CHAPTER 1

Introduction

1.1 Background

Today, computer games are one of the most popular and profitable fields in the world. Shooting games are now one of the most popular games across the globe. The first shooting game is often attributed to the game "Maze War," a 2D game that was developed by Steve Colley, Howard Palmer, and Greg Thompson at the NASA Ames Research Center in the early 1970s. Over time, shooting games evolved from 2D to 3D. These factors have inspired us to make a 3D shooting game.

1.2 Objective

The application is an action-packed 3D shooting game designed for battle enthusiasts seeking an exhilarating experience. This shooting game allows users to reduce their stress and boredom. In this game, players have to eliminate multiple enemies to win. The controls are user-friendly, which allows anyone to play the game easily.

1.3 Purpose,Scope and Applicability

1.3.1 Purpose

The purpose of this game is to provide players with a stress-relieving experience who are interested in action-packed 3D shooting games. Players will engage themselves in thrilling battles, eliminating multiple enemies while having fun with user-friendly controls, ensuring accessibility for all players.

1.3.2 Scope

The scope of the game includes action-packed 3D gameplay that includes diverse maps. It provides interesting graphics, a user-friendly interface, and stress-relieving gameplay.

1.3.3 Applicability

The game is applicable to gamers who enjoy 3D shooting games and seek stress relief through engaging battles and user-friendly controls.

1.3.4 Achievement

The achievement of this 3D shooting game lies in its high level of immersion and visual appearance. The use of 3D graphics and environments enhances the gaming experience, making it more realistic and engaging for players. Additionally, the complex and varied gameplay mechanics contribute to a deeper and more realistic gaming experience overall.

CHAPTER 2

Survey of Technology

2.1 Existing System

As the gaming industry evolves, it continues to attract more individuals, resulting in a significant increase in game development platforms. These platforms play a crucial role in making games captivating and engaging. They use various aspects of game creation, mechanics, user interface, visuals, and overall presentation, allowing developers to create more fascinating and realistic gaming experiences.

2.2 List of Technologies

Unity Engine is a versatile and user-friendly game development platform known for supporting multiple platforms, a vast asset store, and a strong community. It uses C# for scripting and caters to both beginners and experienced developers.

CryEngine: A powerful game engine known for its amazing graphics and realistic environments, popular for AAA titles.

Titanium (Appcelerator Titanium): A cross-platform mobile app development framework used for developing basic 2D games with JavaScript.

App Game Kit (AGK): A versatile 2D game development engine supporting C++, BASIC, and Tier 1 languages Simplifies game development with built-in functions.

Unreal Engine: A powerful game engine renowned for high-quality graphics, advanced features, and C++ and blueprint support. Popular for creating immersive games across various platforms.

Lumberyard: Lumberyard is a game engine by Amazon, designed for AAA-quality games with integrated Amazon Web Services. Offers tools for creating visually impressive and multiplayer experiences.

2.3 Comparitive Study

Technology	**Features**	**Advantages**	**Disadvantages**	**Written in**
Unity	-Multi-platform support -asset store	-User friendly -Strong community	-Performance limitations for AAA projects	-C#
CryEngine	-Stunning graphics -realistic environments	-High-quality visuals -AAA game potential	-Steeper learning curve -limited documentation	-C++
Titanium	-Cross-platform mobile apps -Javascript	-Rapid mobile app development -code reuse	-Limited for complex game development	-Javascript
App Game Kit	-2D game development -various languages	-Simplified development -buit-in functions	-Limited for advanced 3D or complex projects	-C++,BASIC, Tier 1
Unreal Engine	-High-quality graphics -advanced features	-AAA game development -extensive tools	-Steeper learning curve -resource-intensive	-C++,Blueprints
Lumberyard	-AAA game development -Amazon Web Services	-Integrated with AWS -visually impressive	-Less extensive community -limited platform	-C++ -Lua

2.4 Selected Technology

Unity is an excellent pick for game development due to its versatility, user friendliness, and strong community backing. With support for multiple platforms, a vast asset store, and beginner-friendly tools, Unity provides an ideal environment to create captivating and high-quality games. Its wide adoption in the industry ensures ample resources and continuous updates for a smooth development journey.

CHAPTER 3

Requirements and Analysis

3.1 Problem definition

In the world of shooting games, there are lots of cool ones with fancy features and amazing graphics. Some of these games ask you to buy stuff in the game, which can be a bummer. Another thing is that these games sometimes look weird or have funny graphics. They don't always work smoothly; they might suddenly freeze or feel slow. Some of the really good games need a good computer to run well, so if you don't have that, get ready for a lot of crashes. And most of these games kind of force you to play from the angle where you can see your character, which takes away from feeling like you're really inside the game.

3.2 Requirement Specification

3.2.1 Functional Requirements

1. User Registration/Login:

To access the system, users need to register an account using their unique email address. Logging in requires the provision of a valid email and an associated password.

2. User Input:

User engagement within desktop applications will be facilitated primarily through interactions with the left mouse button.

3. Media Content:

The integration of media content significantly enriches player immersion and engagement by effectively capturing the game's ambiance and vitality. This includes a diverse range of sound effects accompanying characters, settings, and in-game entities.

3.2.2 Non-Functional Requirement

1. Reliability:

The game needs to work just the way it's supposed to, and players should be able to use it whenever they want without any issues. It should be strong and not easily break down.

2. Speed:

Even when you're using a bunch of other apps, the game should still run smoothly. Your device's speed should stay consistent and not slow down.

3. Security:

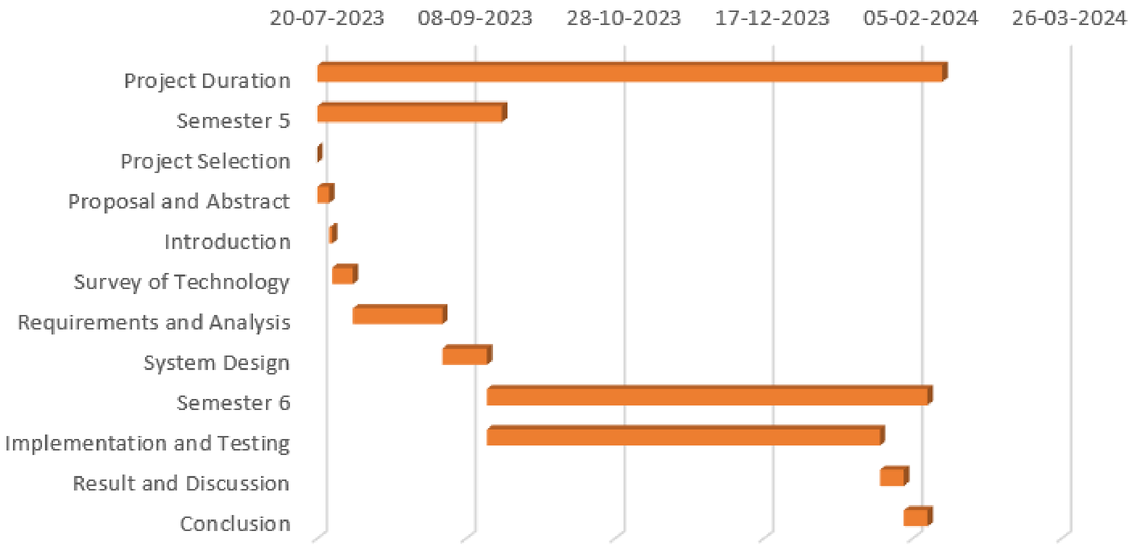
Only the right person with the correct username and password should be able to get into their account.

4. Portability :

No matter if you're using different types of systems, the game should work well on all of them. This way, you can easily play the game on different devices without any problems.

3.3 Planning and Scheduling

Gantt Chart



3.4 Software and Hardware

3.4.1 Hardware Requirements

- Desktop/Laptop with minimum 4/8GB RAM
- 64 bit OS

3.4.2 Software Requirements

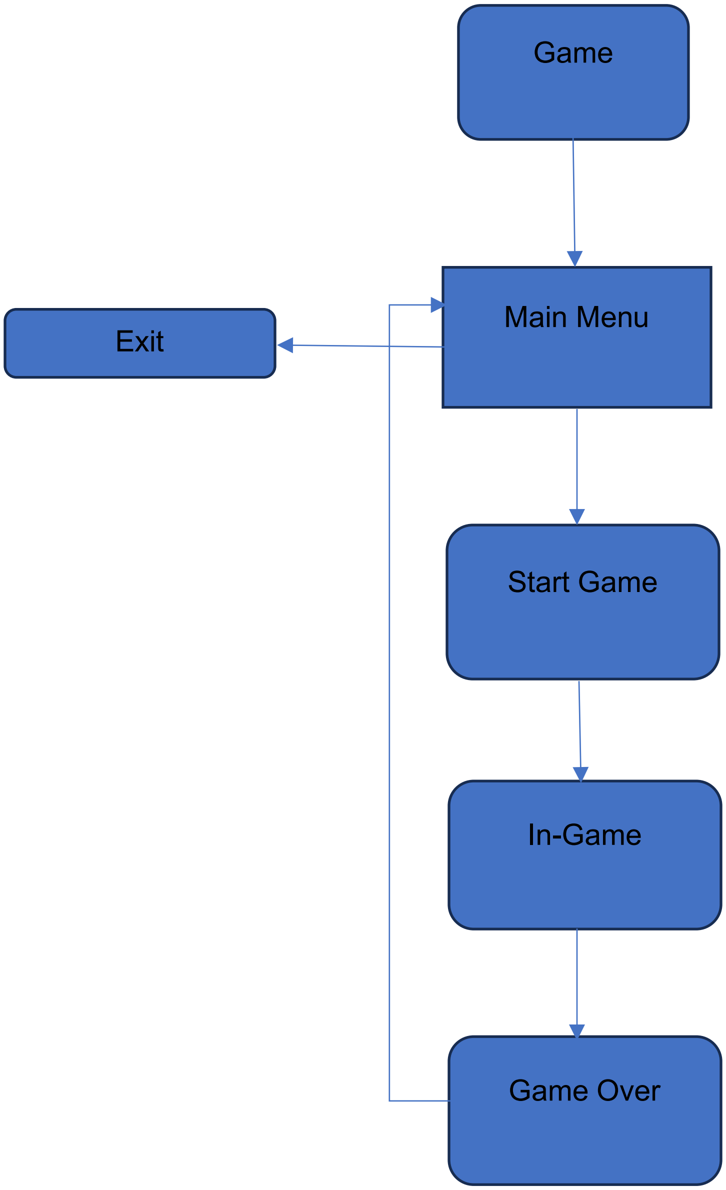
- Unity (Frontend)
- Visual Studio (Backend)
- C# (Programming Language)

3.5 Feasibility

This is a solo-player offline game, eliminating hosting expenses. Engaging in first-person shooter gameplay has been known to enhance cognitive abilities and hand-eye coordination. This genre of games challenges competencies like spatial reasoning, focused spatial awareness, visual precision, and effective decision-making.

3.6 Conceptual Model

Flow Diagram



CHAPTER 4

System Design

4.1 Basic Modules

A module refers to a cohesive assembly of source files and configuration settings within a project. This organizational structure enables the project to be compartmentalized into distinct functional units, promoting modularity and ease of management.

Below are some of the basic modules in the project.

1.Lobby : It contains option like start game.

2.In-Game : It contains map,character,etc.

3.Setting : It contains settings for sound and music.

4.Help : It helps to give instructions about the game.

4.2 System Design

Use Case Diagram

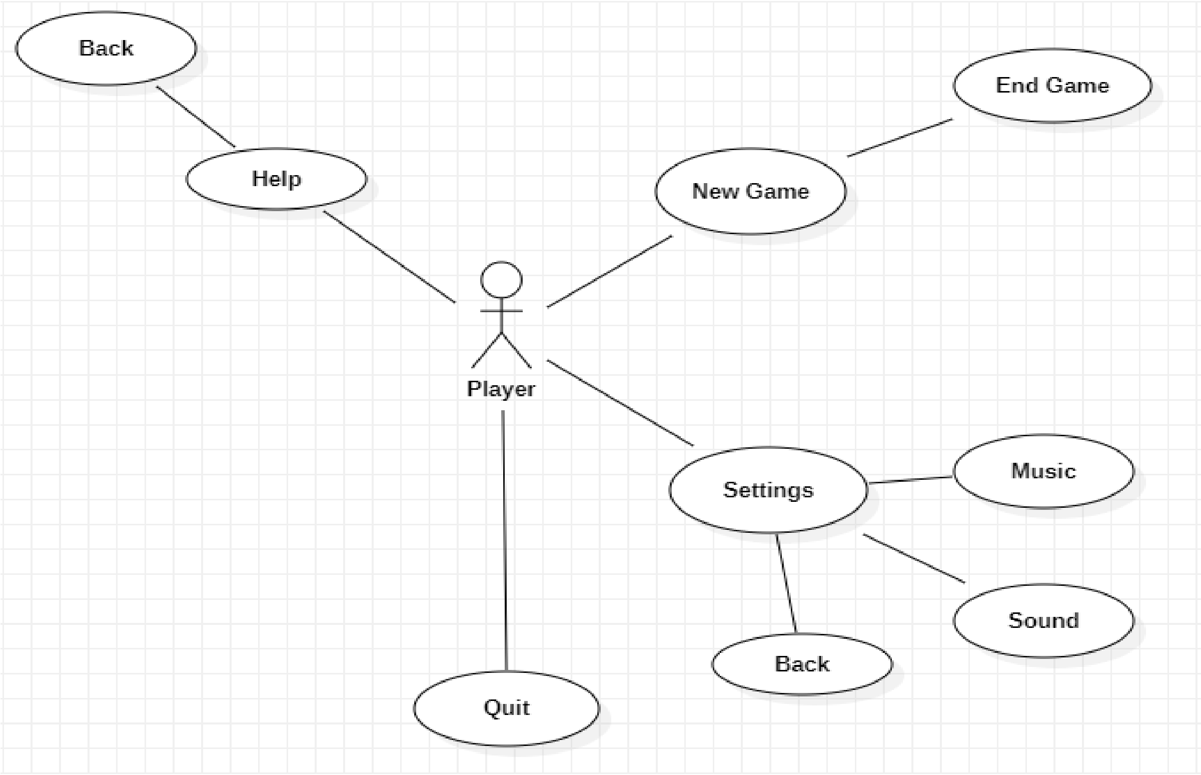


Figure 4.1 Use Case Diagram

Sequence Diagram

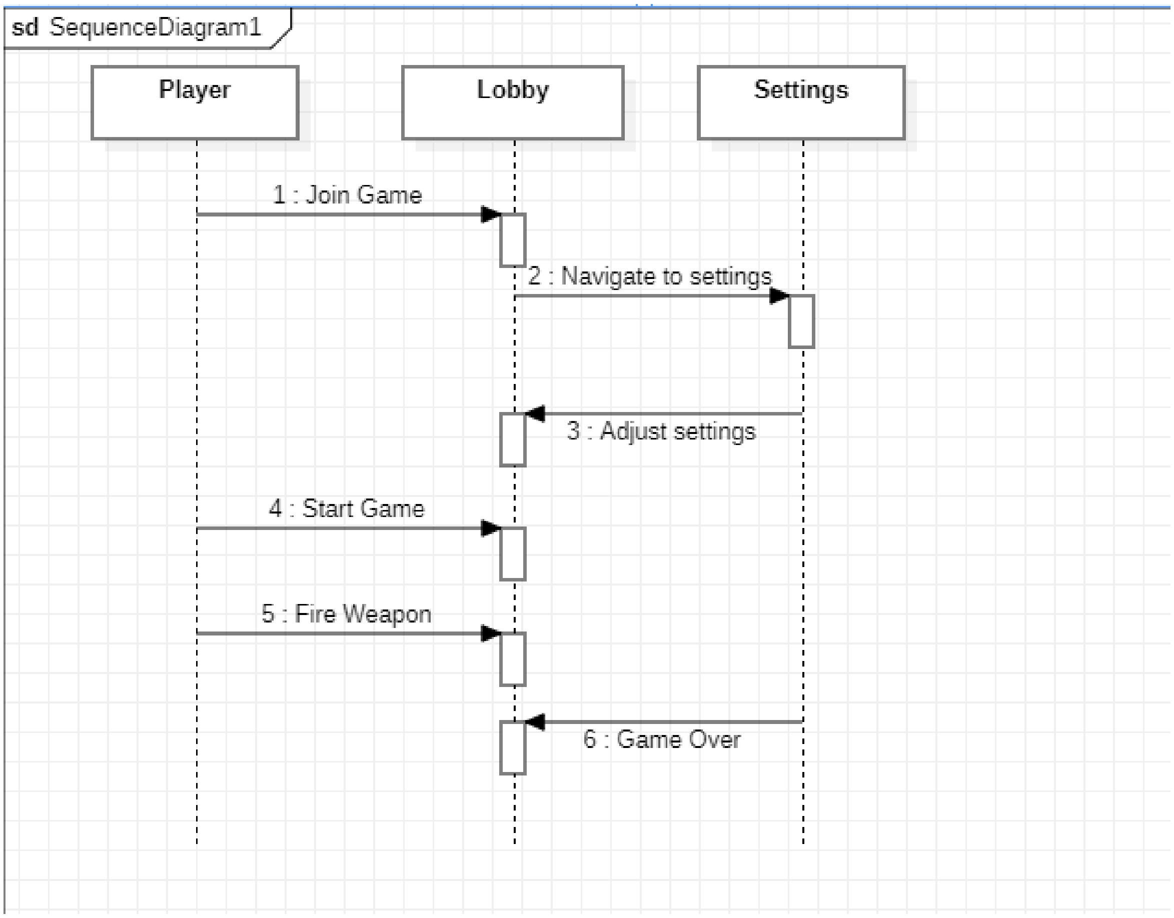


Figure 4.2 Sequence Diagram

4.3 Security Issues

Occasionally, security concerns can arise within games, often as a result of hacking attempts. In certain instances, hackers may compromise the game's integrity by attempting to manipulate a player's score. Additionally, Distributed Denial of Service (DDoS) attacks can occur, leading to server performance issues, such as lag or crashes, which can adversely impact the overall gaming experience. Another potential threat involves the creation of counterfeit login screens by hackers, with the intention of unlawfully obtaining users personal information.

4.4 Test Cases Design

Fields	Description	Expected Output
Left and Right Arrows	Player's input controls to move left and right in the game	Player uses the left key to move the character to the left, and when the right key is used, the character should move to the right.
Up and Down Arrows	Player's input controls to move front and back	Player uses the up arrow key to move the character forward, and when the down arrow key is used, the character should move backward.
Mouse Left Click	Player's input control to shoot	Player uses the mouse left click to shoot in-game.