Software Requirements Specification

for

Fire Ball Game

Version 2.0 approved

Prepared by Team 5

22nd March 2022

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to enlist the requirements for the game Fire Ball, created for theDepartment of Multimedia development revolving around Metaverse and Omniverse.

1.2 Intended Audience and Reading Suggestions

This document is for all the members of team 5. This includes Aishwarya Reddy Mannem, Rakshana Bagavathi, Aneerban Chakraborty, Anirudh Reddy Gotike, Rohan MaheshkumarAswani, and Manak Rajeev Agarwal. This document is also for the Project Director Ms.
Simranpreet Kaur and Dr Aznam Yacoub. The SRS should be used by the team for the completepipeline.

1.3 Product Scope

This project aims to provide a fun and engaging game for all individuals. It will incorporate the concepts of Augmented Reality to make the game feel more intuitive to play. This focus on Augmented Reality also gives the opportunity to the user to interact with real world objects in a virtual setting, fulfilling the Metaverse and Omniverse need of the user. The project also aims to add a multiplayer feature that will allow the users to interact with each other and meet new peoplevirtually. The multiplayer mode is focused to promote mental health in the post-pandemic era andpromote healthy interactions with others. The game is also based on physical movement by the user, promoting health and fitness, through its facial tracking for the user input.

2. Overall Description

2.1 User Needs

The planning to develop this game has been done by keeping in mind the things that were requiredby the customer and our analysis of how the game could help other people. The requirement of having a game in metaverse or omniverse, it was decided to make a game in Augmented Reality (AR). The game should be a stress buster for those who are not able to go out due to the pandemic, it should also involve physical activity and interaction with the outside world digitally. So,we propose to make a game that will solve all of these problems. The game will be played on windows devices and can be played solo or multiplayer and even in localhost and online. To score in the game, the user will have to dodge the fireballs and collect the in-game currency.

2.2 Operating Environment

The first phase of development, which is the ten-week period focusses on the Windows PC versionof the game. Additionally, the game is planned to be ported to both Android and iOS platforms. It will be developed using C# and Unity engine for cross-platform support.

2.3 Design and Implementation Constraints

The project is on a strict timeline with limited workforce. The team is only a group of six engineersthat will be working on this game for eight weeks, putting 500 hours altogether on the project. Any adjustments or requirement changes that go beyond this document will take a longer time.

3. Use Cases

1. Menu for Navigating the game:

- **Input:** The user starts the game
- **Output:** A Menu screen
- **Processing:**
    - The user clicks on the game application and the game starts.
    - The game loads Menu for navigating the game.

2. Single Player Mode:

- **Input:** The user chooses the start single player mode option
- **Output:** Game screen with falling objects where the user can play the game
- **Processing:**
    - The user clicks on the game application and the game starts.
    - It provides a menu where the user can choose playing on a single player mode.
    - On clicking the single player mode, a game screen with single player mode should start.

3. Falling objects:

- **Input:** After the user selects the single player mode, the objects start falling from the top of the game screen and these objects include fireballs and coins
- **Output:** The falling objects(fireballs and coins) must fall at an equal pace and from the random positions
- **Processing:**
    - After the game starts, the objects start falling from the top of the screen one by one.
    - Depending on the object the user must decide whether they have to collect the object or dodge the object.
    - When the object is coin, the player should collect it and when it is fireball, they should dodge it.
    - When one object falls and the operation like collecting or dodging is

done then the next object should come into picture.

4. **Object collision**:
   - **Input:** When the object meets the player then there is set to be a collision.
   - **Output:** There are two outputs for this depending on what the object is, either there is an increase in the points or the game ends.
   - **Processing:**
     - There are two different objects falling from the top, they are coins and fireballs.
     - When the coin collides with the player, then the points in the reward system increases.
     - When the fireball collides with the player, then the game ends.

5. **Reward System**:
   - **Input:** When the coin object collides with the player.
   - **Output:** There is an increase in the points which is displayed on the reward system.
   - **Processing:**
     - When the coin objects falling from top meets the player the points in the reward system are increased.
     - The initial score of the player is set to 0.
     - As the score increases and reach a target level points, the player is reached to the next higher level where the speed of the objects also increases.

6. **Room Creation for Multi-player Mode**:
   - **Input:** The Player is asked to give their name and select one of two options: whether they want to join or create the room.
   - **Output:** The player can add the other player into the game or the player can join other player's game.
   - **Processing:**
     - When the player chooses the multi-player mode displayed on the menu.
     - The player is asked to give their name, Once the player enters the name, It gets connected to the photon server, and then they are provided with two options: join or create.
     - When the player selects join they are provided with a text box where they have to enter a four-digit code to enter the game.
     - When the player selects the create option they are provided with a four-digit code that they can share with their friends so that they can join the game.
     - Multi-player game can be accessed using room creation.

7. **Multi-player Mode**:
   Input: After the room is created, the players join the game.
   Output: Game screen with falling objects where the users can play the game in Multiplayer mode.
   Processing:
   - The users can play the game in multiplayer mode, where they can track each other's Reward system and engage in a healthy competitive game environment.
   - The player is declared the winner when one of the other player comes in contact with the fireball.

4. Functional Requirements

1. **Room**: It is defined as a virtual session where at most 2 people can connect using proton DB. The code must be generated by one of the game participants and the room code must be communicated to other game participants explicitly.
2. **Action Reflection**: When the players are in a room, their score and body movement will bereflected on each game participant's screen.
3. **Avatars**: For the detected body in the video feed, the face can have a filter from a given listof filters already provided by the game. Each individual face filter is termed an Avatar. The number of avatars is limited to 10.
4. **Modes:** The game will provide three modes of play:
   a. **Single Player**: This mode is defined as one person playing the game standing insidethe bounds of the camera frame.
   b. **Multi-Player (Online)**: This mode is defined with a maximum of 2 people joining into 1room and playing together. Action Reflection will perpetuate among all the participants.
   c. **Multi-Player (Random)**: This mode is an extension of (b) and is defined by the pairing ofa user with another user randomly online.
   d. **Party Mode (Local Multiplayer)**: This mode is defined with a maximum of 2 peoplestanding within the camera frame and playing the game together.
5. **Falling 3D Objects:** There are two kinds of 3D objects falling from the top of the game screen, one is the fireball and the other is the coin. The 3D objects fall randomly at a random pace.
6. **Rewards:** The Reward System displays the score of the player on top of the game screen. Whenever the player collects the coins, the score is incremented by 1.
7. **Menu**: The menu of the game is defined as the options visible to the user where he cannavigate to choose the game mode, create a room, see the high score, and choose an avatar.

4.1 External Interface Requirements

1. **Camera**: The game requires a webcam/front facing camera on the dplayersevice to function.
2. **Computation**: The game uses face-mesh deep learning algorithm based on ARCore to allow smooth functioning on mobile devices.
3. **Internet Connectivity**: The multiplayer mode requires good internet

connectivity for asmooth experience.

4. **Surroundings**: Being an AR game, the user surrounds play a part in the gameplay, therefore it is recommended to have ample space to move around in the camera frame.

4.2 Software Interfaces

- On game start the Operating System should start compiled Unity build
- Unity script calls should call ARCore methods and interfaces.
- The Device Camera should send the video feed to ARCore and ARCore should return Face locations.
- ARCore should return Face locations on interface call from C# Scripts.
- The updates in the game by Unity should be sent to the Operating System on the displayed video.
- Unity build should look into its database for available falling objects.
- Unity build should look into its database for available face masks.
- Unity should communicate with the Photon Server for available multiplayer rooms
- The Photon server should communicate to Unity the available rooms.
- Unity should push the local player updates to Photon Server.
- Photon Server should sync updates from each player on screens of all other players.
- Unity should communicate the addition of a new player into the room.
- Photon server should add the person into the room.

5. Constraints

- Android 8.1 Oreo or higher Or IOS 11 or higher for ARCore
- At least 2GB RAM
- A minimum of 100MB of available storage
- 720p or higher front-facing camera

6. Other Nonfunctional Requirements

6.1 Performance Requirements

The game should function smoothly without any errors, meeting the demands of the majority ofusers, and the processes will be monitored to maximize performance.

6.2 Compatibility

Initially, the game will be a PC standalone version and later followed by mobile platforms like iOS, Android. Updates will be issued on a regular basis, and compatibility with the majority of versionswill be ensured.

6.3 Software Quality Attributes

The application must deploy all the features listed in the game menu without latency or glitches,and it must be extensively tested by the testing team.

6.4 Availability

We don't require any specialized, expensive, or difficult-to-acquire devices for this game, so usersmay play it simply on a PC or a mobile phone.