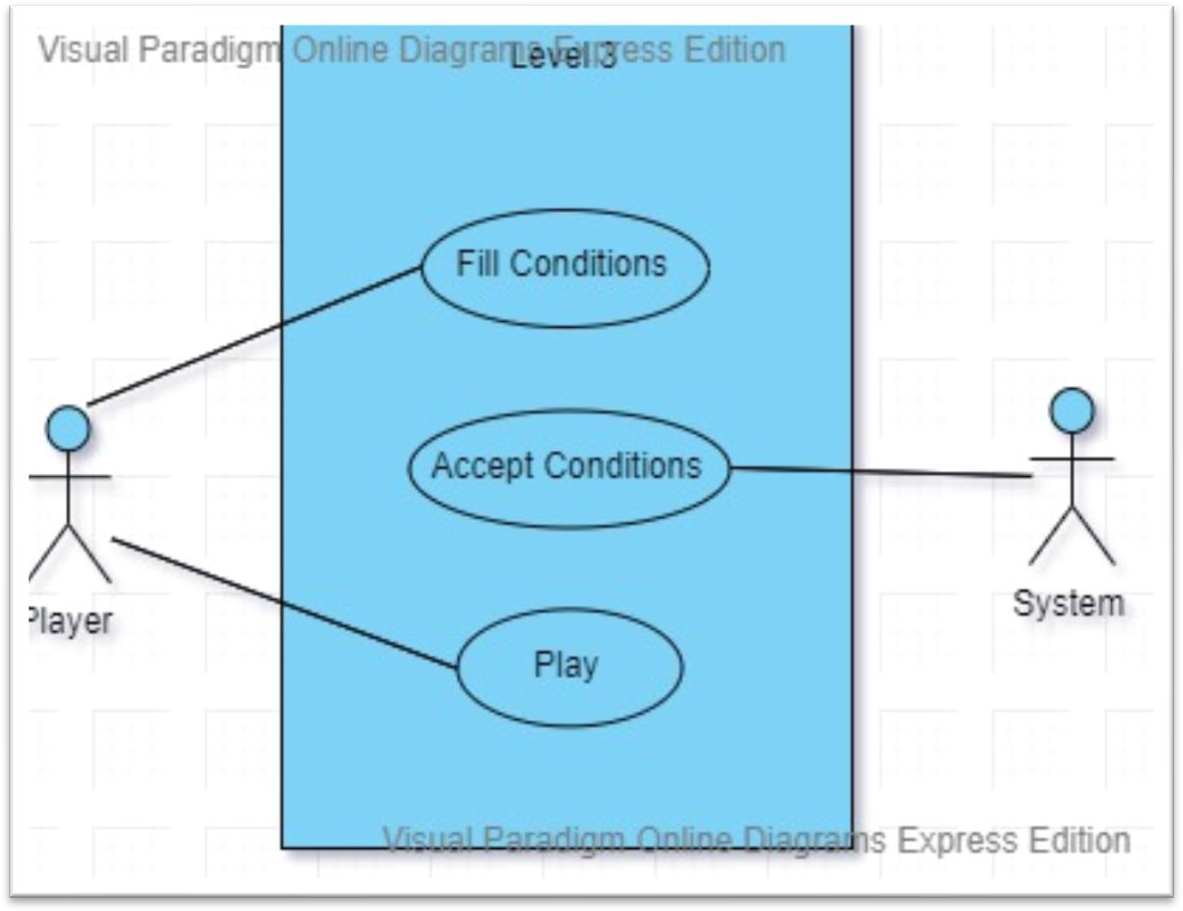
**Use Case Loop level:**



**Figure 3 6 Use Case Loop**

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-6** |
| **Use Case Name:** | Level 2 use case |
| **Actors:** | Player, System |
| **Description:** | Player must have to understand the basics of loops |
| **Preconditions:** | Player must succeed in the data type level |
| **Post conditions:** | Player successfully fulfil the loop syntax |
| **Normal Flow** | System must accept the syntax |

**Use Case Conditions level 3:**

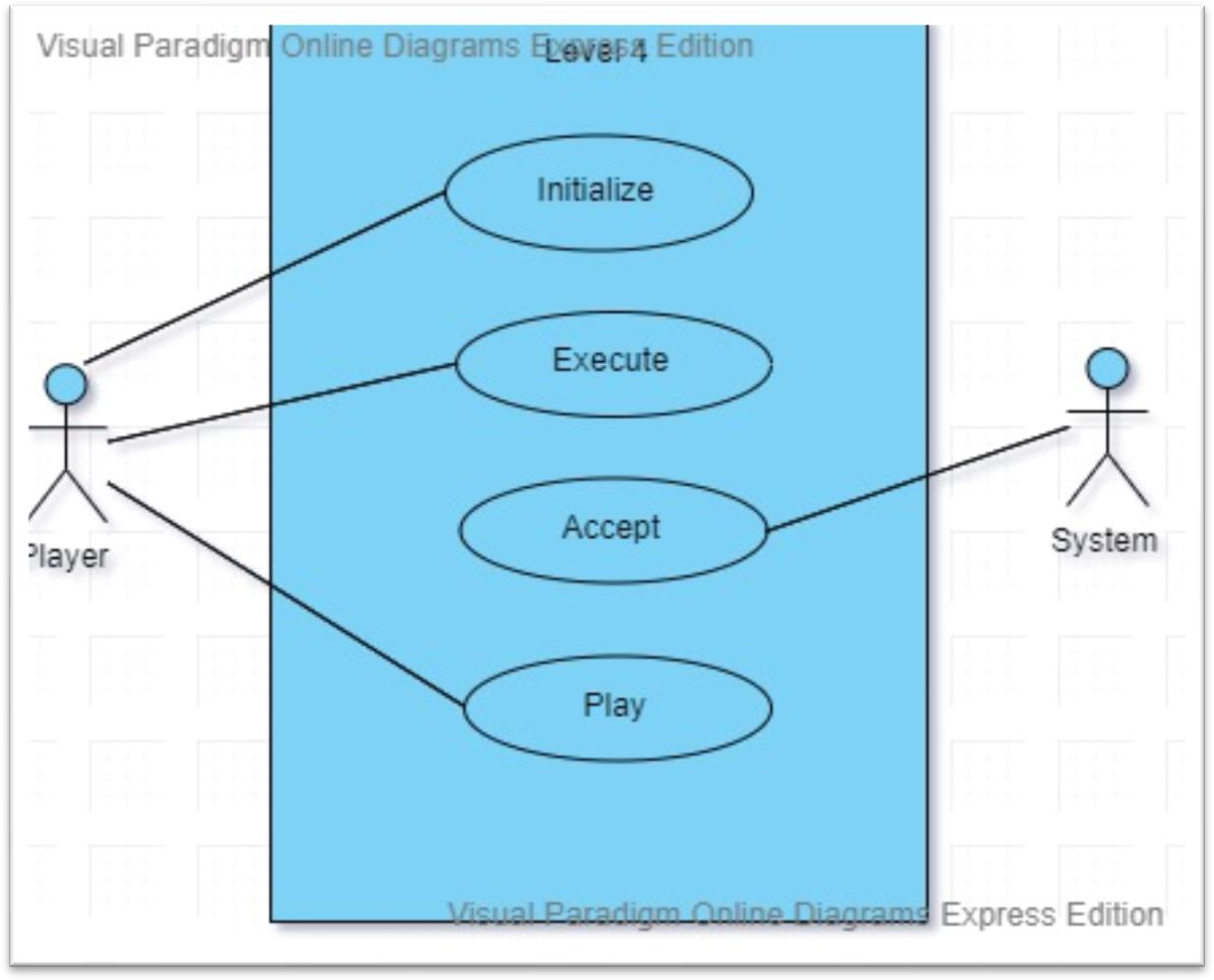


**Figure 3 7 Use Case Conditions**

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-7** |
| **Use Case Name:** | Level 3 use case |
| **Actors:** | Player, System |
| **Description:** | Player must have to understand the concept of conditions. |
| **Preconditions:** | Player must succeed in the loop level |
| **Post conditions:** | Player successfully fulfil the conditions. |
| **Normal Flow** | System must accept the condition. |
|  |  |

***Table 3.7 UC-7***

**Use Case arrays level 4:**



**Figure 3 8 Use Case arrays**

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-8** |
| **Use Case Name:** | Level 4 use case |
| **Actors:** | Player, System |
| **Description:** | Player must have to understand the basics of operators. |
| **Preconditions:** | Player must have opened the operator level. |
| **Post conditions:** | Player successfully apply operators. |
| **Normal Flow** | System must accept and execute the operators. |

***Table 3.8 UC-8***

**Use Case level 5(Methods):**

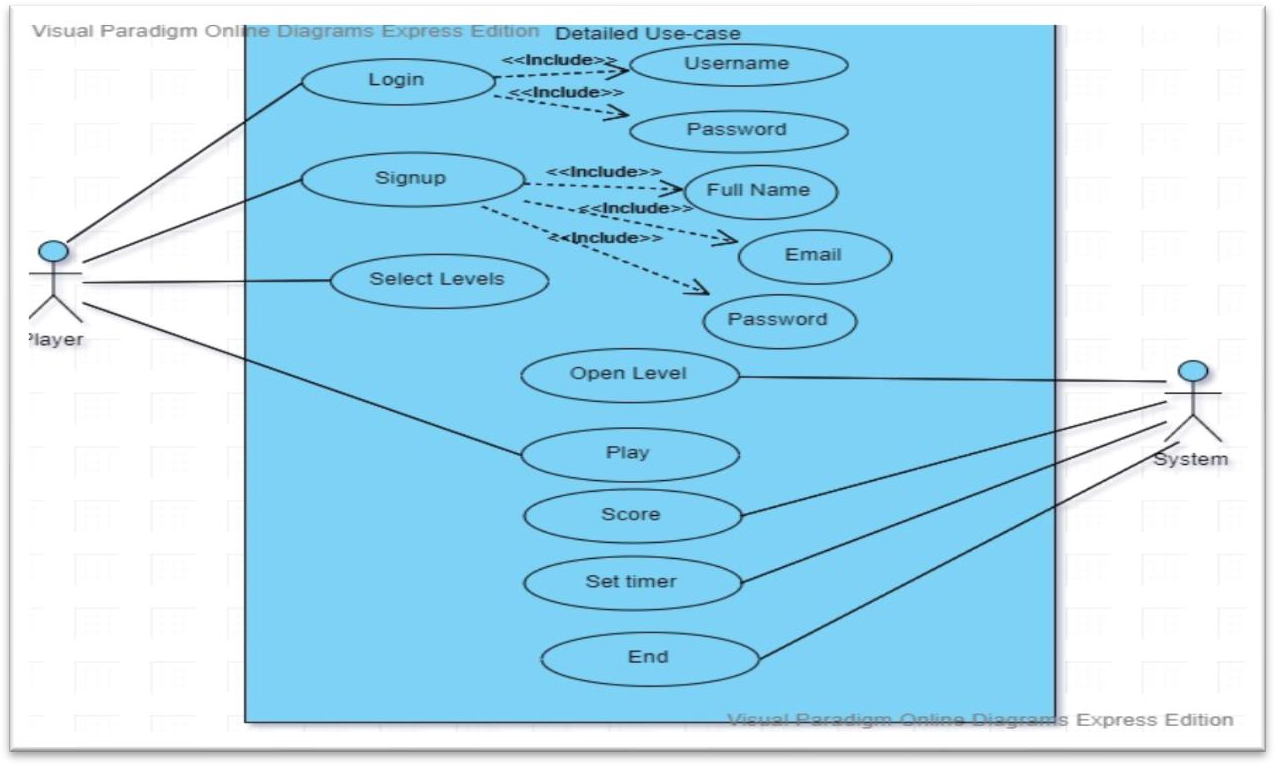


**Figure 3 9 Use Case Methods**

***Table 3.9 UC-9***

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-9** |
| **Use Case Name:** | Level 5 use case |
| **Actors:** | Player, System |
| **Description:** | Player must have to understand the basics and code of methods. |
| **Preconditions:** | Player must have opened the method level. |
| **Post conditions:** | Player successfully select the code of given method. |
| **Normal Flow** | System must execute the code |

## Detailed Use Case:



**Figure 3 10 Detailed Use Case**

***Table 3.10 UC-10***

|  |  |
| --- | --- |
| **Use Case ID:** | **UC-10** |
| **Use Case Name:** | General use case |
| **Actors:** | Player, System |
| **Description:** | Player can register himself then play the game |
| **Preconditions:** | Player must have to open the main menu |
| **Post conditions:** | Player will quit from the game |
| **Normal Flow** | * 1. Player will register himself   2. If a user has already register then he will login   2.0 Player will select the play button  3.0 System will open the level menu  4.0 Player will start playing  5.0 After end of game System will calculate the marks of Player  6.0 User will quiz from game |

## Functional Requirements:

The Functional Requirement of any system can relate to both hardware and software in terms of specific functionality, which can defines that what can a specific system able to accomplish. A functional requirement is in the form of professional document, which explains the desired output when system is deploys in certain kind of environment.

### FR01 Display

***Table 3.11 FR-01: Display***

|  |  |
| --- | --- |
| FR No | FR-01 |
| FR Name | Display |
| Summary | The game must be play on PC and android phones. When the player opens the game, the game resolutions will be set according to the device requirement. |

### FR02 Control

***Table 3.12: Control***

|  |  |
| --- | --- |
| FR No | FR-03 |
| FR Name | Control |
| Summary | The game must be control with default keys. Player will use different buttons for different functionalities in the game. |

### FR03 Device

***Table 3.13: Device***

|  |  |
| --- | --- |
| FR No | FR-04 |
| FR Name | Device |
| Summary | The device should be able to play 3D |

## Non-Functional Requirements:

### User Interface:

***Table 3.14: NFR-01 User Interface***

|  |  |
| --- | --- |
| NFR No | NFR-01 |
| NFR Name | User Interface |
| Summary | User interface should be clean and compatible with PC and UI must be control through buttons. The UI of our game is very easy and understandable. All the functions in the game visible. Even a new user can also understand the interface of game in a very first go. |

### Performance

***Table 3.15: NFR-02 Performance***

|  |  |
| --- | --- |
| NFR No | NFR-02 |
| NFR Name | Performance |
| Summary | The primary performance requirements are the speed of the gameplay. It must be with good FPS on system to have more reality. |

### Quality Graphics

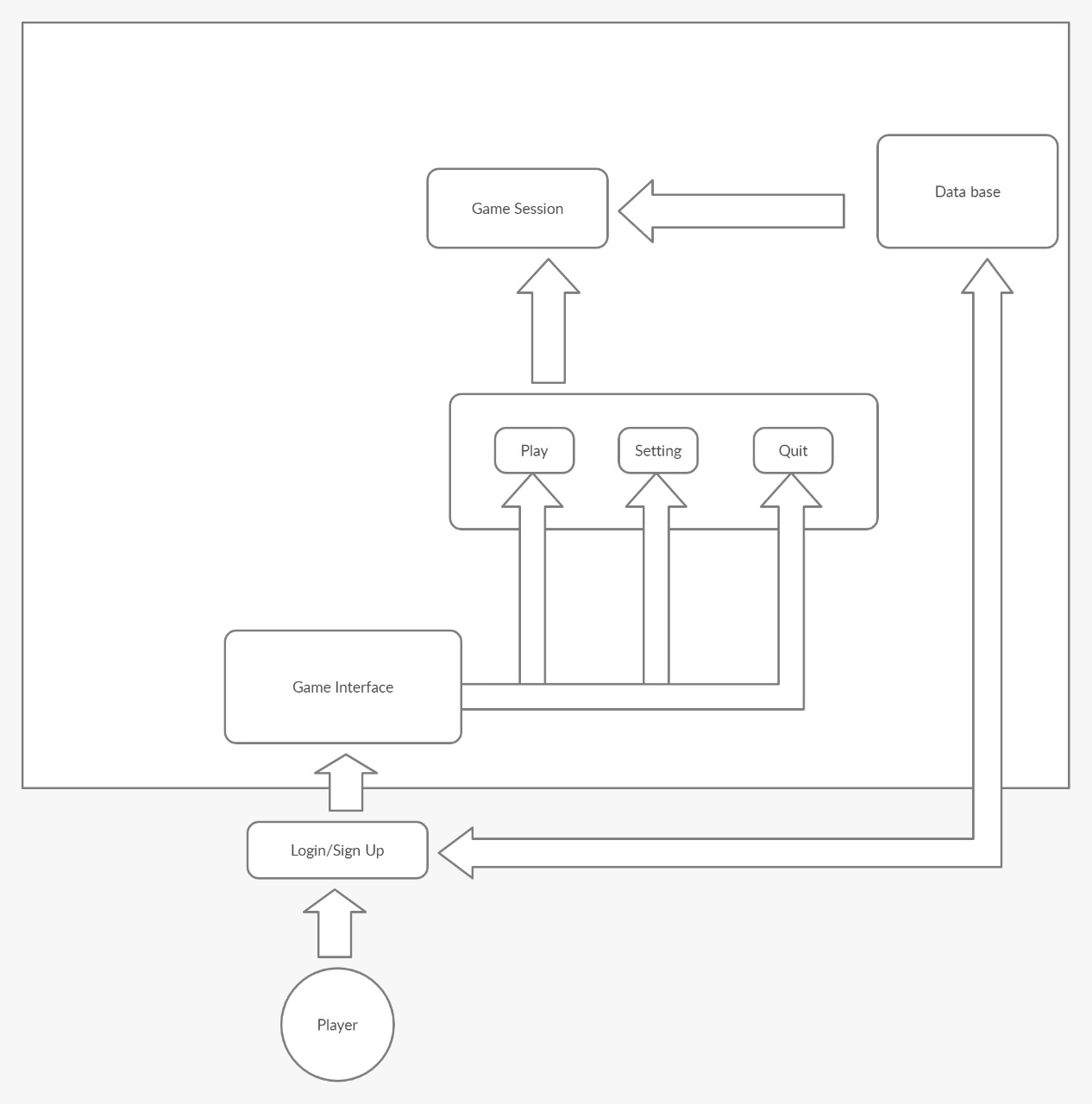
***Table 3.16: NFR-03 Quality Graphics***

|  |  |
| --- | --- |
| NFR No | NFR-03 |
| NFR Name | Quality Graphics |
| Summary | The game must have high-end 3D graphics. |

**CHAPTER 4**

# Design and Architecture

## System Architecture:

****

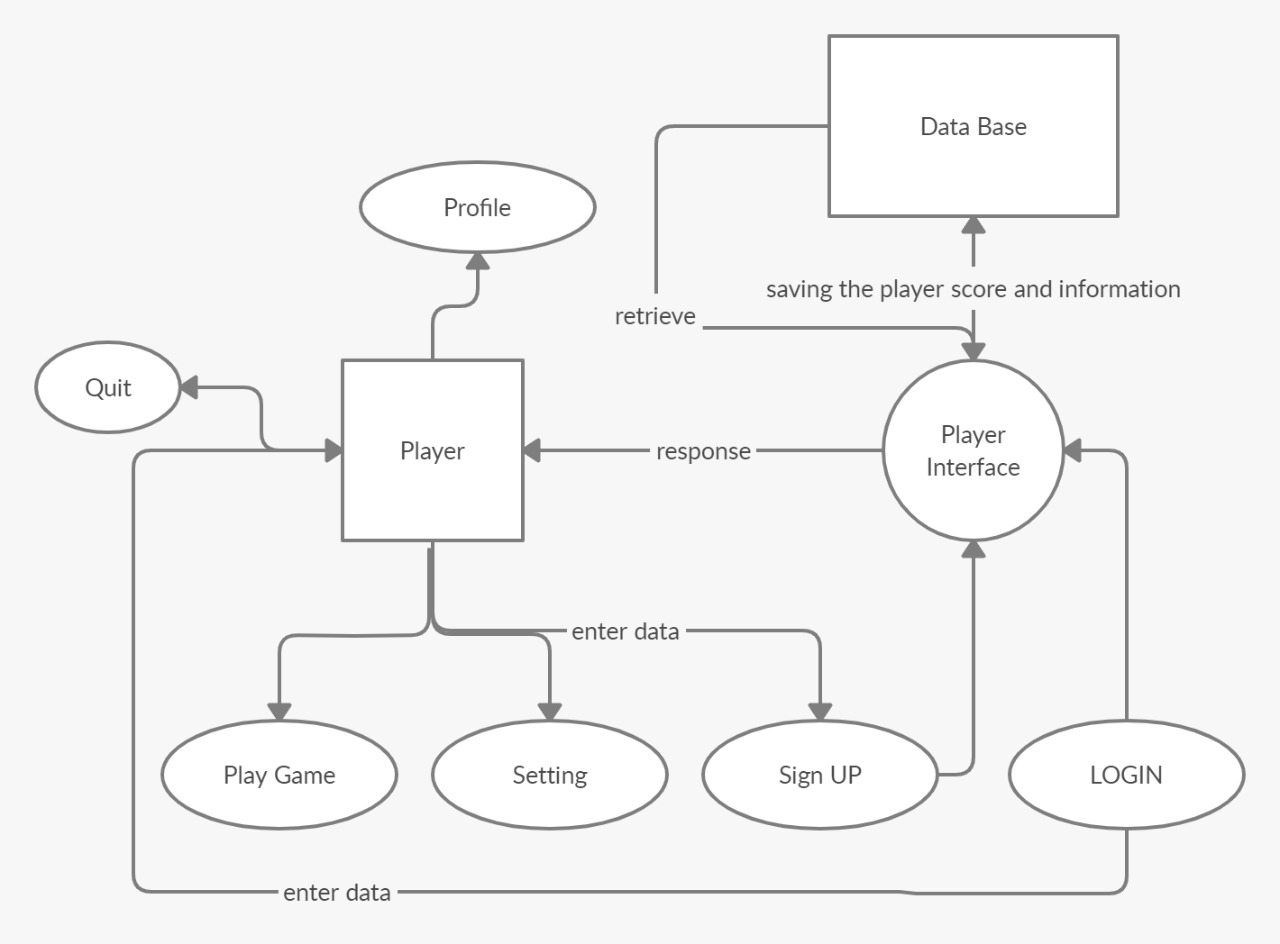
**Figure 4 1 Architecture Overview**

## Data Representation: Level 0 Diagram:



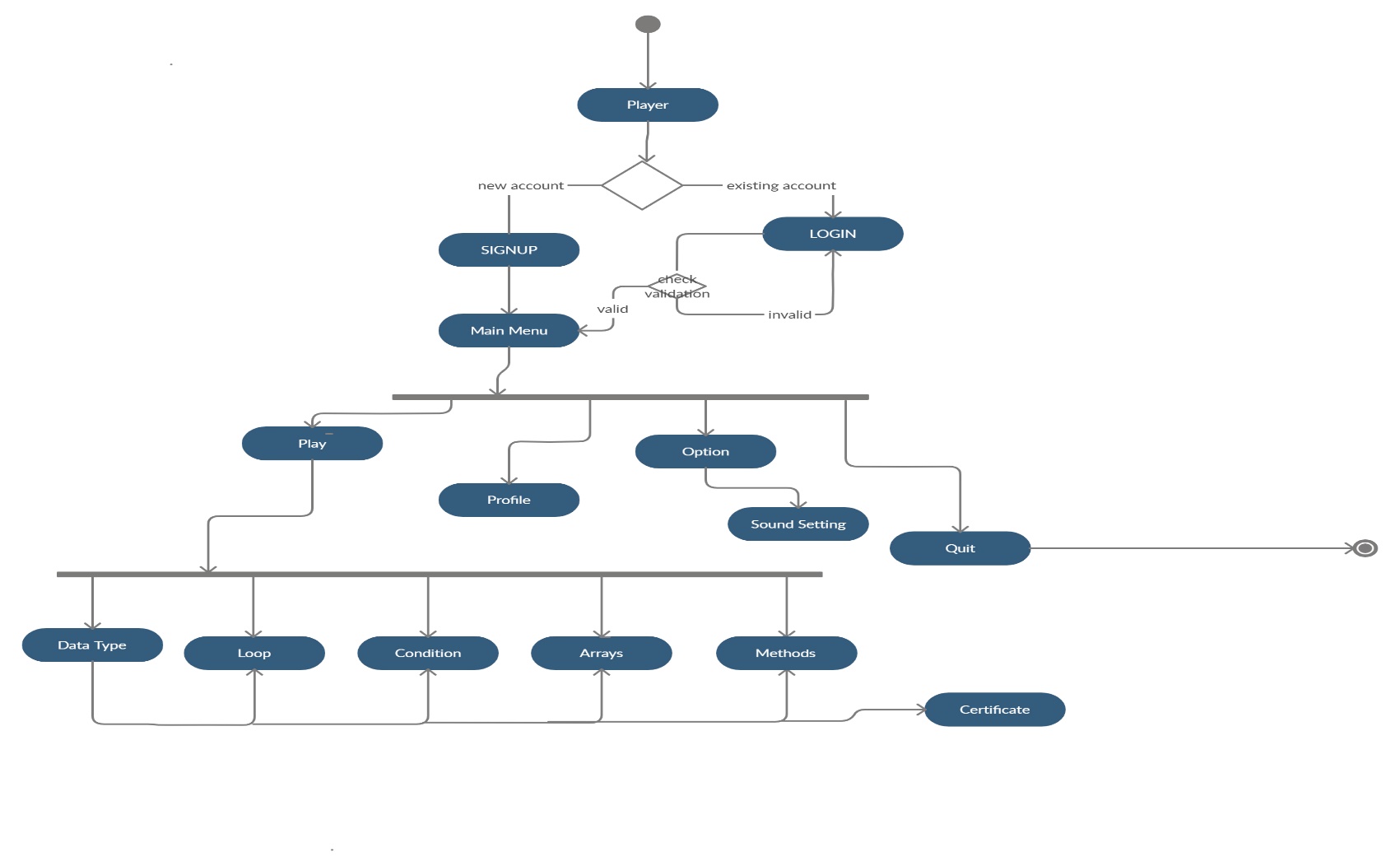
## Level 1 Diagram:

**Figure 4 2 Level 0 Diagram**



**Figure 4 3 Level 1 Diagram**

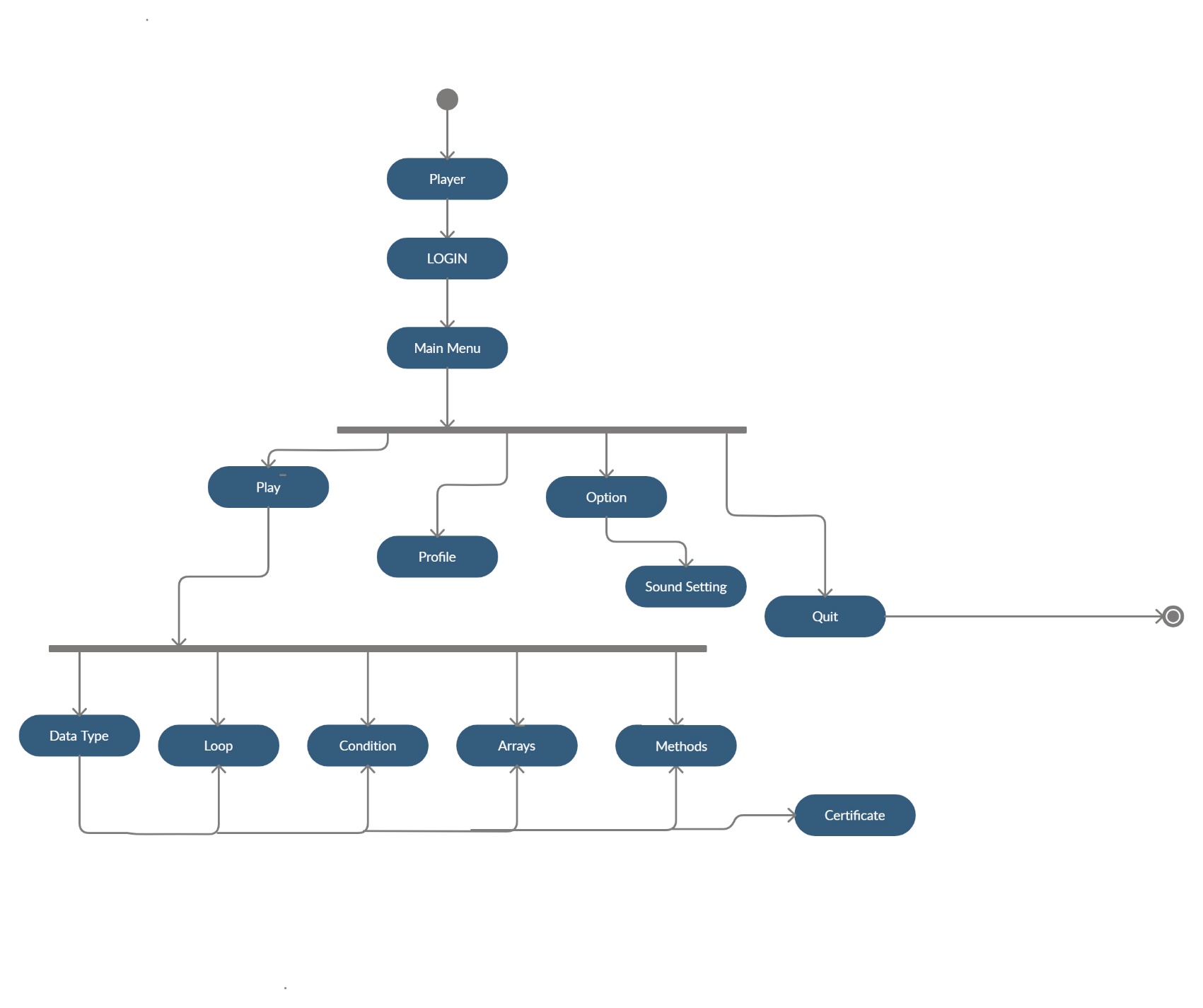
## Activity Diagram:

****

**Figure 4 4 Main Activity Diagram**

The working is started from main activity of the application. The system will check at the start whether the player already has an account or not, if the player already has an account then the application activity will be moved to login, and if the player has not an account then the application activity will be moved to sign-up. When the signing process completes the activity will be moved to the main menu of the application. In main menu activity the has multiple options, the player could play game, set the settings of the game, could also be able to visit his profile section, check his/her current position in the leaderboard, and could also quit the game. If the player chooses the “play game” option, the activity will be moved to level 1, then level 2 and so on up-to level 6.

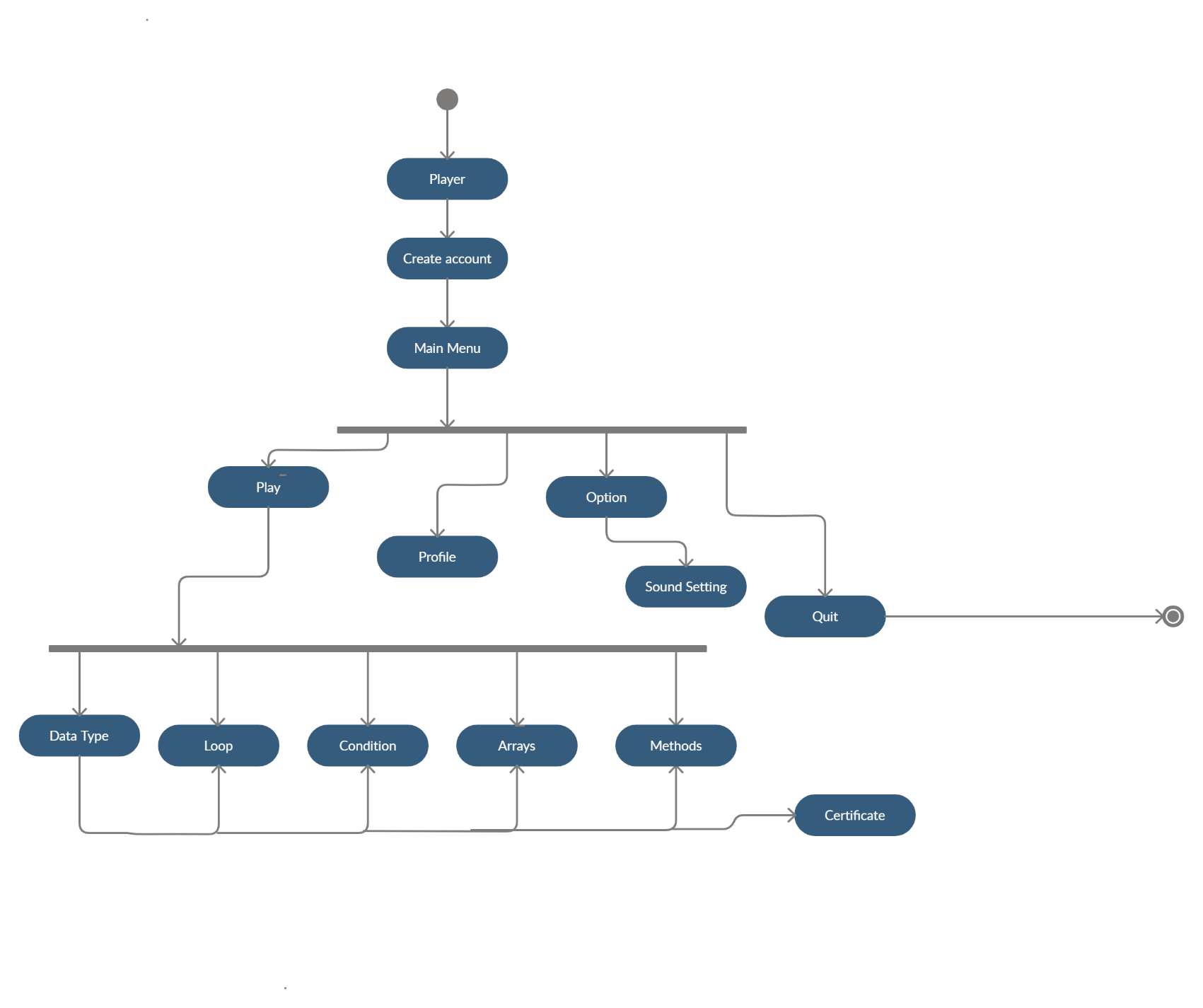
## Login-in Activity Diagram:

****

**Figure 4 5 Sign-in Activity Diagram**

The working is started from main activity of the application. While logging-in the player would be asked to enter his/her email and password. The system will then match the email and password with the database, if the account details are matched then the system will send a verification code, the user receives that code and enter it and the activity will be moved to the main menu.

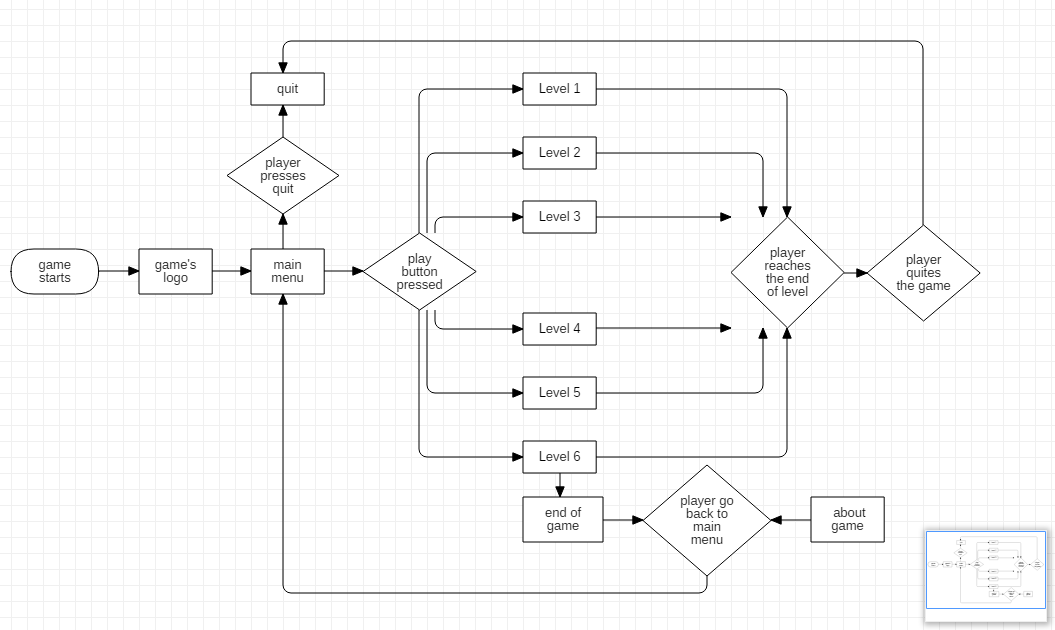
## Sign-up Activity Diagram:

****

**Figure 4 6 Sign-up Activity Diagram**

The working is started from main activity of the application. If the player hasn’t an account then the application activity will be moved to sign-up. The application activity will move to sign-up activity, here the player will fill the form to register himself and after this process the activity will be moved to main menu

## Process Flow/representation:

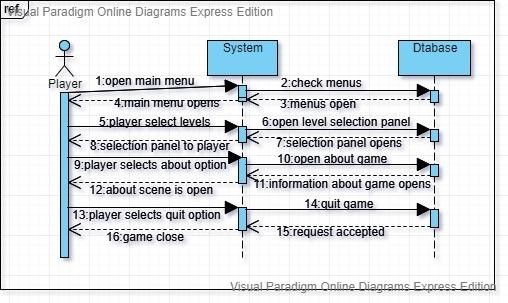


**Figure 4 7 Process Flow/representation**

## Design Models: Sequence Diagrams:

A sequence diagram simply represents interplay between items in a sequential order i.e. the order in which these interactions take region. Sequence diagrams define how and in what order the gadgets in a machine characteristic.

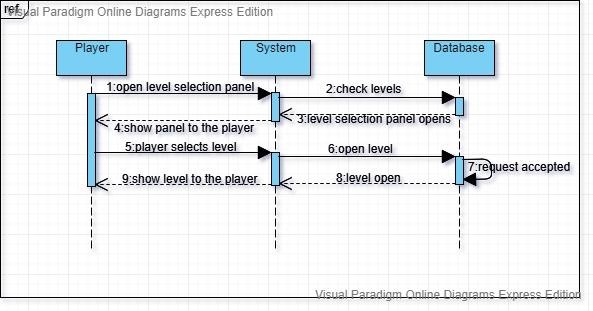
### Main Menu Sequence Diagram:

* + - Open main menu
    - Player selects level selection panel, about or quit option.
    - Request accepted by system

**Figure 4 8 Main Menu Sequence Diagram**

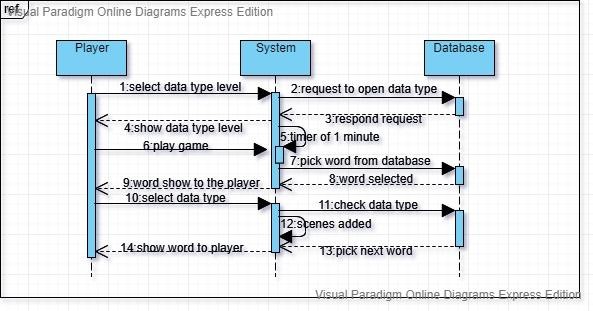
### Selection Panel Sequence Diagram:

* + - Open level selection panel level
    - Player selects level
    - Level open

`

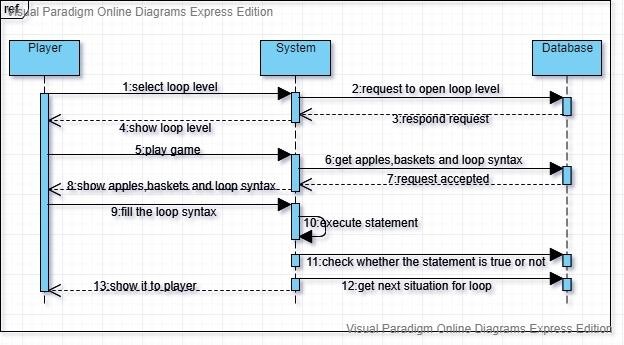
**Figure 4 9 Selection Panel Sequence Diagram**

### Data-type Sequence Diagram:

* + - Select data type level
    - Select data type of given word
    - Scores are added
    - Exit

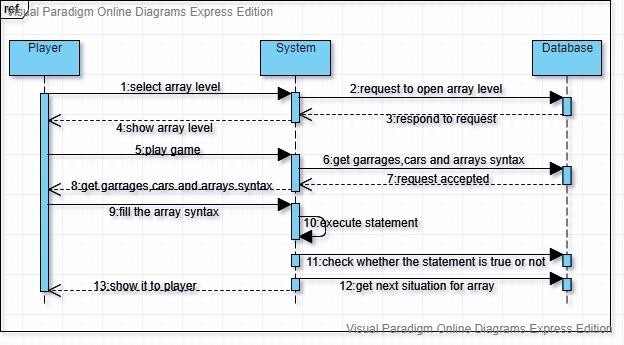
**Figure 4 10 Data-type Sequence Diagram**

### Loops Sequence Diagram:

* + - Select loop level
    - Play game
    - Fill the loop syntax
    - Execute

**Figure 4 11 Loops Sequence Diagram**

### Arrays Sequence Diagram:

* + - Select Array Level
    - Play game
    - Fill the array syntax
    - Execute

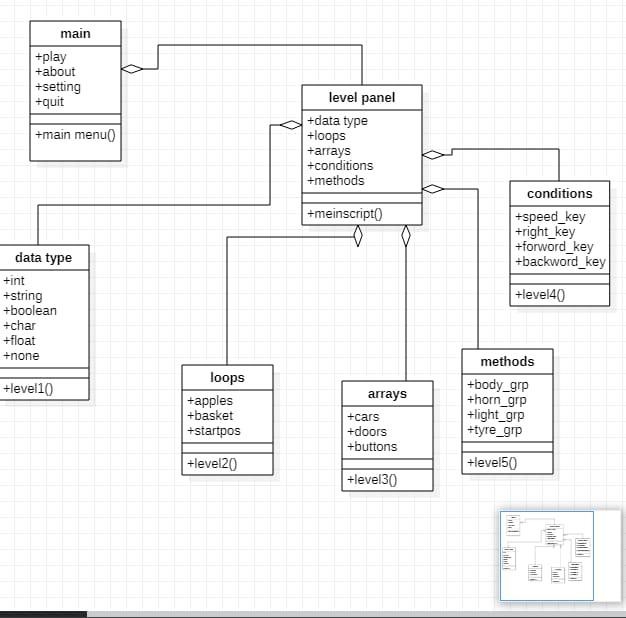
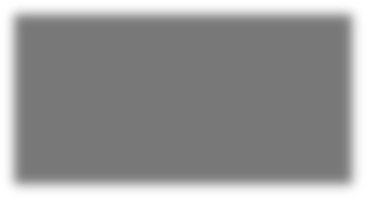
**Figure 4 12 Array Sequence Diagram**

### Conditions Sequence Diagram:

* + - Select Condition Level
    - Fill the conditions
    - Execute
    - Play game
    - Exit

**Figure 4 13 Conditions Sequence Diagram**

## Class Diagram:



**Figure 4 14 General Class Diagram**

**Chapter 5**

# Implementation

## External APIs:

### Firebase:

## Firebase is a Backend-as-a-Service — BaaS — that began as a YC11 start up and grew up into a cutting edge

## Application improvement stage on Google Cloud Platform. ... Firebase liberates designers to Centre making

## Fabulous client encounters. You don't have to oversee servers

## User Interface:

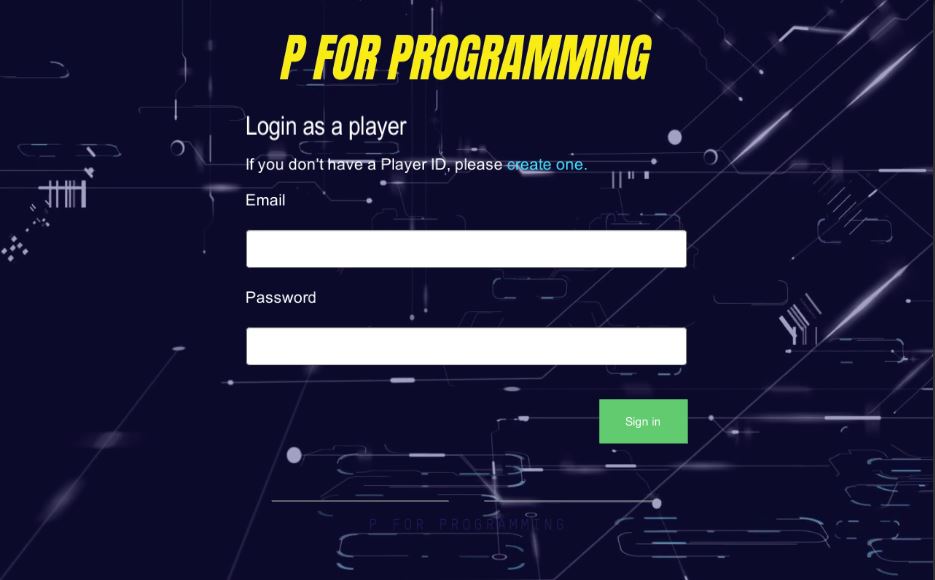
User Interface implies the collaboration of user and machine or we can say human machine connection. In

Game User interface is important to discuss player with game module.

### Login/SignUp:

**Table 5.1 Login/Signup**

|  |  |  |
| --- | --- | --- |
| **Login/Signup** | | |
| **Text Fields** | Username | User will enter a unique username. |
| Email | User should enter a valid Email |
| Password | User needs to enter a password. Password should be 8 Characters long |
| **Buttons** | Login/Register | Check validation of username, Email and password. If user does not exist, register him with same provided credentials. |



**Figure 5 1 Login Interface**

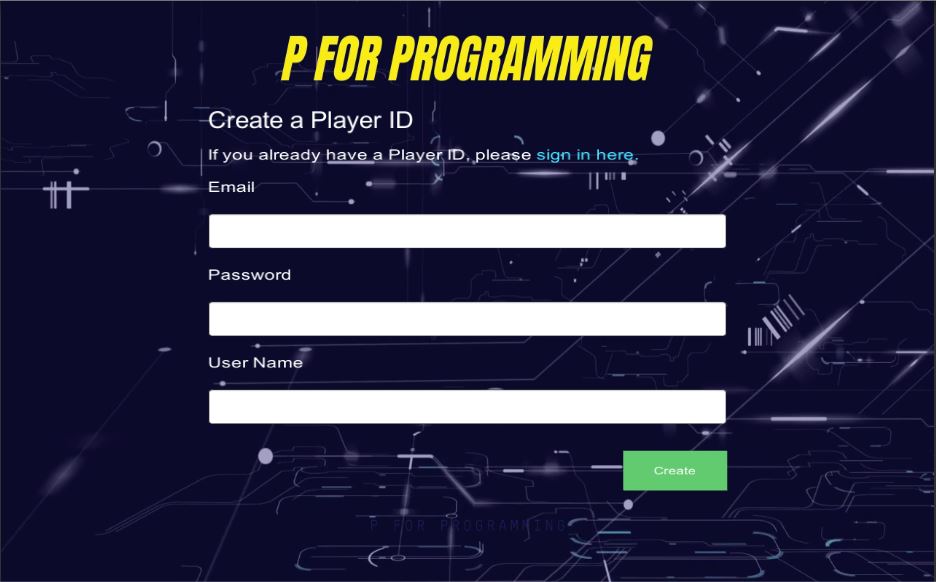


Figure 5 2 Signup Interface

### Main Menu

**Table 5.2 Main Menu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Main Menu** | | | |
| **Buttons** | Play | Play different levels |
| Option | Set the graphic quality of the game. |
| Profile | Display the progress of player |
| Exit | Quit from the game. |
| **Text** | Username | Displays Username |



Figure 5 3 **Main Menu Interface**

### Data Type Level:

**Table 5.3 Data Type**

|  |  |  |
| --- | --- | --- |
| **Data Type** | | |
| **Text** | Score | Display score of the player. |
| Timer | Display timer for data type level. |
| Text | Displays question |
| **Check Boxes** | Check boxes | Displays different options |
| **Button** | Back | Set to display the main menu. |

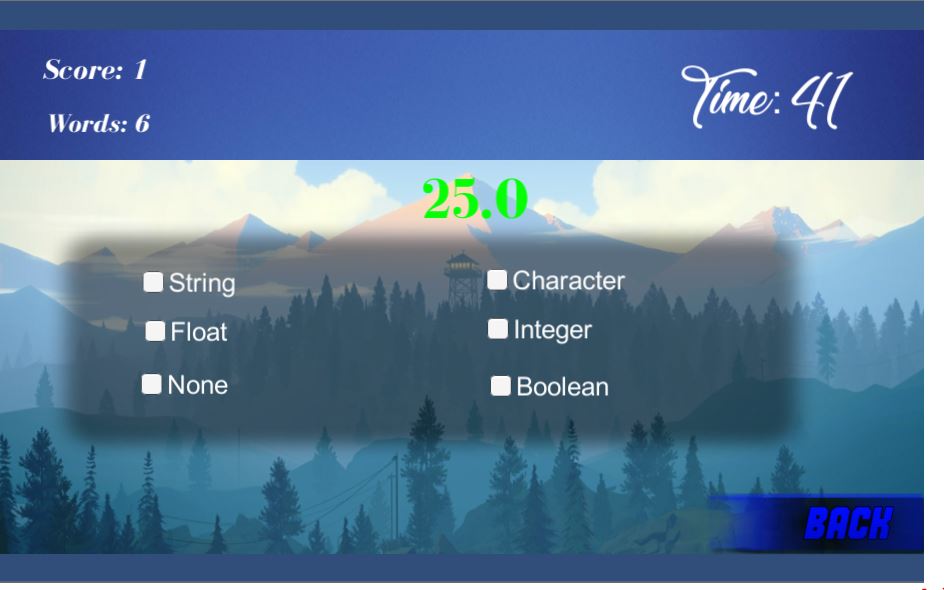


Figure 5 4 **Data Type Interface**

### Loop Level:

**Table 5.4 Loop**

|  |  |  |  |
| --- | --- | --- | --- |
| **Loop** | | | |
| **Buttons** | Next | Play different levels |
| Execute | Set the graphic quality of the game. |
| Back | Exit from the loop level. |
| **Text field** | Initial field | Player will enter initial number |
| Final field | Player will enter final number |

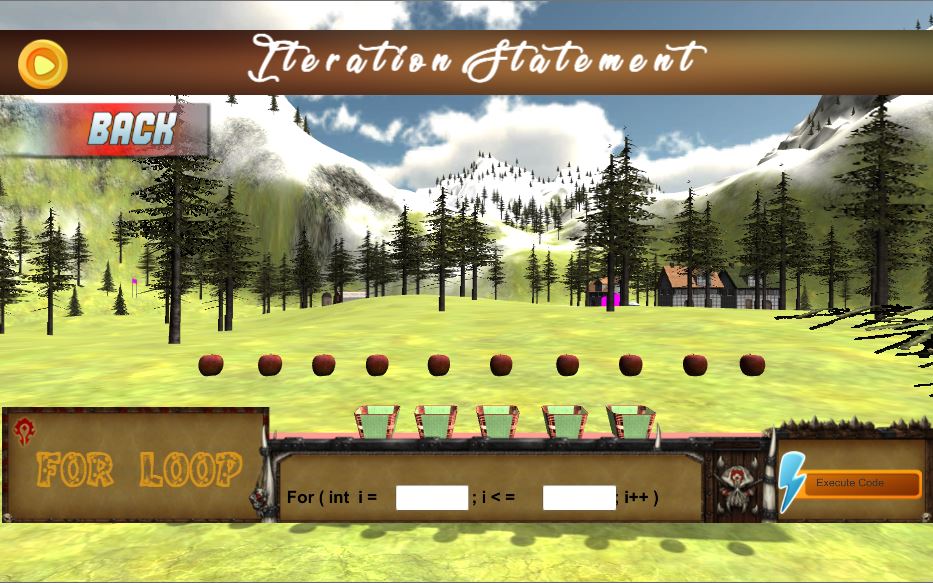
****

Figure 5 5 **Loop Interface**

### Condition Level:

**Table 5.2.1.3 Condition level**

|  |  |  |
| --- | --- | --- |
| **Condition** | | |
| **Text fields** | Text field 1 | Player will input text for right condition |
| Text field 2 | Player will input text for left condition |
| Text field 3 | Player will input text for forward condition |
| Text field 4 | Player will input text for backward condition |
| **Buttons** | Next | Set to display the condition level |
| Back | Set to display the main menu |
| Drive | Execute the conditions. |
| Accelerator | Accelerate the speed of car. |
| Break | Break the speed of car. |
| Left | For android users it will move the car to the left side. |
| Right | For android users it will move the car to the right side. |





Figure 5 6 **Condition Interface**

### Arrays Level:

**Table 5.2.1.3 Arrays**

|  |  |  |
| --- | --- | --- |
| **Arrays** | | |
| **Buttons** | Next | Set to display next scene. |
| Back | Set to display the main menu. |
| Execute array | Execute the command. |
| **Text Field** | Text field | Player will enter number to be executed. |



Figure 5 7 Arrays **Interface**

### Methods Level:

**Table 5.2.1.3 Methods**

|  |  |  |
| --- | --- | --- |
| **Methods** | | |
| **Buttons** | Execute Code | Execute the input text field |
| Body color | Set color for car |
| Light color | Set the color of light |
| Horn sound | Set the sound type for horn |
| back | Set to main menu panel |
| Hide UI | Hide all buttons from interface |
| Add | Display new condition |
| Subtract | Display the previous step |
| **Text field** | Text field | Enable the condition for tyre |

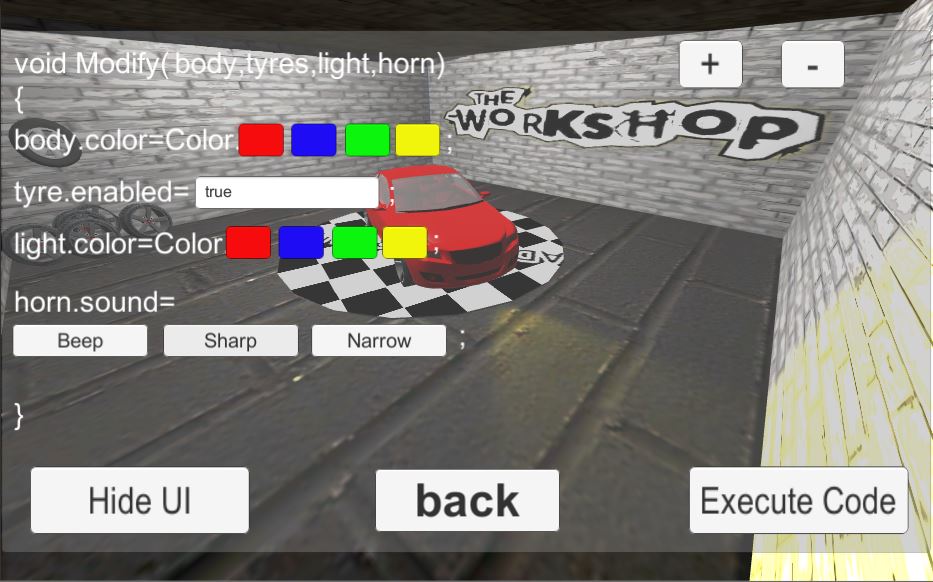


Figure 5 8 Methods **Interface**

**Chapter 6**

# Testing and Evaluation

Testing and EvaluationTesting Evaluation is vital a part of development. During this technique, every functionality of system is

Checked whether it's working consistent with methodology or not.

## **Manual Testing:**

### Unit Testing

**Unit Testing 1:**

#### Testing Objective: To ensure game is working properly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Game Playing | Provide username and password to login | Successfully log in and play all the levels. | Pass |

#### **Unit Testing2:**

#### Testing Objective: To ensure registration form is working properly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Creating new Profile | Provide all the details in registration form | Successfully register a new account for player | Pass |

## Functional Testing:

After unit testing, we tested the multiple functionalities of the game. These functionalities

Include game display, sound setting, graphic setting and flow control of the entire game.  
Functional Testing1: Login with different roles  
**Objective**: To ensure that the player login to their account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Sound setting | Click on sound setting in option menu | Set the volume low or high | Pass |
| 2. | Level 1 Button | Click on data type in level menu | Data type level will be opened | Pass |
| 3. | Quit Button | Click on quit button in main menu | Game will be closed | Pass |

## Integration Testing:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Login as player | Provide email and password in login page | Login successful and the player play all the levels | Pass |
| 2. | Player score record is saving | Click on record button in main menu | Score of the player is shown in record section | Pass |

## **Automated Testing:**

**Tools used:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool Name** | **Tool Description** | **Applied on [list of related test cases / FR / NFR]** | **Results** |
| Alt Unity Tester | It is an open-source UI driven test automation tool that helps you find objects in your game. | Loop scene | Pass |
| Data type scene | Pass |

**Chapter 7**

# Conclusion and Future Work

## Conclusion:

The main objective of our project is to assist the people that want to find out programming but in a simple

Way. We created a friendly environment during which beginners can learn and even have fun. We created

Our application on Unity3d version 2018. This is often an easy PC's/android application, which will be

Employed by people that want to find out the fundamentals concept of programming.

## Future Work:

Our project is providing an easy way to learn the basics of programming by having 5 initial levels. However, in

The future, we can target programming languages too. We will enhance the game and it will be for those who have

Basics concepts Therefore, we will increase the number of levels and the complexity of the game.

**Chapter 8**

# References

**Refrences:**

1. Andre F.S. Barbosa, Pedro N. M. Pereira, Joao A. F. Dias, Frutuoso G. M. Silva. “A new Methodology of Design and Development of Serious Games”, International Journal of Computer Games Technology, 2014
2. [chipapk.com](http://www.chipapk.com/)
3. [www.codeconquest.com](http://www.codeconquest.com/)
4. [www.cargocreative.co.uk](http://www.cargocreative.co.uk/)
5. apps.gs1.org
6. [www.game.radinost.com](http://www.game.radinost.com/)

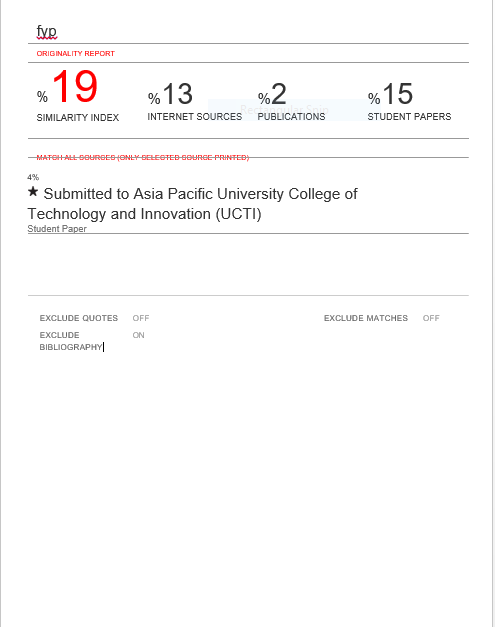
**Plagiarism Report**

Skip to Main Content

fyp

*by* Fyp Fyp

|  |  |  |  |
| --- | --- | --- | --- |
| FILE | PLAGARISM\_DOCUMENT.DOCX (3.96M) |  | |
| TIME SUBMITTED | 25-JUL-2020 01:29PM (UTC+0500) | WORD COUNT | 4621 |
| SUBMISSION ID | 1361909274 | CHARACTER COUNT | 22344 |

****