Software Requirements Specification

for

RUN BOY RUN

(Unity 3D game)

Version 1.0

Prepared by

Tehreem Sultan	**SE-011**	tehreemsultanali@gmail.com
Iqra Farooq Malik	**SE-013**	iqramalik200021@gmail.com

Instructor: Miss Shehnila Zardari

Date: 28th August,2020

ROLES

Tehreem Sultan (SE-011) and **Iqra Farooq Malik (SE-013)** gathered all the requirements for their project “RUN BOY RUN” which is a 3D game. To gather the requirements, they observed the gaming trends in the young generation like what type of games gain popularity and what should be the difficulty level of the game. Tehreem Sultan presented the idea to associate a theme to the game and implemented the background scenery as well as the particle effects for boosters and coins in the game. Iqra Farooq Malik suggested and implemented the logic for the selection of different characters for the player and also the musical theme of the game. Both the members worked together on the animations of the game. Tehreem Sultan proposed and organized the performance requirements, external interfaces and design constraints. Iqra Farooq Malik proposed the scope and perspective of the product along with all the kinds of interfaces, assumptions and dependencies of the system and the software system quality attributes. Tehreem sultan made the class diagram to describe the logical structure of the data while Iqra Farooq Malik made the block diagram and the use-case diagram.

Contents

1 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.

1.1 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.

1.2 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. As the game is addictive, the users especially of young age will definitely like the product and thus the product has a very wide scope.

1.3 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.
Unity 3D[4]	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 Development Environment	Software application that provides some good and useful facilities to programmers for software development.
IEEE	The Institute of Electrical and Electronics Engineers
FPS	Frame per second
Scene	In unity every different screen is represented as scene
Windows[2]	A group of several graphical operating system families, all of which are developed, marketed and sold by Microsoft. It powers desktop computers and laptops.

Blender[3]	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 Studio[5]	An integrated development environment (IDE) from Microsoft. It is to develop computer programs, as well as websites , web apps, web services and mobile apps.

1.4 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 August 22, 2020, from https://en.wikipedia.org/wiki/Microsoft_Windows
3. Retrieved August 22, 2020, from https://en.wikipedia.org/wiki/Blender_(software)
4. Retrieved August 22, 2020, from http://unity3d.com/unity
5. Retrieved August 22, 2020, from https://en.wikipedia.org/wiki/Microsoft_Visual_Studio

1.5 Overview

This SRS document include 3 more parts from this point.
Section 2 is about overall description of the project. It includes detail interface requirements, use cases description, user characteristics as well as assumption and dependencies.
Section 3 includes functional requirements in sub-heading functions, non-functional requirements in sub-headings performance requirements, design constraints and software system quality attributes it also contains class diagram to show logical structure of the data and their attributes.
Section 4 includes appendixes.

2 Overall description

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

2.1 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. Run Boy Run-3D will run on windows operating system. Moreover, it will be implemented making use of Unity3D for visualizing its features. The game seeks advantage of some built-in libraries of unity engine to include some functionalities like the scene management system and the particle management system inside the unity engine. The game logic also uses the animation system of the unity engine to put into play all the animations of the game. The game logic interacts mainly with sub-systems and other libraries inside the unity engine but no external dependencies are on the game. In terms of hardware, Run Boy Run-3D will be compatible with computers be it laptop or desktop but the system should have windows operating system and there are not many dependencies form the hardware perspective as well.

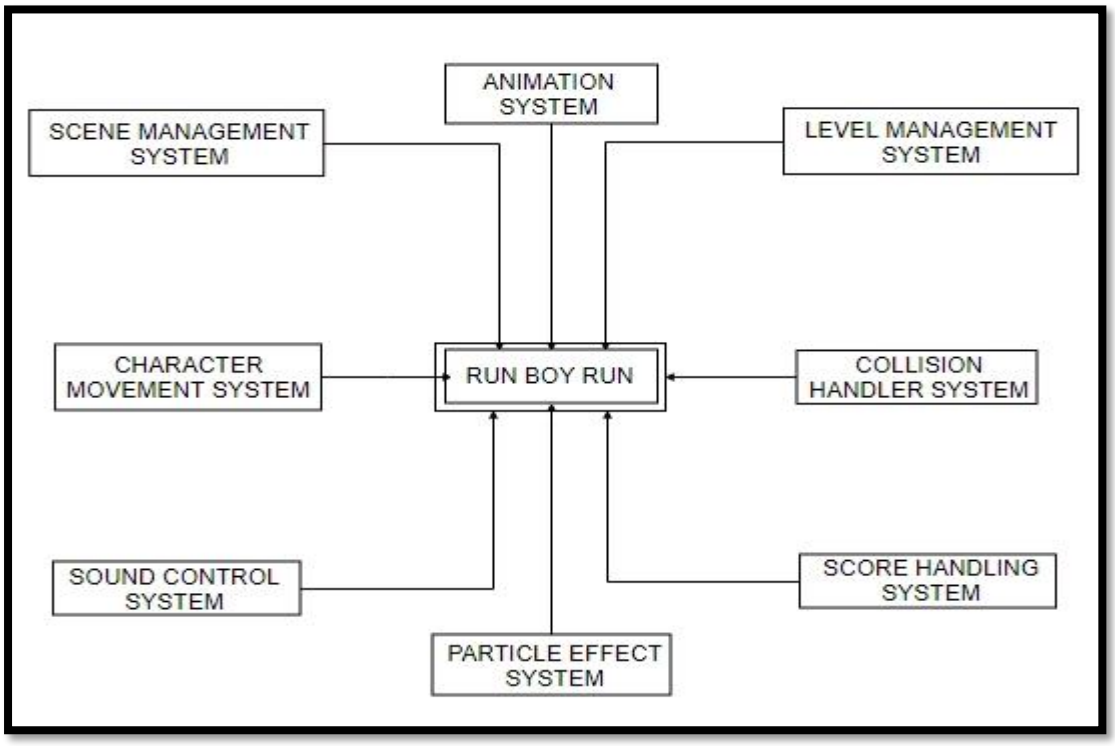


Figure 1- Block Diagram

The above diagram describes the context of the system, the other systems with which the system interacts.

2.1.1 System Interface

One system Requirement is an operating system which should be windows preferably 64-bit (but also works fine on 32-bit). The user must have a keyboard to control the characters in the game and for other navigation in the GUI of the game.

2.1.2 User Interface

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

- When “up arrow key” or “W” button is activated, the character JUMPS upward.
- When “rightward key” or “D” button is activated, the character moves in RIGHT direction.
- 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.

2.1.3 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 & 4GB RAM for computers. For external server if found appropriate Parse servers will be used.

2.1.4 Software Interface

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

2.1.5 Communication Interface

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

2.1.6 Memory Constraints

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.

2.1.7 Operations

There are not many operational constraints. Most of the operations are visible to user if he/she navigates through the graphical user interface. Regarding the operations, user can find help menus also. After the user and system interactions, some operations will take place at the background implemented through coding, such as the score and coin increments after player collects each coin or boosters.

2.1.8 Site Adaptation Requirements

There won't be any site adaption requirements because the game is a desktop application and also no communication over the internet is required.

2.2 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.

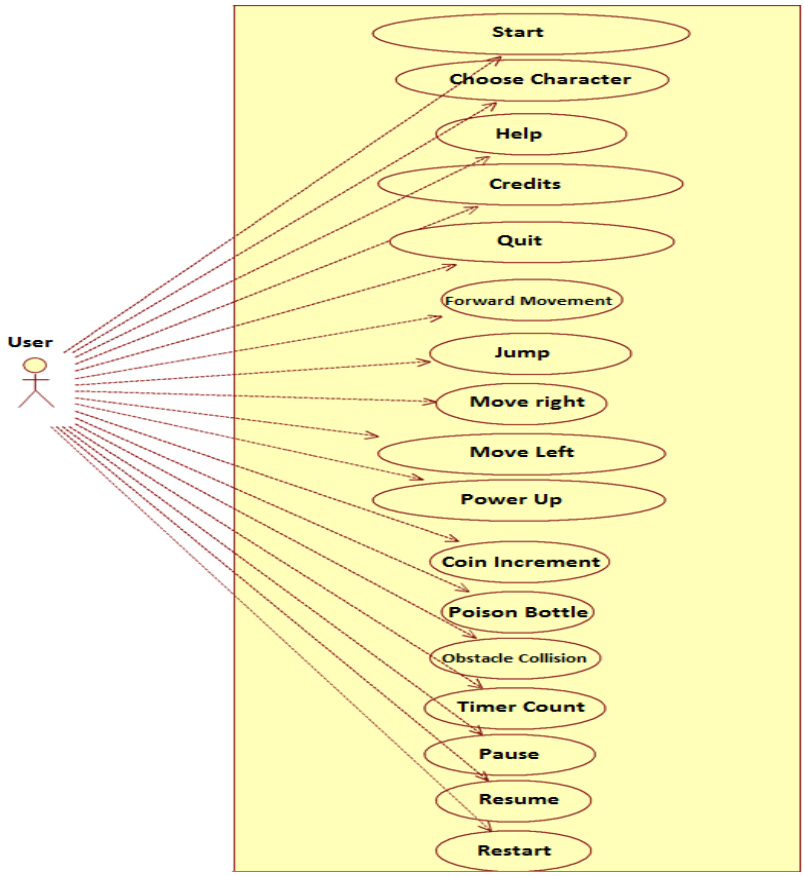
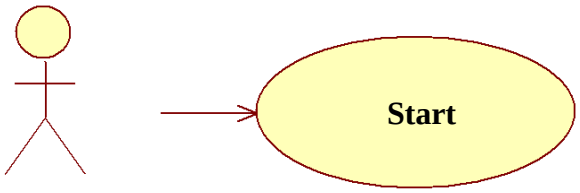


Figure 2- Use Case Diagram

2.2.1.1 *Start*

Diagram:

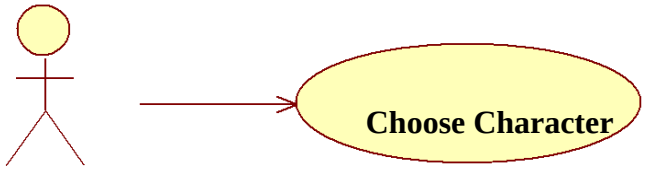


Brief Description:

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.

2.2.1.2 *Choose Character*

Diagram:

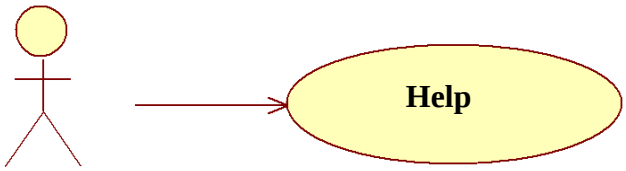


Brief Description:

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.

2.2.1.3 *Help*

Diagram:

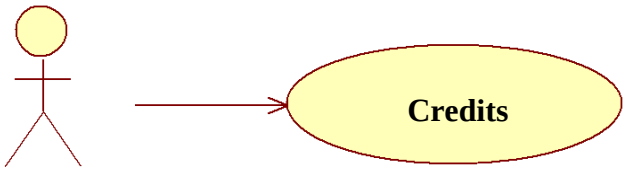


Brief Description:

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

2.2.1.4 *Credits*

Diagram:

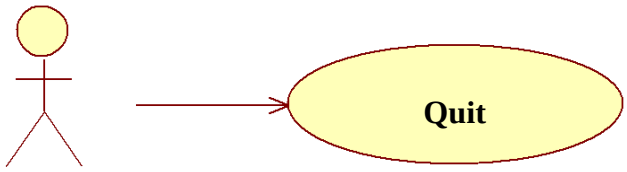


Brief Description:

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

2.2.1.5 *Quit*

Diagram:

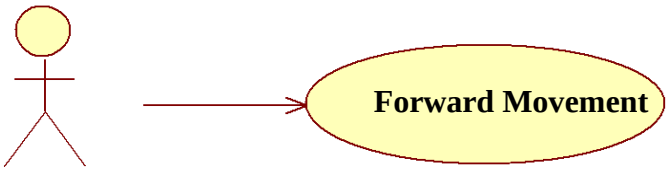


Brief Description:

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

2.2.1.6 *Forward Movement*

Diagram:

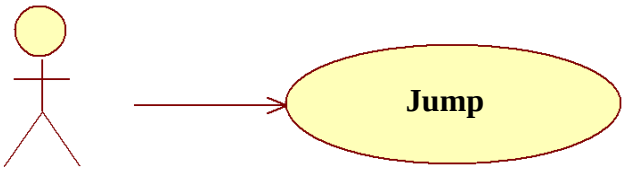


Brief Description:

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.

2.2.1.7 *Jump*

Diagram:

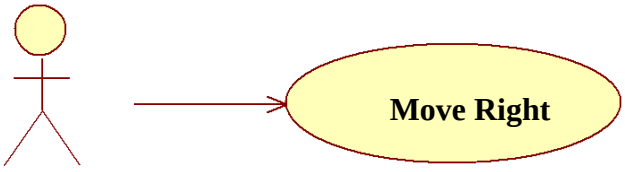


Brief Description:

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

2.2.1.8 *Move Right*

Diagram:

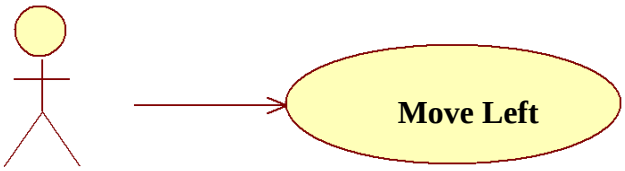


Brief Description:

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

2.2.1.9 *Move Left*

Diagram:

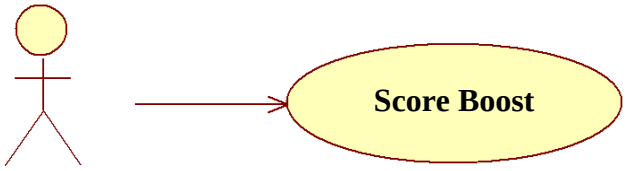


Brief Description:

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

2.2.1.10 *Power Up*

Diagram:

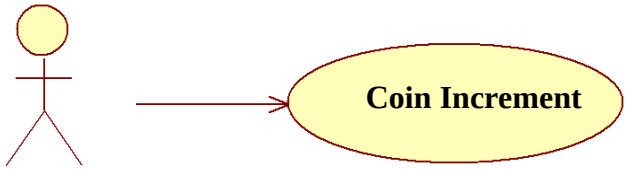


Brief Description:

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

2.2.1.11 *Coin Increment*

Diagram:

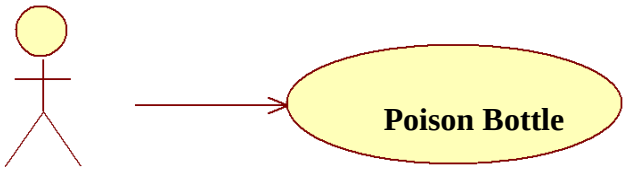


Brief Description:

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

2.2.1.12 *Poison Bottle*

Diagram:

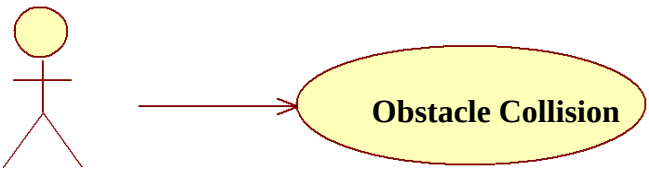


Brief Description:

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.

2.2.1.13 *Obstacle Collision*

Diagram:

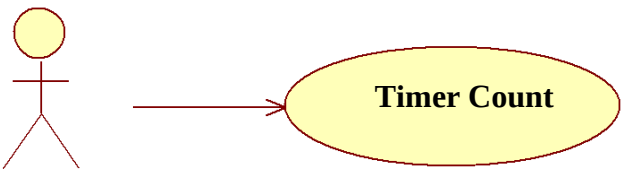


Brief Description:

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.

2.2.1.14 *Timer Count*

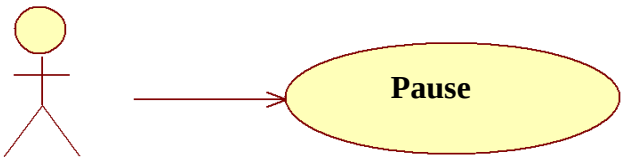
Diagram:



Brief Description:

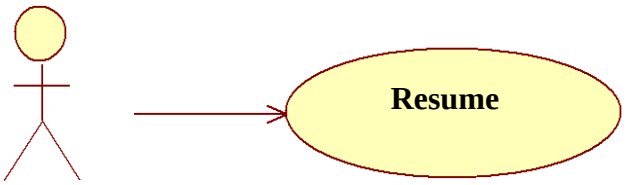
This functionality becomes activated when the character starts running after pressing SPACEBAR. This function counts the time that has passed since the user started running and displays it on the screen.

2.2.1.15 *Pause*
Diagram:



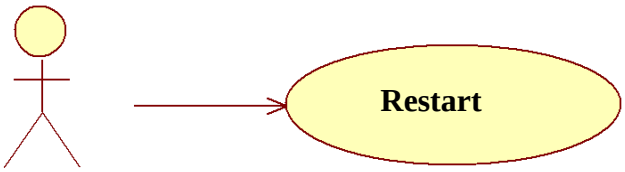
Brief Description:
This functionality becomes activated when the user clicks on the “PAUSE” button on the screen while the character is running. This function pauses the game and displays PAUSE menu.

2.2.1.16 *Resume*
Diagram:



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

2.2.1.17 *Restart*
Diagram:



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

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

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

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

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

- At least 10 different characters for the player to select.
- More than 5 levels of the game with different themes
- Endless runner game mode selection option.

3 Specific Requirements

3.1 External Interfaces

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

3.2 Functions

Following are the functional requirements of the game

- The user should be able to choose characters from a number of characters given in the game.
- The system shall provide help functionality to the user which will include instructions of how to play the game.
- The user shall also be able to see developers name using the credit button.
- The user shall be able to enable the forward movement of the character using the spacebar key.
- The user shall be able to make the character jump using the forward key or W key on the keyboard.
- The user shall be able to make the character move right using the rightward key or D key on the keyboard.
- The user shall be able to make the character move left using the leftward key or A key on the keyboard.
- The game environment shall have power up which will increase the total score by 1000 whenever the character strike with it.
- The game environment shall have coins which will increase the total score by 1 when the character strike with it.
- The game environment shall have a poison bottle whenever the character strike with it the game should end.
- The game environment shall have obstacles like rocks and boxes whenever the character strike with it the game should end.
- The user shall be able to pause the game by clicking on the pause button which is always available on the screen while the game is running.
- The user should be able to resume the game, resume button will pop up when the user have paused the game.
- The user should be able to restart the game, restart button will pop up when the user have paused the game.

3.3 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.

3.4 Logical Database Requirements

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

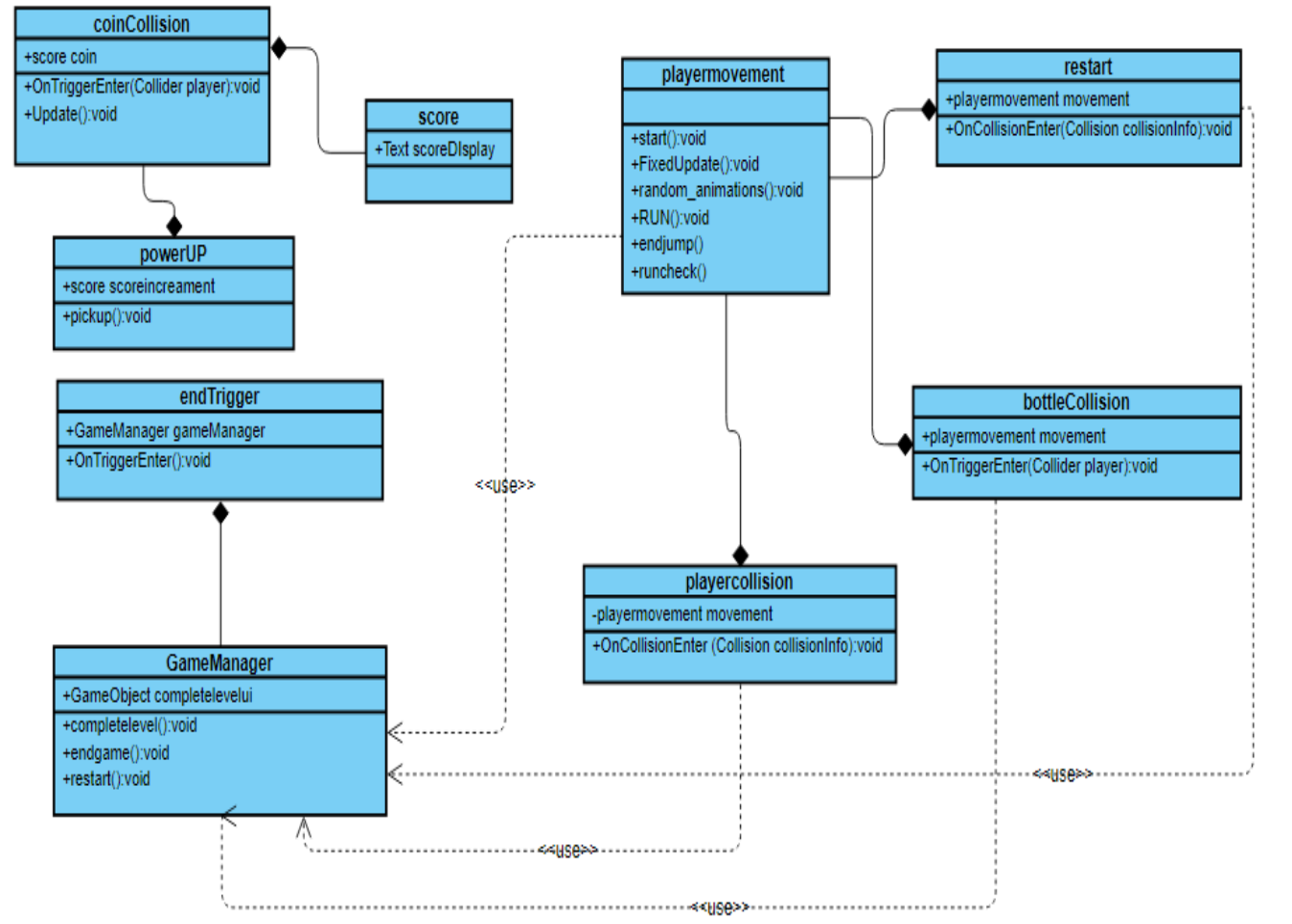


Figure 3- Class Diagram

3.5 Design Constraints

The game only support windows 7 and above. User must have a 64 bit operating system with a processor also of 64-bit to install and run the game. Minimum 4GB of RAM is required to run the game whereas more number of cores in CPU will help to run the Game more smoothly. Unity 2019 will be used for the development of the game and it supports 7.3 version of C# therefore for further upgradation of game Unity 2019 and 7.3 version C# or above must be used.

3.6 Software System Quality Attributes

3.6.1 Reliability

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

3.6.2 Availability

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

3.6.3 Security and Privacy

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

3.6.4 Maintainability

All code related to game should be completely documented. Comments of last change and date should be included in the program files. The code shall be in modular form to allow future changes.

Objects that will be use in game should not be embedded in code rather they should be stored in a separate file so that the modifications in the set of objects will be easily possible.

3.6.5 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.

3.6.6 Safety

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

4 Appendixes: Glossary

Following terms and abbreviations have been used throughout the document.

User	**:**	Person who can play the single player or multiplayer mode of the game.
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.
Game Engine	**:**	Software framework designed for the creation and development of video games.
FPS	**:**	Frame per second
Integrated Development Environment	**:**	Software application that provides some good and useful facilities to programmers for software development.
Unity 3D	**:**	A cross-platform game creation system developed by Unity Technologies, including a game engine and integrated development environment.
Visual Studio	**:**	An integrated development environment (IDE) from Microsoft. It is to develop computer programs, as well as websites , web apps, web services and mobile apps.
Blender	**:**	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.