Software Requirements Specification

for





(Unity 3D game)

Version 1.1

Prepared by

Group Name: CODE BLOODED

|  |  |  |
| --- | --- | --- |
| Tehreem Sultan | SE-011 | tehreemsultan@gmail.com |
| Iqra Farooq Malik | SE-013 | iqramalik200021@gmail.com |
| Rana Muhammad Ibrahim | SE-034 | muhammadibrahim.ibrahim9@gmail.com |
| Muhammad Abdullah Hayat | SE-041 | malik\_muhammad222@outlook.com |

|  |  |
| --- | --- |
| Instructor: | Miss Zainab |
| Date: | 11th November,2019 |

Contents

[**1** **Introduction** 3](#_Toc24338800)

[1.1 Problem Definition 3](#_Toc24338801)

[1.2 Purpose 3](#_Toc24338802)

[1.3 Scope 3](#_Toc24338803)

[1.4 Definitions, Acronyms and Abbreviations 4](#_Toc24338804)

[1.5 References 5](#_Toc24338805)

[1.6 Overview 5](#_Toc24338806)

[**2** **Overall Description** 6](#_Toc24338807)

[2.1 Product Perspective 6](#_Toc24338808)

[2.1.1 User Interface 6](#_Toc24338809)

[2.1.2 **Hardware Interface** 6](#_Toc24338810)

[2.1.3 **Software Interface** 7](#_Toc24338811)

[2.1.4 **Memory Constraints** 7](#_Toc24338812)

[2.2 Product Functions 7](#_Toc24338813)

[2.2.1 **Start** 8](#_Toc24338814)

[**3** **Specific Requirements** 13](#_Toc24338815)

[3.1 Interface Requirements 13](#_Toc24338816)

[3.2 Functional Requirements 13](#_Toc24338817)

[3.3 Non-functional Requirements 13](#_Toc24338818)

[3.3.1 **Performance Requirements** 13](#_Toc24338819)

[3.3.2 **Design Constraints** 13](#_Toc24338820)

[**4** **Data Model and Description** 14](#_Toc24338821)

[4.1 Data Description 14](#_Toc24338822)

[4.1.1 **Data Objects** 14](#_Toc24338823)

[4.1.2 **Data Dictionary** 15](#_Toc24338824)

[**5** **Behavioral Model and Decription** 16](#_Toc24338825)

[5.1 Description for Software Behavior 16](#_Toc24338826)

[5.2 State Transition Diagram 16](#_Toc24338827)

[**6** **Planning** 17](#_Toc24338828)

[6.1 Team Structure 17](#_Toc24338829)

[6.2 Estimation 17](#_Toc24338830)

[6.3 Process Model 17](#_Toc24338831)

[**7** **Conclusion** 17](#_Toc24338832)

[**8** **Supporting Information** 18](#_Toc24338833)

# **Introduction**

This software requirement specification (SRS) report includes description of Run Boy Run-3D Game Project, by team “CODE BLOODED” of sophomore students of Software Engineering Department in NED University of Engineering and Technology, Karachi, Pakistan. All the functionalities and specifications about the project will be explained in detail.

## Problem Definition

There are some games in markets that have the theme of obstacle avoiding but they are boring, simple and unsatisfying from the point of view of users. So, in order to create much more entertaining and unique game we have chosen our project.

Moreover, the target problem that this project objects to tackle is to make contribution to Software Engineering Departments creative image and to take it out form conventionality of projects like managements systems and booking websites. We will design an application that is a creative and addictive game which is multi-level and single-player. It includes the 3D model of a character that runs on a scenic platform. In the game the user can select a player of his/her own choice and then play the game. During the game the user has to collect coins and boosters, he/she has to avoid the obstacles or any poisons placed in the path. When the user completes the path, it will be transferred to the next level or, if all the levels are complete, then to the end screen to end the game. The aim of this gaming application is to help people enjoy their leisure time in an easy-going and entertaining game.

## Purpose

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

The intended audience of this document is the members of the project group and developers who are willing to implement the application explained in this document. The document will guide the developers through the implementation phase. In addition to this, the document also help the developers to see early misunderstandings, inconsistencies and possible defects of the system.

## Scope

The project name is RUN BOY RUN – 3D Game. “RUN BOY RUN” is so because the music in the game is a song named ‘Run Boy Run’ so this is a perfect match. “3D” represents the 3D model of the characters and environment which will be used.

The software will make the user to be able to play a obstacle avoiding, addictive game. While playing the game, the user runs on a path and collects boosters and coins while its score is continuously increasing was its runs forward. The end-product will run on 64-bit windows PC for the people who wants to access the product through desktop and laptop computers.

The software will include one basic operations. This operation is running. The user will be able to change its directions or lanes by the arrow and direction keys as well as alternate keys like ‘a’ (left), ‘d’ (right) and ‘w’ (jump).

## Definitions, Acronyms and Abbreviations

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Term** | |  | |  | **Description** | |
|  | **User** | | Person who can play the single player or multiplayer mode of  the game. | | | |  |
|  |  | |  |
|  | |  | |  |  | |  |
| **SRS** | |  | |  | Software Requirements Specification. | | |
| **Class Diagram** | | |  | | A type of static structure diagram in UML that describes the  structure of a system by showing the system’s classes, their  attributes, operations (or methods), and the relationship among  the classes. | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  | |  |  | |  |
| **Use Case Diagram** | |  | |  | A type of diagram in UML that represents the user’s interaction | | |
|  |  | |  | |  | with the system. | |
|  | |  | |  |  | | |
|  | **Sequence Diagram** | |  | |  | An interaction diagram that shows how processes operate with |  |
|  |  | |  | |  | one another and in which order. |  |
|  | |  | |  |  | |  |
| **Unity 3D4** | |  | |  | A cross-platform game creation system developed by Unity | | |
|  |  | |  | |  | Technologies, including a game engine and integrated | |
|  |  | |  | |  | development environment. | |
|  | |  | |  |  | | |
|  | **Game Engine** | |  | |  | Software framework designed for the creation and development |  |
|  |  | |  | |  | of video games. |  |
| **Integrated** | |  | |  | Software application that provides some good and useful | | |
| **Development** | |  | |  | facilities to programmers for software development. | | |
| **Environment** | |  | |  |  | |  |
|  | |  | |  |  | | |
|  | **IEEE** | |  | |  | The Institute of Electrical and Electronics Engineers |  |
| **FPS** | |  | |  | Frame per second | | |
|  | **Scene** | |  | |  | In unity every different screen is represented as scene. |  |
| **Windows2** | |  | |  | A group of several graphical operating system families, all of which are developed, marketed and sold by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It powers desktop computers and laptops. | | |
| **Blender3** | |  | |  | A free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, and computer games. | | |
| **Visual Studio5** | |  | |  | An [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program), as well as [websites](https://en.wikipedia.org/wiki/Web_site) , [web apps](https://en.wikipedia.org/wiki/Web_app), [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). | | |

## References

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

1. IEEE. (1998). IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society.
2. Retrieved November 9, 2019, from

<https://en.wikipedia.org/wiki/Microsoft_Windows>

1. Retrieved November 9, 2019, from <https://en.wikipedia.org/wiki/Blender_(software)>
2. Retrieved November 9, 2019, from

<http://unity3d.com/unity>

1. Retrieved November 9, 2019, from

<https://en.wikipedia.org/wiki/Microsoft_Visual_Studio>

## Overview

This SRS document include 7 more parts from this point.

Section 2 is about overall description of the project. This section includes 2 subsections.

Section 3 is about modules and interfaces. This section includes 3 subsections.

Section 4 is about data models and descriptions. This section includes 1 subsection.

Section 5 is about behavioral design which are mentioned in 2 subsections.

Section 6 is about planning the project with specifying team structure, basic schedule of project and process model.

Section 7 is the conclusion part of SRS.

Section 8 includes appendixes.

# **Overall Description**

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

## Product Perspective

Run Boy Run-3D gaming application is totally independent system that is not related to any other system and not a component of a larger system. This program has only one type user, so there is no functionality differences between users which means there exists only one type user interface. At the beginning of the program there will be 5 options; firstly START to enter the game, secondly SELECT CHARACTER, thirdly HELP to read instructions on how to play, and how they developed the game, fourthly CREDITS to read about the team CODE BLOODED and lastly EXIT to quit the game. When the user presses will select a character to play with and then starts the game. In the beginning of start, the user is told to press “Spacebar” to start the game. After pressing “Spacebar” the user automatically starts to move in forward direction with constant speed. The user can move any direction he/she wants with direction keys.

In terms of hardware, Run Boy Run-3D will be compatible with computers be it laptop or desktop.

In terms of software, Run Boy Run-3D will run on windows operating system. Moreover, it will be implemented making use of Unity3D for visualizing its features.

This brief information of interfaces is explained in more detailed below.

### User Interface

There will be one type of user. Therefore there are no differences between users in terms of functionality, visualization and interface.

At the beginning of the program there will be 5 options;

* START - to enter the game
* SELECT CHARACTER – To select the character to play the game with
* HELP - to read instructions on how to play
* CREDITS - to read about the team Code Blooded and how they developed the game
* EXIT - to quit the game

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

1. When “up arrow key” or “W” button is activated, the character JUMPS upward.
2. When “rightward key” or “D” button is activated, the character moves in RIGHT direction.
3. When “leftward key” or “A” button is activated, the character moves in LEFT direction.

By moving with the help of these keys, the user can collect the coins scattered all over the path. Moreover, the user can also collect boosters in same manner as the coins.

### **Hardware Interface**

Only computers, be it desktop or laptop, which have keyboard to use arrow keys or WAD keys will be suitable for the application. These devices should have some limit requirements to make the application run effectively. We expect 1GHz processor, 100MB minimum internal memory & 2GB RAM for computers.

### **Software Interface**

Computers will be used for the application and they must have Windows to run the application. Moreover, application will be coded by C#. The Characters will be made using Blender Software. To make the actual game, Unity3D will be used. For Scripting, Visual Studio will be used.

### **Memory Constraints**

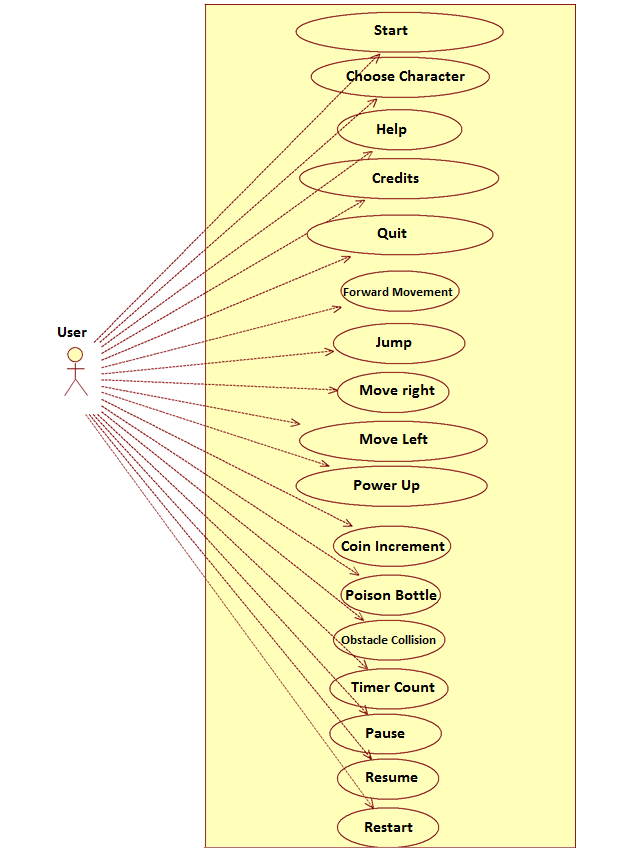
Since we will be using Unity as a game engine, Unity will pack lots of memory for its own. The computer’s minimum available RAM should be 2GB and minimum internal memory 100 MB.

## Product Functions

Use case diagram of the Run Boy Run-3D application is revealed in Figure 1. Steps are gathered in distinct entities, the functions of which are stated in further subsections.

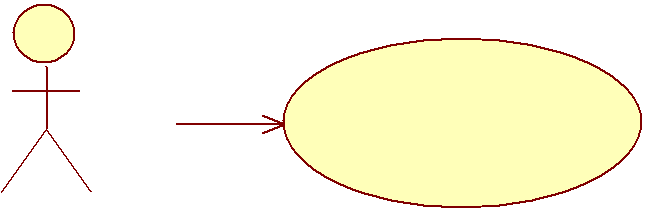
**Figure 1- Use Case Diagram**

**Figure 2-Use case Diagram**



### **Start**

*Diagram*:



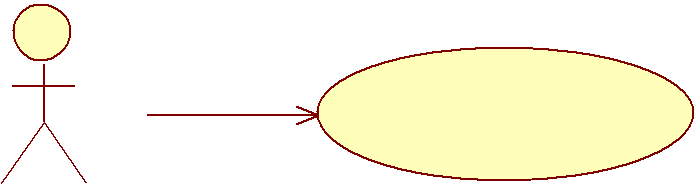
**Start**

*Brief Description:*

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

#### Choose Character

*Diagram*:



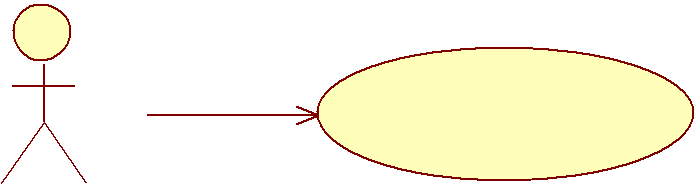
**Choose Character**

*Brief Description:*

This functionality becomes activated when the user presses “CHOOSE CHARACTER” button on the Main menu. This function makes the user to choose between different characters to play with.

#### Help

*Diagram*:



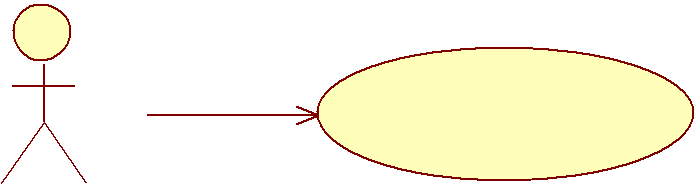
**Help**

*Brief Description:*

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

#### Credits

*Diagram*:



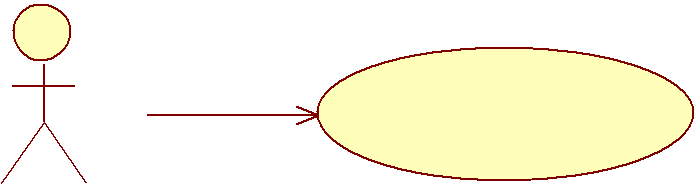
**Credits**

*Brief Description:*

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

#### Quit

*Diagram*:



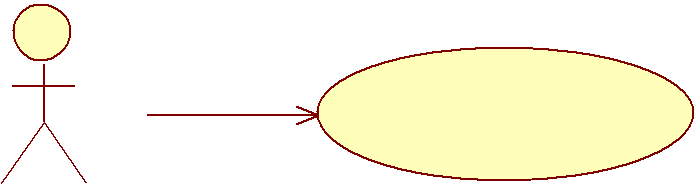
**Quit**

*Brief Description:*

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

#### Forward Movement

*Diagram*:



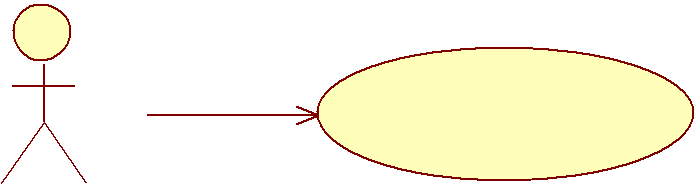
**Forward Movement**

*Brief Description:*

This functionality becomes activated when the user presses “START” button on the Main menu and then presses SPACEBAR key on the keyboard. This function enables the forward movement of the character and thus starts the run.

#### Jump

*Diagram*:



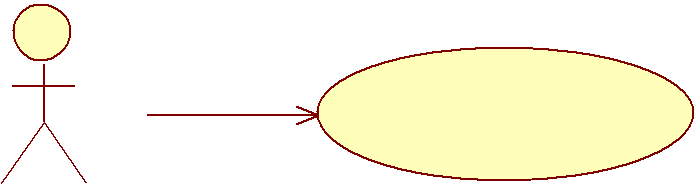
**Jump**

*Brief Description:*

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

#### Move Right

*Diagram*:



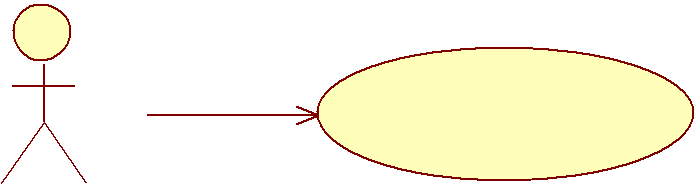
**Move Right**

*Brief Description:*

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

#### Move Left

*Diagram*:



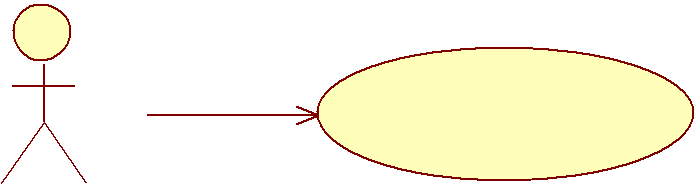
**Move Left**

*Brief Description:*

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

#### Power Up

*Diagram*:



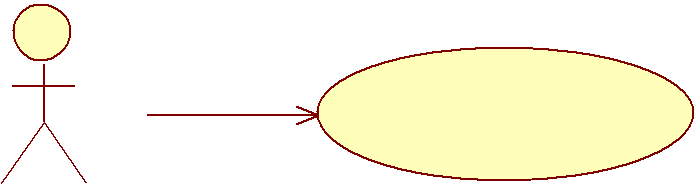
**Score Boost**

*Brief Description:*

This functionality becomes activated when the character touches a SCORE BOOSTING DIAMOND. This function makes the score get incremented by 1000 points.

#### Coin Increment

*Diagram*:



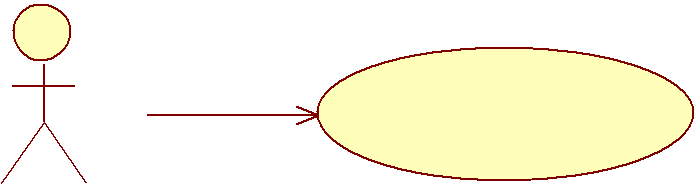
**Coin Increment**

*Brief Description:*

This functionality becomes activated when the character touches a COIN. This function makes the coin count increment by 1.

#### Poison Bottle

*Diagram*:



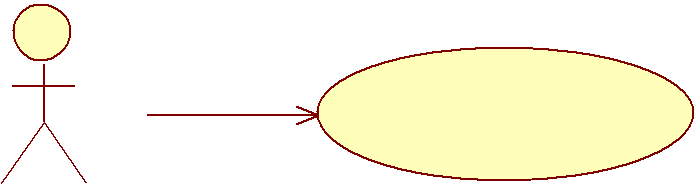
**Poison Bottle**

*Brief Description:*

This functionality becomes activated when the character touches a POISON BOTTLE. This function makes the game over for the user and thus it dies and the level restarts.

#### Obstacle Collision

*Diagram*:



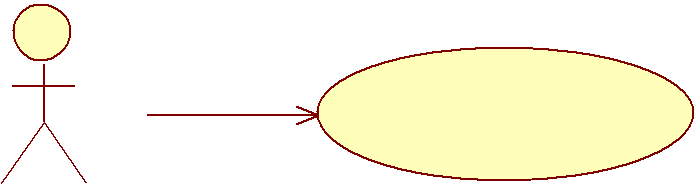
**Obstacle Collision**

*Brief Description:*

This functionality becomes activated when the character touches an OBSTACLE (Crates, Moving Crates and Stone). This function makes the game over for the user and the user dies thus the level restarts.

#### Timer Count

*Diagram*:



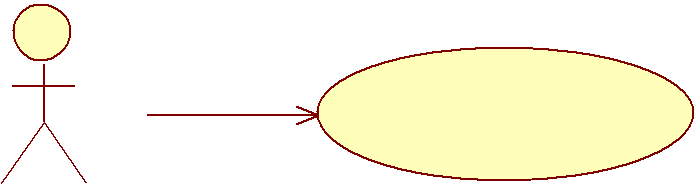
**Timer Count**

*Brief Description:*

This functionality becomes activated when the character starts running after pressing SPACEBAR. This function counts the time that has passed since the user started running and displays it on the screen.

#### Pause

*Diagram*:

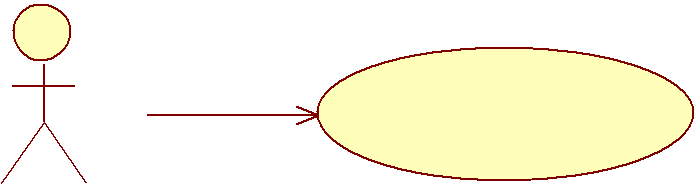


**Pause**

*Brief Description:*

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

#### Resume

*Diagram*:

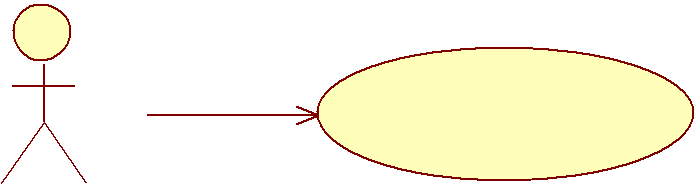
**Resume**

*Brief Description:*

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

#### Restart

*Diagram*:



**Restart**

*Brief Description:*

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

# **Specific Requirements**

The detailed user interface description is given in Section 2.1.1.

This section describes the details of software requirement in 3 subsection:

Interface Requirements, Functional Requirements and non-Functional requirements.

## Interface Requirements

There is only one user, and the user interface is shown in the figure which indicates that user can reach the only from the game icon on windows.

**USER**

**WINDOWS**

## Functional Requirements

Section 2.2 has a detailed information about the functional requirements by explaining use cases in detailed manner, so there won’t be any detailed information in here.

## Non-functional Requirements

### **Performance Requirements**

Since there will be only one user playing the game in his/her local environment, the system does not need to handle multiple user case.

In single player mode, the game should work at least 30 FPS, so that the game will be fluent.

Loading phase of the game should not take more than 40 seconds; in other words the game should start in 40 seconds after the user opens it -this requirement is about the game itself.

The game can be run at any computer which has game setup, keyboard and windows.

#### A player who has not played the game before should learn all of its functionalities in 20 minutes.

System does not require high performance.

### **Design Constraints**

This section has five subsections which are security, usability, availability, maintainability, respectively

#### .Security

Since the game does not have any critical data, no security level need to be introduced.

#### Usability

The scope of the game is widespread. People from every age should play the game without any effort.

#### Availability

The game will be available after installation from set-up.

#### Maintainability

The system should be maintainable in order to add new features to the newer versions. Also, the game should work on newer versions of the related operation systems (the game should be updated then if necessary). Hence, all design aspects should be well documented and easily understandable.

#### Portability

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

# **Data Model and Description**

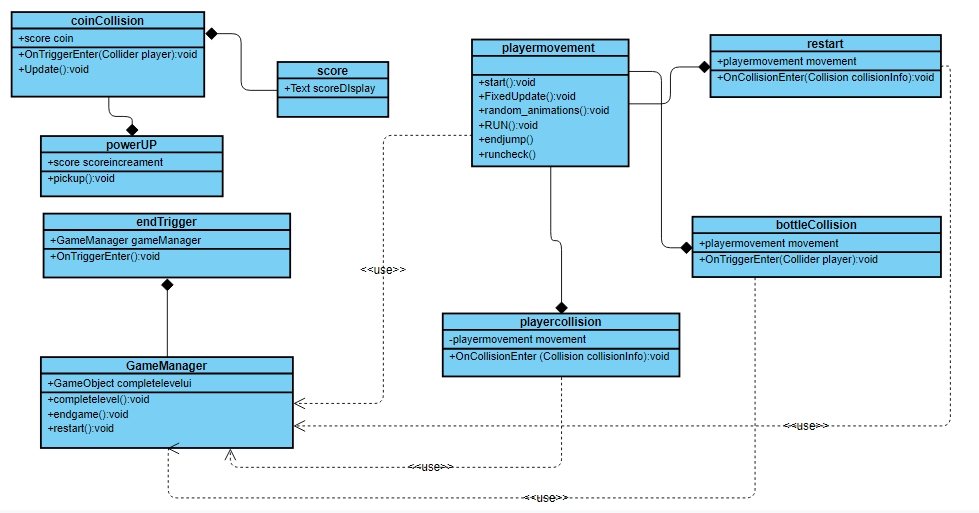
## Data Description

### **Data Objects**

UML Class Diagrams will be presented in this section. The purpose is to introduce the objects that the system is required to have a representation for. Their explanation is given below.

1. CoinCollision: This class deals with the collision of player with coins and increment score.
2. PowerUp: This class deals with the collision of player with the diamond and increment score by 1000.
3. Score: this class displays the score on screen.
4. Playermovement: this class deals with all the basic movement of player including jump, left, right, start movement.
5. Restart: this class restart the game after the player dies.
6. Endtrigger: This class deals when the player reaches the end of level, it starts new level.
7. GameManager: This class deals with all the main functions of level complete, endgame, and restarting level.
8. Playercollision: This class deals with the collision of player with all obstacles.
9. BottleCollision: This class deals with the Collison of player with the bottle resulting in dying and game end.

UML diagram of all above mentioned classes.



### **Data Dictionary**

All definitions are explained in section 4.1.1.

# **Behavioral Model and Decription**

This section presents a description of the behavior of the software.

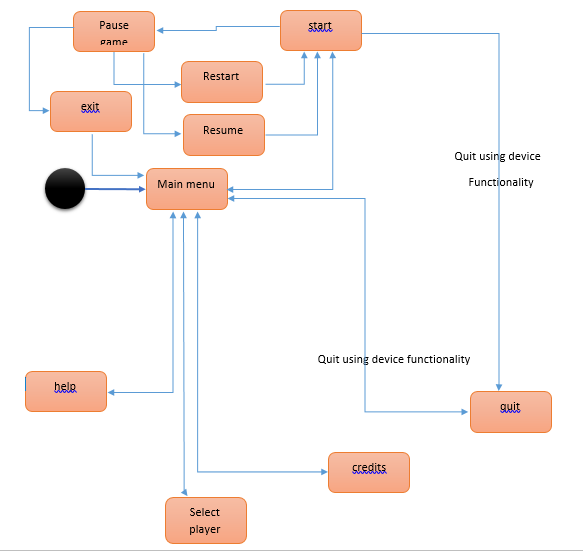
## Description for Software Behavior

When the game starts with a single click on game icon the user will see main menu page. On the main menu page user will see five options.

1. **Start**
2. **Select character**
3. **Help**
4. **Credits**
5. **Quit**

After clicking the space bar button character will start running the user will be able to move the character left and right to dodge the obstacles with the help of right, left arrow keys or with A-D keys. User can also make the character jump through up arrow key or W key and can also make the character collect coins and diamonds to increase the score. If the character will be able to reach to the end point of the first level second level will be started automatically and the game will keep going until the character hit an obstacle. When the game will end user will be able to see final score and number of coins collected. The user can quit anywhere in the game by clicking on ESC key and can also pause game by clicking on the pause button.

## State Transition Diagram



# **Planning**

In this part of the document, the structure of the team responsible for the project, the basic schedule, and the process model will be presented.

## Team Structure

The project team is group of 4 people, and each group member has the same responsibility on the project. Iqra Farooq Malik is the coordinator and the contact person for the group, who synchronizes the meetings and work plan since the team is a small group of people and the members know each other, all the decisions are given after meetings and discussions. The labor division is made evenly and decided in weekly meetings. Beside these meetings the members are in contact with each other all the time which will ease the project development and handling faced problems. Moreover, group has contact with the advisor assistant, by having meetings every week. All the members of the group has the same level of knowledge about the project. Therefore, to get the necessary knowledge sooner, the research areas about the requirement of the project are divided into 4 different parts and assigned to each member, and the useful information is shared with the other members

## Estimation

|  |  |
| --- | --- |
| **Estimation Date** | **Task** |
| 20.11.2019 | Detailed requirements |
| 25.11.2019 | Design(modeling of basic things) |
| 5.12.2019 | implementation |
| 15.12.2019 | Testing |
| 17.12.2019 | Deployment |
| 20.12.2019 | Maintenance |

## Process Model

We are going to use the **Waterfall model** in the development phase of the project. By following this model the software development activity is divided into different phases and each phase consists of a series of tasks and has different objectives. Development of one phase will starts only when the previous phase is complete.

# **Conclusion**

This Software Requirement Specification document is prepared to give requirement details of the project “Run Boy Run”. The detailed functional and nonfunctional requirements, system, user, software and hardware interfaces, data and behavioral model are stated in an extended outline. This document will be helpful at constituting a basis for design and development of the system to be developed.

# **Appendixes**

D

[Data Description 14](#_Toc24338822)

[Data Dictionary 15](#_Toc24338824)

[Data Objects 14](#_Toc24338823)

[Definitions, Acronyms and Abbreviations 4](#_Toc24338804)

[Description for Software Behavior 16](#_Toc24338826)

[Design Constraints 13](#_Toc24338820)

E

[Estimation 17](#_Toc24338830)

F

[Functional Requirements 13](#_Toc24338817)

H

[**Hardware Interface** 6](#_Toc24338810)

I

[Interface Requirements 13](#_Toc24338816)

N

[Non-functional Requirements 13](#_Toc24338818)

O

[Non-functional Requirements 13](#_Toc24338818)

P

[Performance Requirements 13](#_Toc24338819)

[Problem Definition 3](#_Toc24338801)

[Process Model 17](#_Toc24338831)

[Product Functions 7](#_Toc24338813)

[Product Perspective 6](#_Toc24338808)

[Purpose 3](#_Toc24338802)

R

[References 5](#_Toc24338805)

S

[Scope 3](#_Toc24338803)

[Software Interface 7](#_Toc24338811)

[Start ... 8](#_Toc24338814)

[State Transition Diagram 16](#_Toc24338827)

T

[Team Structure 17](#_Toc24338829)

U

[User Interface 6](#_Toc24338809)