

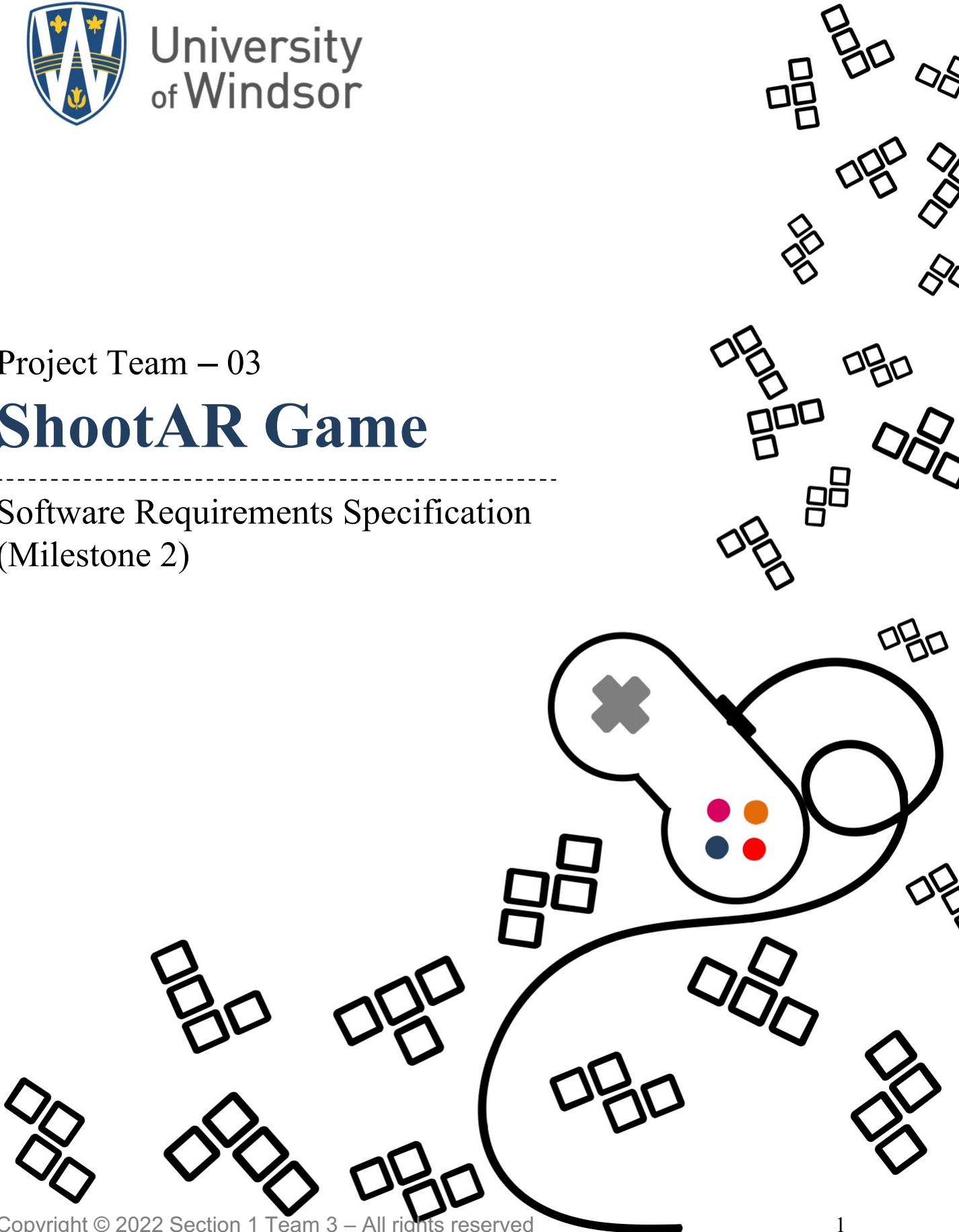


University
of Windsor

Project Team – 03

ShootAR Game

Software Requirements Specification
(Milestone 2)



Object:

Document Title
Software Requirement Specifications

Submitted To:

Project Director
Ms. Simranpreet Kaur

Project Customer
Ms. Shivani Kapadia

Submitted By:

Team Members
Kameswara Saidatta Srinivas Peddada - 110081329
Kartik Peddinti – 110086484
Keneel chirag shah – 110073464
Krishna Sravanthi Telapudi – 1100727725
Mrinal Walia - 110066886
Varnita Sharma - 110074272
V V R Sricharan Apparayacheruvu – 110086372
Venkata Sai Vardhan Seepala - 110074717

Date : **20-March-2022**

Table of Contents

1. Introduction	5-7
1.1. Document Purpose	5
1.2. Project Scope	5
1.3. Intended Purpose	
1.4. Definitions, Acronyms and Abbreviations	5
1.4.1. Definitions	5
1.4.2. Acronyms	5
1.4.3. Abbreviations	7
1.5. References & Standards	7
2. Overall Description	8-10
2.1 Product: Overview	8
2.2 Product: Functionality	8
2.3 Product: Targeted Users	8
2.4 Product: Design and Implementation Constraints	9
2.5 Product: Operating Environments	9
2.6 Assumptions and Dependencies	10
3. Specific Requirements	11-14
3.1. User Interface	11
3.2. Hardware Interface	11
3.3. Software Interface	12
3.3.1. Unity & C#	12
3.3.2. JIRA & GitHub	12
3.4 Flow Chart	13
3.5 SADT Diagram	14



4. System Features	14
5. Non-functional Requirements	15-17
5.1. Performance	15
5.2. Safety	15
5.3. Security	16
5.4. Software Quality Assurance	16
5.4.1. Adaptability	16
5.4.2. Flexibility	16
5.4.3. Maintainability	16
5.4.4. Reliability	17
6. UML Diagrams	17-18
6.1. Use Case Diagram	17
6.2. Activity Diagram	18

Software Requirement Specification

1. Introduction

This Software Requirements Specification (SRS) outlines the fundamental SRS with purpose, scope, definitions, acronyms, abbreviations, references and intended audience of the SRS.

1.1 Document Purpose

Our SRS document aims to provide detailed specifications and descriptions of the game we develop. Our system is categorized into three parts for this milestone.

1. Different levels implementation where we have successfully executed different levels with the timer option to play the game.
2. Sound effects that includes GBM sound effects and enemy kill sound effects.
3. Lastly, we have implemented pop up menus for the starting, next-level and the ending of the gameplay.

1.2 Product Scope

Although AR provides a futuristic vision or may sound like a revolutionary technology, the facts say that it has been around for more than five decades now. As an AR game, ShootAR brings your digital environment to reality by identifying virtual objects in the real world. Our game recreates the movements of a human hand in remarkable detail, giving it an entertainment angle, too.

Our project seeks to determine the influence of AR mobile games on fine motor skills in young adults, an area of incomplete result and verification. With ShootAR, we aim to positively influence the basic motor skills of individuals, such as precision, aiming, speed, agility, or tremor. Our game players will perform significantly better in the accuracy of arm-hand movements with lower time and error rates.

1.3 Intended Audience

This document aims to collect, analyze and produce an in-depth understanding of the features we

have implemented until milestone 1 to our professor, project manager and customer. We will describe the game features, characteristics, and design implementations. The detailed requirements of the shootAR system are provided in this document.

1.4 Definitions, Acronyms, And Abbreviations

This section describes all terms, acronyms, and abbreviations used in this SRS.

1.4.1 Definitions

None.

1.4.2 Acronyms

IEEE- Institute of Electrical and Electronics Engineers

SRS- Software Requirement Specification

GUI- Graphical User Interface

UML- Unified Modeling Language

SADT- Structured Analysis & Design Technique

FAST- Functional Analysis System Technique

AR- Augmented Reality

iOS- Operating System

FPS- Frames-per-second

1.4.3 Abbreviations

None.

1.5 References

This section highlights a list of documents referenced in the SRS.

- I. <https://krazytech.com/projects/sample-software-requirements-specificationsrs-reportairline-database>
- II. <https://futurism.com/the-byte/augmented-reality-first-person-shooter-awesome>
- III. <https://thearea.org/area-resources/augmented-reality-functional-requirements/>
- IV. <https://assetstore.unity.com/packages/templates/packs/ar-shooter-ar-foundation-arkitarcore-augmented-reality-fps-game-192936>
- V. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5643620>

1.6 Standards

We will review every submission to JIRA and GitHub before adding it to the project. After that, we will successfully test that code block, and then it will be merged into the project code. Every code block will follow the defined rules and coding standards (below). With all the updates and modifications to the coding scripts, comments will be added. The software methodology will increase engagement, enthusiasm, and transparency from our daily scrum meetings and team interactions.

Below are the set of standards used for this SRS:

1. [2]IEEE Std 830 IEEE Recommended Practice for Software Requirements Specifications
2. C# <https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/codingstyle/coding-conventions>
3. Unity <https://docs.unity3d.com/Manual/UnityManual.html>

2. Overall Description

Stability is essential for our wrists. We should move through the regular planes of movement while keeping correct alignment and support when loads are applied. We all shoot videos with our smart devices, and you can make sure the video you shoot with your smartphone is something worth sharing.

We have got you covered by developing an effective hand stabilization and yet entertaining game that will help you improve your aim and stabilize your hand movements. ShootAR is an augmented reality (AR) shooting game system based on IOS devices and runs in real-time.

While playing the game, some monster will appear on your device's monitor, and the user is equipped with multiple options of guns. The user must open fire and shoot at the monster within the allocated time to move on to the next level.

Now we will explain the implemented game design and features. It also describes the game's functionality and design constraints with assumptions and dependencies. We will also highlight the target users we are planning to reach.

2.1 Product: Overview

We are building a game that will help users stabilize their hand movements. It corrects alignment and stability when loads are applied to the hands of the users. It involves shooting enemies with the intelligent device using AR technology, and our game runs on iOS and Android devices in real-time. With the options of multiple guns and game modes, the users can fire and shoot in real space within the allocated time and then proceed to the next level.

2.2 Product: Functionality

The game is still in the expansion phase. Until milestone 2, we have implemented some level functions to our game. Some of the functions that are successfully executed and are working without any bugs include:

- The AR technology is the backbone of our game project.
- The player can now play level 1 successfully, and they can proceed to the following levels now.
- Each level now has a timer, which shows when the level is finished.
- When the user sees the enemies, he can shoot them in real-time, and he can also hear the shooting sound when the bullet hits enemies.
- After launching the game, the player is introduced to a pop-up intro screen. The game after the opening will be in AR mode, and it displays the PLAY button in 3D space and in the background, the user can hear the BGM sound effects.
- After shooting the enemies and completing level 1, they can see the next level pop-up intro screen when the player proceeds to level two.
- If the player fails to kill all enemies on time, they can see the game over the pop-up screen.

2.3 Product: Target Users

Our target user base is anyone aged more than 12 years old. This is a hyper-casual game. This game is meant for entertainment and includes fantasy characters. This game uses cutting-edge technologies like the Augmented Reality (AR) Unity game engine. Augmented Reality (AR) games use a smartphone camera. This can help people with shaky hands take better photos. This is because shooting in AR needs hands to be stable.

2.4 Product: Design and Implementation Constraints

We have put some features through the development stage and are not yet completed. Some of the constraints in the design are:

- Adding the scoreboard requires connecting the game to the database. This high-score section is not yet added to the opening page.
- The game won pop-up window is not implemented yet and requires much research.
- Adding a custom 3D enemy model requires high-level knowledge of unity and C# libraries. We are still learning to work on that part.
- Adding a custom gun model requires high-level knowledge of unity and C# libraries. We are still learning to work on that part.
- The additional feature to save the Queen and shoot all the enemies is yet to be implemented. It could not be completed due to time constraints.

Some of the implementation constraints are: Right now, as this is milestone 2, we have not implemented all the features discussed in the proposal. Later in the latter stage, all the features will be implemented with time. The player will feel the AR feature in the game but not be able to completely utilize all the AR features.

2.5 Product: Operating Environments

As an AR game, ShootAR brings your digital environment to reality by identifying virtual objects in the real world. Our game recreates the movements of a human hand in remarkable detail, giving it an entertainment angle, too. The AR works in a real-world environment where computer-generated software and codes enhance the enemies in 3D space. ShootAR is compatible with iOS and Android devices with the help of Unity software and the packages we are using to develop the game.

2.6 Assumptions and Dependencies

To test our game in the real-world using AR technology, we need a powerful computer to render the game quickly. We also need android and iOS devices with different versions to test the dependability of the game.

Our game prototype is an accurate AR model that provides the users with an entertaining experience. The game's features are precise and are not throwing any bugs right now. The game is faster, giving playable FPS and good quality.

3. Specific Requirements

This section discusses the software system and functions implemented in the project's first phase. It also includes features of prime focus to the user and the customers.

3.1 User Interface

When it comes to the User interface of the AR games, we wanted to reinvent the wheels of the UX design. Our interface is clean, interactive, and compatible with all iOS and Android devices.

It is tested on real devices and emulators for better performance. In this part, we took advantage of the existing knowledge from our project directors and customers.

Currently, the initial page and warmup page allow you to direct your input in AR space through an action (shooting the PLAY button) and launch the playable screen. At this stage, by interacting with our game, you will feel like you are interacting within a natural world environment. After this, the level 1 starts, and the users can play the level 1 completely and see the timer for the game. If the player loses, there is a Game over pop-up window.

3.2 Hardware Interface

As we all know, AR is an up-and-coming technology available to us today. The major hardware equipment available to us today includes smartphones (iOS Phones (iOS > 7), Android Phone (Version > 7.0)) which are revolutionizing every day. Some other hardware components are covered under this segment with the smartphone, like a processor, display screen, camera, microphone, etc.

Some of the hardware sensors used in building our AR game are accelerometers, orientation, position tracker, and gyroscope sensors that come inbuilt with the most intelligent devices. With the help of all the above hardware components, we are successfully able to perform 3D-modelling, real-time tracking of the player, intelligent interactions, multimedia sensing and much more.

3.3 Software Interface

The software for ShootAR is made using a variety of complementary technologies where the core software technologies include:

3.3.1 Unity & C#

Unity is the backbone of our immersive ShootAR game. It provides rich toolsets, libraries and packages to develop industry-leading games. We are using Unity with the C# coding language to build the scripts which perform different functions. It provides engaging augmented reality experiences that intelligently interact with the game user.

The coding scripts are also used to create the starting warmup page, add BGM effects, implement level 1, respond on touch, add icons like gameplay button and timer of the current level on the gameplay, and more. Using different libraries supported by Unity, we make and test our game on different platforms like Android or iOS without any additional efforts. This package is open- source and easily downloaded for Linux, Windows, or Mac OS.

C# scripts are well supported by Unity, and it powers everything the engine does for 3D rendering. Using C# scripts is the best tool that allows a developer to make an AR game on Unity and create custom actions and interactions within a game space.

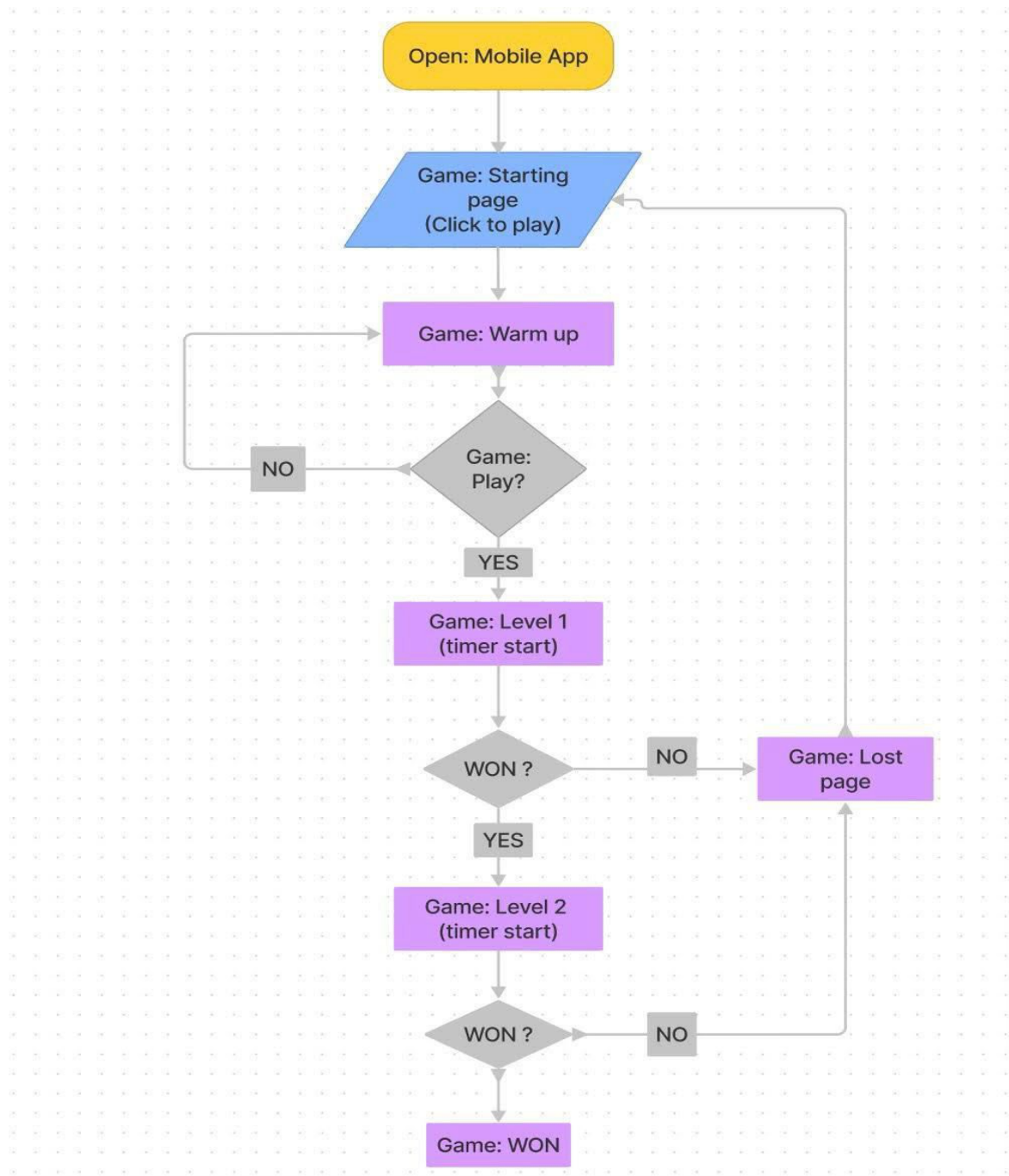
3.3.2 JIRA & GitHub

JIRA and GitHub are software technology tools that help keep track of the progress. JIRA is an open-source software using which we assign, organize, and implement changes and modifications. It helps our team to drive workload to agile software development. It also helps us report and resolve bugs and issues by tracking them in the cloud storage.

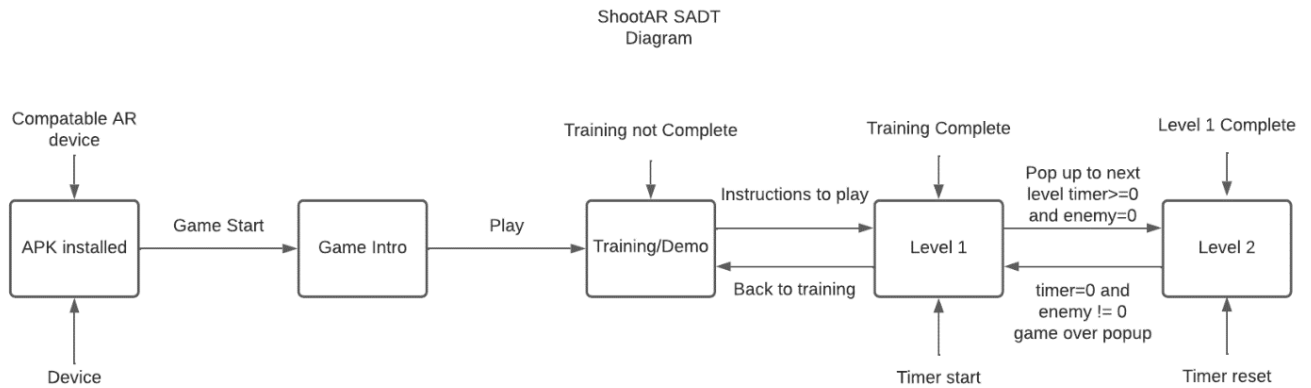
GitHub is another open-source internet hosting tool for software hosting and version control. We track source code functionality and other game features being developed using this tool. It helps all the team members to collaborate and work together on the ShootAR game remotely. It is a great tool that is safe, secure, and easy to use for all software engineers.



3.4 Flow Chart



3.5 SADT Diagram



4. System Features

The functional needs implemented in the game are referred to as system features. They are the game's primary aspects with which the user interacts and has an experience while playing.

The following is a list of the qualities that have been implemented:

- **Game Levels:** We have implemented level 1 successfully in this sprint, and we are working on the following levels.
- **Pop-Up Windows:** We have successfully designed and implemented the level 1 promotion pop-up screen, game introduction screen, next-level promotion pop-up screen and the game over screen.
- **Level Timer:** We have implemented the timer displayed once the level is started. It is the time left to kill the enemies and progress to the next level.

- **Sound Effects:** The background sound is added to the gameplay. Additionally, the sound effects from killing the enemy and shooting the bullet are also successfully implemented for this milestone.

5. Non-functional Requirements

Non-functional Requirements serve as the rules on the system's design across the implementation of various features and usability of the software.

5.1 Performance

1. The performance of the current sprint is supported with a quality FPS and doesn't result in frame drops most of the time. This fulfils our goal to deliver the game as close to reality.
2. We will maintain the code quality, code length, graphics and features of our game and sprint to provide the most satisfying experience to the players.
3. The sprint's quality will be checked for performance simultaneously.

5.2 Safety

1. The game is very safe to play, and we took care of the character's animations so that children cannot be influenced by the violence.
2. There is no bloodshed in the game, and the weapons are animated, keeping children in mind.
3. We follow the CIA standard to maintain the integrity privacy of the target users.
4. No data phishing can be done using our app, as this game does not access the user's private data.

5.3 Security

1. Authentication will be achieved right after the game starts.
2. The game is entirely offline, so hacking the user's devices is impossible.

3. We don't have access to the user's confidential data. So, there are no conflicts or security vulnerabilities present in our game.

5.4 Software Quality Assurance

Software quality assurance (SQA) ensures software quality by evaluating the software engineering processes and methods utilized in a project.

5.4.1 Adaptability

In our game, there are different levels. The player can experience different interactions throughout, and the game is designed so that the player can quickly adapt to the levels. We guarantee we deliver more reality and fun with good FPS, making our game more adaptable to users' interests.

5.4.2 Flexibility

The game is very flexible for every user. Let it be an adult playing it or a child. Everybody will enjoy the game. A person can start with the basic level and go further into the game. The game's overall quality will be checked for flexibility during the testing phase, and it can quickly render the game frames without buffering or dropping frames.

5.4.3 Maintainability

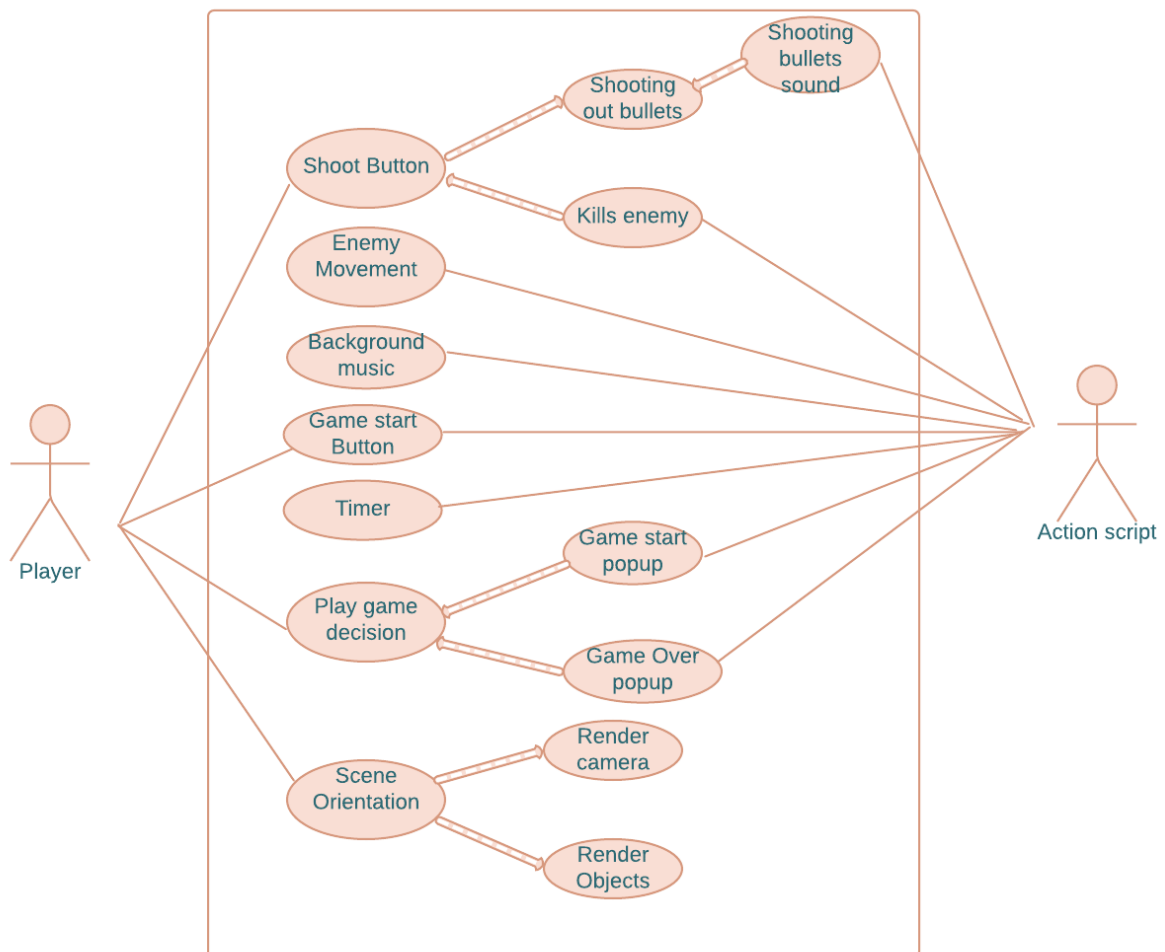
The ShootAR game is tested and designed by a team of experts and ensures the easy maintainability of the game and hardware parts involved in the ShootAR. Every aspect of the code is written with precision and is reliable to its fullest. The supported IOS devices (iOS 11 & above) and Android devices (Android version > 7) are supported, and our game has high-end 3D graphics with interactive objects and items which are written in Unity and C# in a structured and clean formatted script.

5.4.4 Reliability

Our game for the users is designed to deliver entertainment with excellent motor skills. It outperforms every feature suggested by the customer efficiently and at a consistent level of accuracy. We provide a reliable and straightforward user interface that is easy to use. The ShootAR game is reliable to stand against the different classes of error using different types of testing, making it more reliable.

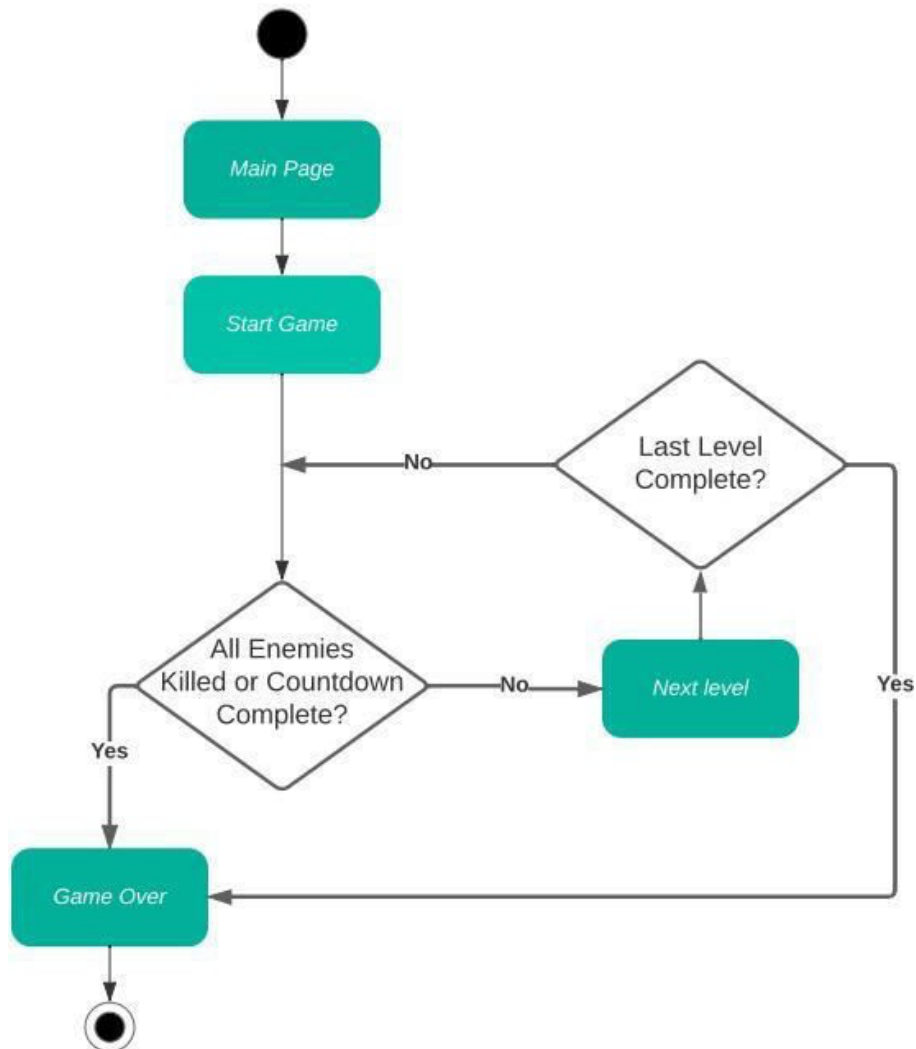
6. UML Diagrams

6.1 Use Case Diagram



ShootAR Use case Diagram

6.2 Activity Diagram



ShootAR Activity Diagram