Gangster's War

Mehtab Naseer **(BCS163046)**

Raja Usman Yaqoob Bhatti **(BCS163037)**

Sohaib Ahmed **(BCS163104)**

Spring-2020

Supervised By

Ms. Mammona Qudsia

Department of Computer Science

Capital University of Science & Technology, Islamabad

Submission Form for Final-Year

PROJECT REPORT

Version	2.0

NUMBER OF MEMBERS	3

TITLE	Gangster's War

SUPERVISOR NAME	Mammona Qudsia

MEMBER NAME	REG. NO.	EMAIL ADDRESS
Mehtab Naseer	BCS163046	mehtabnaseer01@gmail.com
Raja Usman Yaqoob Bhatti	BCS163037	usman.bhatti7787@gmail.com
Sohaib Ahmed	BCS163104	khansohaib817@gmail.com

MEMBERS' SIGNATURES

_____

_____

_____

Supervisor's Signature

APPROVAL CERTIFICATE

This project, entitled as "**Gangster's War**

" has been approved for the award of

Bachelors of Science in Computer Science

Committee Signatures:

Supervisor: _____

(Ms. Mammona Qudsia)

Project Coordinator: _____

(Mr. Ibrar Arshad)

Head of Department: _____

(Dr. Nayyer Masood)

DECLARATION

We, hereby, declare that “No portion of the work referred to, in this project has been submitted in support of an application for another degree or qualification of this or any other university/institute or other institution of learning”. It is further declared that this undergraduate project, neither as a whole nor as a part thereof has been copied out from any sources, wherever references have been provided.

MEMBERS’ SIGNATURES

ACKNOWLEDGEMENTS

The success and final outcome of this project required a lot of guidance and assistance from our supervisor and other panel members and I am extremely privileged to have got this all along the completion of our project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Ms. Mammona Qudsia, for providing us an opportunity to do this project and giving us all support and guidance which made us complete the project duly. We are extremely thankful to her for providing such a nice support and guidance, although she had a busy schedule managing the classes load.

I am thankful and fortunate enough to get constant encouragement, support and guidance from all project panels which helped us in successfully completing our project work. Our panel gave us great suggestions to complete our project efficiently in time.

DEDICATION

I dedicate this project to God Almighty my creator, my strong pillar, my source of inspiration, wisdom, knowledge and understanding. He has been the source of my strength throughout this program and on His wings only have I soared. Ms. Mammona Qudsia has been the ideal project supervisor. Her sage advice, insightful criticisms, and patient encouragement aided creation of project in innumerable ways.

Executive Summary

Gangster's war game is really a futuristic game that can playable being online and offline at same time. The player's main aim will be to get maximum kills to win the game. There are two different modes in which any player can play game which are online and offline. As online shows it can only be playable when a player have active internet connection and at same time offline mode can be played on normal PC. So while going in online mode the player can create/Join server the server, either when first player goes he has to create server to add his friends or may be any player from globe in his server which he created. If some other player join game he/she can join the already created server. After joining of the server there new step comes to select player category from given two types as Assault and SWAT, which have different guns. All required information will be in on top left side of playing screen. While playing in other mode which don't required to be online compulsory. In offline mode the player have to kill all the bots walking in the map there is different map than online one. The player have to kill all the bots (computer based players) to win the game.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Introduction

This chapter include a basic information about a Desktop online game called "Gangster War". Basically, this will work online. All players will must be connected with internet to join game. All players will fight with each other. The Player who has highest death rate will lose. Any player whose power gone to zero will be respawned. Match is dynamic (When all players left game) game will end. There are multiple players can fight with each other using guns. The project work breakdown and the proposed time to finish different modules of the project have been included in this chapter.

While playing game any player can select from two given modes which he/she want to play as Online and Offline mode. As name of Online shows it can only be play able having the working internet connection to connect with cloud and can only connected to other players using internet. So, when player wants to play online mode he/she must have to create the server for other players to join and can play in multi-player mode across globe. While one player created server other players just have to join that created server just by clicking its name given on Create Server screen. After joining of the server player have to select from given categories as SWAT and Assault. Any player whose power gone zero will be respawned. HP (Health points), Scores, Death and Ammo will be shown on top right of screen. Player can control movement of character. Health Score also displayed on enemy head. Player can use multiple guns such as CZ-805, SVD (Sniper) and MP05.

1.2 Objective

All players who will play the game, their main aim will be to win the match. The win and lose will be on the basis of maximum number of kills in online mode. Any Player can shoot all other players who are playing in the server. Player can use Gun to shoot other players. Every bullet shot can damage the 5%, 80%, and 40% to the health point of player.

1.3 Features

- Respawn
    - Any player whose power gone zero will be respawned.

- Design Characters
    - We will design a character for game.
- Fighting Riffles
    - We will be use built-in texture for guns.
- Score Manage

  Two Types of Score Manager
    - If player die then Death rate increase by one
    - When bullet hit enemy Score rate increase by one
- Server Create
    - Server creation through internet.
    - The minimum player limit is 1.
- Health Points
    - Health point 5% decreases on one bullet by Cz805.
    - Health point 80% decreases on one bullet by Sniper SVD.
    - Health point 40% decreases on one bullet by MP05.
- Map Designing
    - Containers

      The container is used for defense from opponent. Player will be hide behind the container for safety.
    - Desert

      The Ground is like as a Desert.
    - Houses
        - Double story
        - Single story with one door.
        - Single story with two doors.
    - Tunnels

- Swings
- Mountains
- Bridge
- Trees
- Fence wall

➢ Animation:

- Walk
- Run
- Idle
- Jump
- Firing

1.4 Existing Examples / Solutions

There are a number of games developed which can be considered as existing examples for over projects. The Comparison is shown in below table.

Table 1.1 Existing Examples

Serial No	Features	CRAZY PIXEL	MODERN BLOCKY PAINT	Gangster's war
1	Available On	Desktop	Desktop	Desktop
2	Limited Movement	Yes	Yes	Yes
3	ADS Free	No	No	Yes
4	Time freeze	No	No	No
5	Offline	No	No	Yes
6	Respawn	Yes	Yes	Yes

1.5 Business Scope

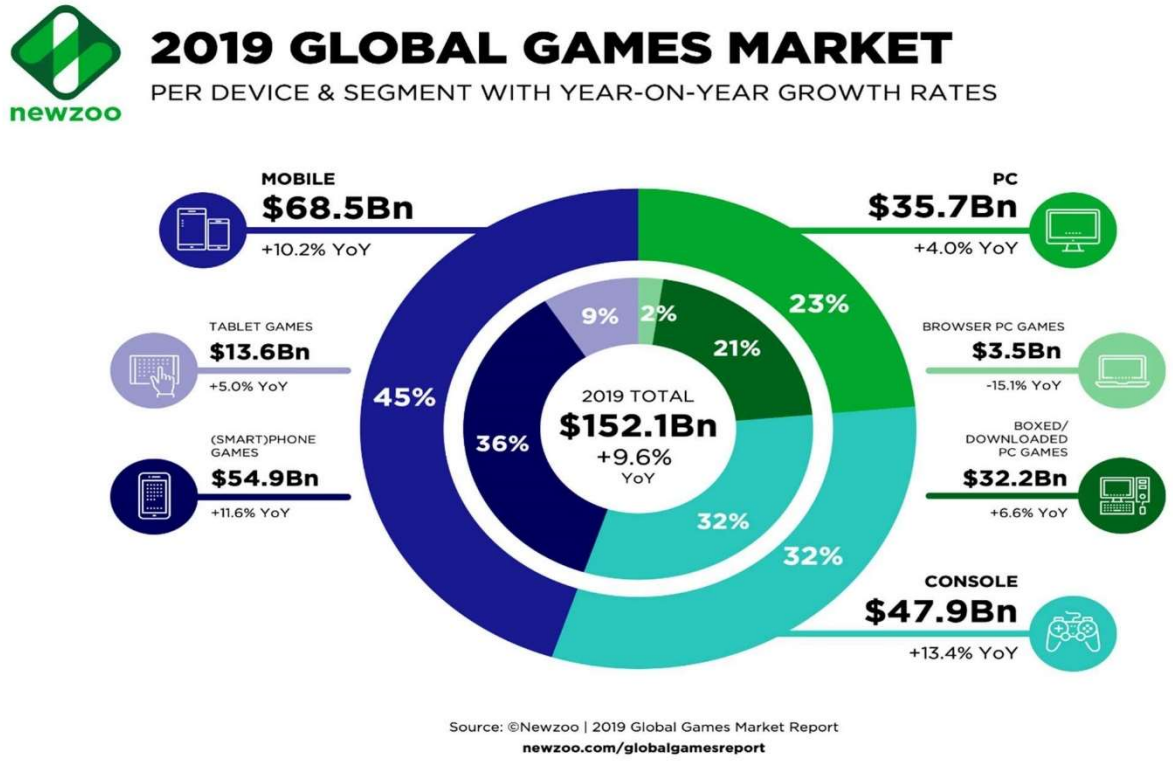

Figure 1-1 Business Scope

There are now more than 2.5 billion gamers across the world. Combined, they will spend $152.1 billion on games in 2019, representing an increase of +9.6% year on year.

Console will be the fastest-growing segment this year, growing +13.4% year on year to $47.9 billion in 2019.

Mobile gaming (smartphone and tablet), meanwhile, remains the largest segment in 2019, growing +10.2% year on year to $68.5 billion—45% of the global games market.

PC gaming will be both the smallest and slowest-growing segment, increasing +4.0% year on year to $35.7 billion.

1.6 Useful Tools and Technologies

We will be using the following tools and technologies in our project.


Unity3D game engine will used as the main development tool to develop the game.


Language will be used for the programming and the development of the game logic and functionality.


Visual studio Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.


Make human is free and open source 3D computer graphics middleware designed for the prototyping of photorealistic humanoids.


Blender is 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.

1.7 Project Work Break Down

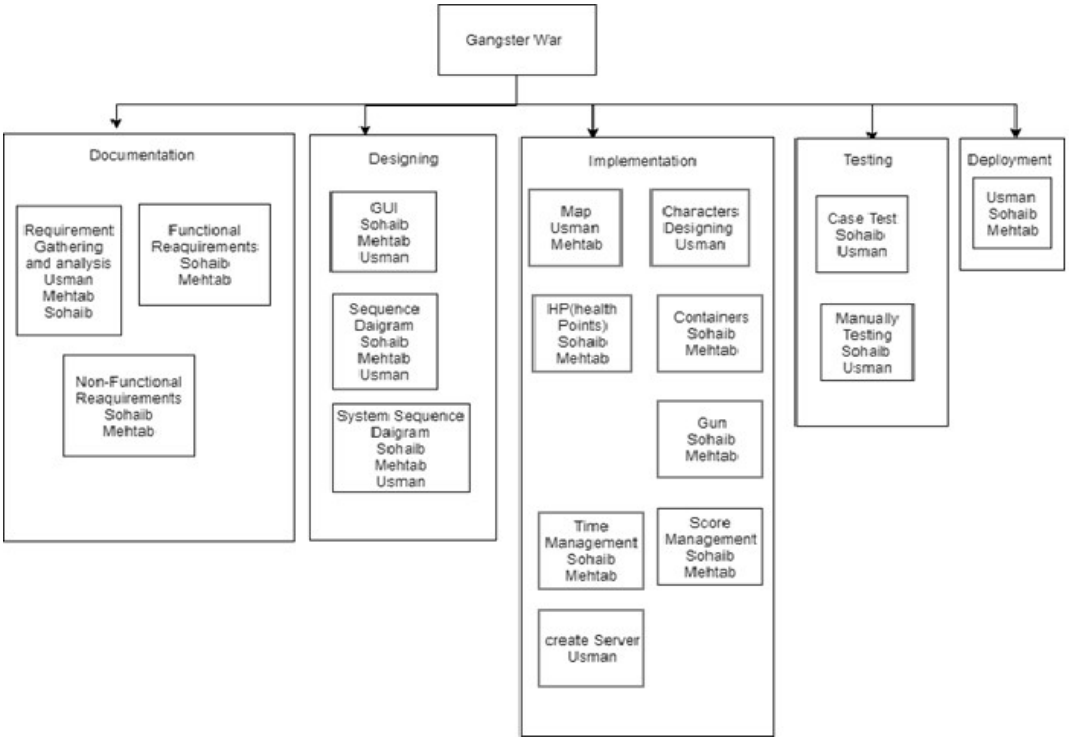

Figure 1-2: Project Work Break Down

1.8 Project Time Lime

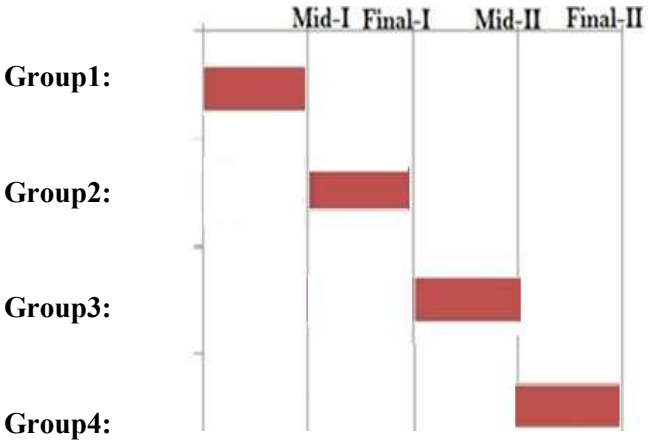

Figure 1-3: Project Timeline

Group 1:

Requirement Gathering and Analysis
Functional and non-functional requirements
Character Animation
System Sequence Diagram
Character Design

Group 2:

Map designing
Billboard
Crosshair
Guns
Containers
Desert
Tunnels
Swing
Mountains
Bridge
Houses

Group 3:

Respawn Object Demo
Reload and Gun monitor
Aiming and Shooting Target
Muzzle Flash
Bullet Holes
Pickup items

Group 4:

Respawn points
Networking
Camera Collision Fix
Deployment

Chapter 2

REQUIREMENT SPECIFICATION and ANALYSIS

This chapter includes all of the requirement specifications and their analysis of the project "Gangster's war". The main modules in this Game include the environment, characters and playing online to let players can play remotely. These components consist of multiple elements such as Container, weapon objects, enemies, controls, shooting, game menu and animation.

2.1 Functional Requirements

The Functional Requirements Specification will give a complete list of functional requirement which will be easily understandable by the customer as well as by the requirements engineer. All of the functional requirements are shown in Table 2.1.

Table 2.1 Functional Requirement

S. No.	Functional Requirement	Type	Status
1	The player should be able to start a game	Core	Completed
2	The player should be able to create the server	Core	Completed
3	Players should be able to join the server.	Core	Completed
4	The player should be able to aim while the character is holding a weapon	Core	Completed
5	The player should be able to fire the gun his character is holding gun.	Core	Completed
6	The player should be able to jump.	Core	Completed
7	The player should be able to move right.	Core	Completed
8	The player should be able to move left.	Core	Completed
9	The player should be able to backward and forward.	Core	Completed
10	The player should be able to control the movements of his character and move it freely in the map	Core	Completed

11	The player should be able to go into houses.	Core	Completed
12	The player should be able to damage opponent's health	Core	Completed
13	The player should be able to leave the server.	Core	Completed
14	The player should be able to view the score, death and ammo.	Nominal	Completed
15	The player should be able to exit the game.	Core	Completed
16	The player should be able to enter the character name	Nominal	Completed
17	The player should be able to interact with the environment (like containers, roads, tunnels and houses.)	Core	Completed
18	The player should be able to attack with gun to other players.	Core	Completed
19	The player should be able to select AI mode from options.	Core	Completed
20	The player should be able to send massage to other players.	Core	Completed

2.2 Non-Functional Requirements

Non-functional requirements include performance, scalability, availability, reliability, maintainability and usability of the system.

All of the non-functional requirements are shown in Table 2.2.

Table 2.2 Non-Functional Requirements

S. No.	Non Functional Requirements	Category
1	There will be multiple types of guns: • Cz805 ➢ damage : 5 ➢ Ammo. : 40 ➢ clip count : 1000 ➢ range : 100	System Level

	• Sniper SVD ➢ damage : 80 ➢ range : 2500 ➢ clip count: 1000 ➢ Ammo:5 • MP05 ➢ damage : 40 ➢ range : 100 ➢ clip count: 1000 ➢ Ammo:40 ➢ range 100	
2	Guns will have 1000 Clip and 1 clip have 40 or 5 bullets and require reloading after zero remaining bullet in clip	System Level
3	The player will be able to move in complete 360 degree	User Level
4	The main objective of player should be to lowest death rate.	User level.
5	The player name entered by the user will be used on character head between the games.	User Level
6	When player got shot its main screen (face view) will become bloody.	User Level
7	The timer and kill count will always be visible to the user in UI.	User Level
8	The player will be shown the number of bullets in the gun available in magazine for the gun	User Level
9	When the gun held by the player goes empty it will reloaded manually by pressing R.	User Level
10	The player will die when they completely lost 100% their health.	User Level
11	The player name can be as long as 16 characters maximum and can be as minimum as 1 character	System Level
12	While playing in the AI mode there will be bots (Computer made fake players) that will shoot at players.	System Level

Items Designed:

- Character.
- Character rigging (motion of bones).
- Desert.
- Containers.
- Houses.
- Tunnel.
- GUI (Main page, Home Page, Settings).
- One Gun.
- Billboard
- Swing
- Trees
- Fence wall.
- Crosshair.

Animations:

- Walk.
- Run.
- Idle.
- Jump.
- Firing.

Advanced Features:

- AI mode.
    - Using built-in map
    - Player
    - Bot
    - Gun
    - Bot will kill when player come in front of Bot.
    - TTP
- Death effect
- Ammo Count.

2.3 System Use Case Modeling

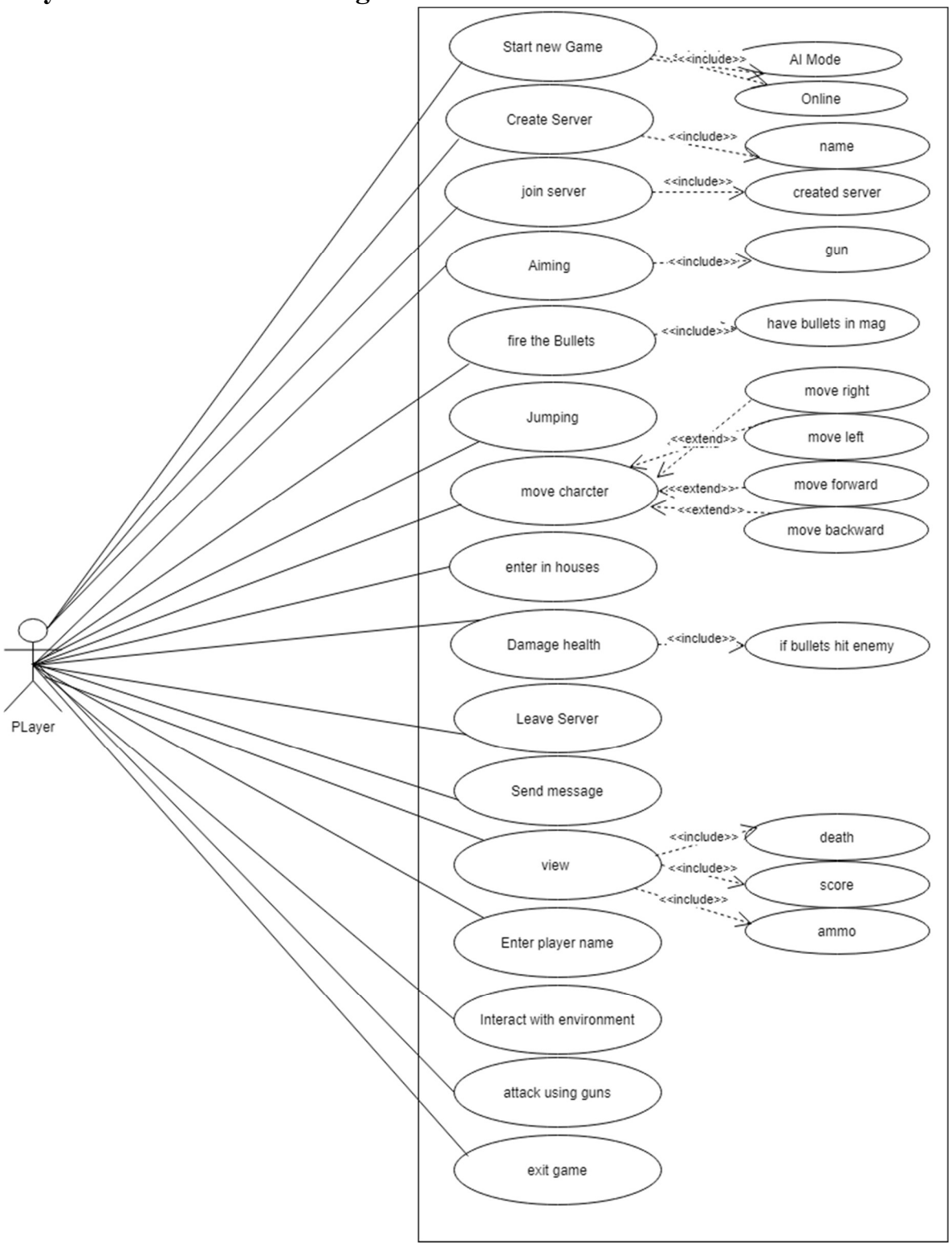

Figure 2-1: Use case Model

2.4 Use Case descriptions

Use Case 1: Start new game

Table 2.3 Use Case 1

Use Case ID:	UC01		
Use Case Name:	Start new game		
Created By:	Usman Yaqoob	**Last Updated By:**	Usman Yaqoob
Date Created:	12th October,2019	**Last Revision Date:**	10th January,2020
Actors:	User		
Description:	Starts new game and shows main menu to user.		
Trigger:	Game loaded.		
Preconditions:	Game started.		
Post conditions:	Game starts.		
Normal Flow:	Actor	System	
	1. User selects start game	2. Shows main page to join/create server.	
Alternative Flows:	The game will be started.		
Exceptions:	Halts, and restart the game		

Use Case 2: Creating server

Table 2.4 Use Case 2

Use Case ID:	UC02		
Use Case Name:	Creating server		
Created By:	Mehtab Naseer	**Last Updated By:**	Mehtab Naseer
Date Created:	17 October 2019	**Last Revision Date:**	10th January,2020
Actors:	Player		
Description:	Creates the server online can be joined by other players.		
Trigger:	Server Creation.		
Preconditions:	Application is launched		
Post conditions:	Server will be created.		
Normal Flow:	Actor	System	
	User selects create sever button.	System leads player to server creation.	
Alternative Flows:	Creating server failed/ Network Error.		
Exceptions:	Alert: Check connection to internet.		

Use Case 3: Joining the server

Table 2.5 Use Case 3

Use Case ID:	UC03		
Use Case Name:	Joining the server		
Created By:	Mehtab Naseer	**Last Updated By:**	Mehtab Naseer
Date Created:	17 October 2019	**Last Revision Date:**	10th January,2020
Actors:	Player		
Description:	The player can join the server already created.		
Trigger:	Joining the server.		
Preconditions:	Server must already create.		
Post conditions:	Joining server done.		
Normal Flow:	Actor	System	
	User selects join sever button.	System leads player to game lobby.	
Alternative Flows:	Joining server failed/ Network Error.		
Exceptions:	If not already created can opt to create server.		

Use Case 4: Right movement.

Table 2.6 Use Case 4

Use Case ID:	UC04		
Use Case Name:	Right Character.		
Created By:	Sohaib Ahmed	**Last Updated By:**	Sohaib Ahmed
Date Created:	17 October 2019	**Last Revision Date:**	10th January,2020
Actors:	Player		
Description:	When player join server, he can control character to right.		
Trigger:	Character movement as press d for right.		
Preconditions:	Server must be joined.		
Post conditions:	Player should move.		
Normal Flow:	Actor	System	
	Motion of character seen be player in the map.	System should the motion of the character to the player.	
Alternative Flows:	Show network error message.		
Exceptions:			

Use Case 5: Enter Server Name:

Table 2.7 Use Case 5

Use Case ID:	UC05		
Use Case Name:	Enter Server Name.		
Created By:	Mehtab Naseer	**Last Updated By:**	Sohaib Ahmed
Date Created:	17 October 2019	**Last Revision Date:**	10th January,2020
Actors:	Player		
Description:	When player creating server it requires server name to create.		
Trigger:	Server created.		
Preconditions:	Game must be in online mode.		
Post conditions:	Server created.		
Normal Flow:	Actor	System	
	Player enter server name.	Creation of server.	
Alternative Flows:	Add legal name.		
Exceptions:			

Use Case 6: Send messages.

Table 2.8 Use Case 6

Use Case ID:	UC06		
Use Case Name:	Send messages.		
Created By:	Sohaib Ahmed	**Last Updated By:**	Sohaib Ahmed
Date Created:	12th October,2019	**Last Revision Date:**	10th January,2020
Actors:	Player.		
Description:	Aim to send messages to other players.		
Trigger:	Messaging.		
Preconditions:	Player must already joined game, press TAB button to move to messages.		
Post conditions:	Message sent.		
Normal Flow:	Actor	System	
	Type something to send to other player	Broadcast message to all players ion server.	
Alternative Flows:	Message not sent.		
Exceptions:			

Use Case 7: Enter in houses.

Table 2.9 Use Case 7

Use Case ID:	UC07		
Use Case Name:	Enter in houses.		
Created By:	Usman Yaqoob	**Last Updated By:**	Usman Yaqoob
Date Created:	12th October,2019	**Last Revision Date:**	10th January,2020
Actors:	Player.		
Description:	When player want to enter houses present in map.		
Trigger:	Entering in house.		
Preconditions:	Player must joined game present in server, character movement.		
Post conditions:	Entered in houses.		
Normal Flow:	Actor	System	
	Go to the gate of house to enter in the house.	System must show the character to be entered in the house.	
Alternative Flows:	May collide will wall while entering in the house.		
Exceptions:			

Use Case 8: Aiming on enemies.

Table 2.10 Use Case 8

Use Case ID:	UC08		
Use Case Name:	Aim at enemies holding the gun		
Created By:	Sohaib Ahmed	**Last Updated By:**	Sohaib Ahmed
Date Created:	12th October,2019	**Last Revision Date:**	10th January,2020
Actors:	User		
Description:	Aim on enemies with gun to shoot at them.		
Trigger:	Aiming on enemies.		
Preconditions:	Player must already joined game.		
Post conditions:	Gun pointer aims to enemies.		
Normal Flow:	Actor	System	
	1. User aim on enemies with gun	1. Gun pointer moves to enemies.	
Alternative Flows:	Gun pointer remains on position		
Exceptions:			

Use Case 9: Firing bullets

Table 2.11 Use Case 9

Use Case ID:	UC09		
Use Case Name:	Fire the bullets.		
Created By:	Usman Yaqoob	**Last Updated By:**	Usman Yaqoob
Date Created:	12th October,2019	**Last Revision Date:**	10th January,2020
Actors:	User		
Description:	Fire the bullet to enemies to damage health.		
Trigger:	Firing bullets.		
Preconditions:	Game must already started.		
Post conditions:	Enemies got damage if they got hit by shot.		
Normal Flow:	Actor	System	
	1. User fire with gun	1. Enemies get damaged.	
Alternative Flows:	Gun status remains default.		
Exceptions:			

Use Case 10: Enter player name.

Table 2.12 Use Case 10

Use Case ID:	UC10		
Use Case Name:	Enter player name.		
Created By:	Usman Yaqoob	**Last Updated By:**	Usman Yaqoob
Date Created:	12th October,2019	**Last Revision Date:**	10th January,2020
Actors:	Player.		
Description:	When player have to create server it must give his name to join game.		
Trigger:	Enter name		
Preconditions:	Game must be started.		
Post conditions:	Name settled as entered by user.		
Normal Flow:	Actor	System	
	Enter player by user through keyboard.	Name shown when player killed and other player on left down side.	
Alternative Flows:	Enter valid name.		
Exceptions:			

Use Case 11: Damage health.

Table 2.13 Use Case 11

Use Case ID:	UC11		
Use Case Name:	Damage of other player's health.		
Created By:	Usman Yaqoob	**Last Updated By:**	Usman Yaqoob
Date Created:	12th October,2019	**Last Revision Date:**	10th January,2020
Actors:	User		
Description:	When players shoot to enemies will get damage.		
Trigger:	Damage health.		
Preconditions:	Enemies must be shouted by player and hit the enemy's body.		
Post conditions:	Damage the health of player who got shot by 5%.		
Normal Flow:	Actor	System	
	1. Shoot at enemies.	1. Reduce enemy health. 2 Decrease bullet from magazine.	
Alternative Flows:	Shot missed.		
Exceptions:			

Use Case 12: View Banners:

Table 2.14 Use Case 12

Use Case ID:	UC12		
Use Case Name:	View Banners.		
Created By:	Mehtab Naseer	**Last Updated By:**	Mehtab Naseer
Date Created:	12th October,2019	**Last Revision Date:**	12th October,2019
Actors:	Player.		
Description:	The player can see Death (how many times got killed), score (how many players killed) and Ammo (number of bullets left).		
Trigger:	View banner.		
Preconditions:	Banner shown to top right corner.		
Post conditions:	Banner shown to every player.		
Normal Flow:	Actor	System	
	Can view Banner as Death, Score and Ammo	Allow to view by player.	

Use Case 13: Select AI mode.

Table 2.15 Use Case 13

Use Case ID:	UC13		
Use Case Name:	Select AI mode.		
Created By:	Mehtab Naseer	**Last Updated By:**	Mehtab Naseer
Date Created:	16 April 2020	**Last Revision Date:**	16 April 2020
Actors:	Player		
Description:	The player can select the AI mode.		
Trigger:	AI mode.		
Preconditions:	Game running.		
Post conditions:	AI mode joined.		
Normal Flow:	Actor	System	
	User selects AI Mode button.	System leads player to AI mode.	
Alternative Flows:	Try again to join.		
Exceptions:	N/A		

Use Case 14: Exit game.

Table 2.16 Use Case 14

Use Case ID:	UC14		
Use Case Name:	Exit game.		
Created By:	Usman Yaqoob	**Last Updated By:**	Usman Yaqoob
Date Created:	12th October,2019	**Last Revision Date:**	12th October,2019
Actors:	User		
Description:	Exit the game.		
Trigger:	Leaving Game.		
Preconditions:	Game must already launched.		
Post conditions:	Game turned off.		
Normal Flow:	Actor	System	
	1.press exit button	1. Stopped the game Application.	
Alternative Flows:	None.		
Exceptions:			

Use Case 15: Jumping.

Table 2.17 Use Case 15

Use Case ID:	UC15		
Use Case Name:	Jumping		
Created By:	Usman Yaqoob	**Last Updated By:**	Usman Yaqoob
Date Created:	12th October,2019	**Last Revision Date:**	12th October,2019
Actors:	User		
Description:	While press SPACE to jump		
Trigger:	Jumping		
Preconditions:	Player must already joined game.		
Post conditions:	Jumping done successful		
Normal Flow:	Actor	System	
	1.press SPACE button	Show jumping of character to player.	
Alternative Flows:	Not jumped.		
Exceptions:			

Use Case 16: Interact with environment.

Table 2.18 Use Case 16

Use Case ID:	UC16		
Use Case Name:	Interact with environment		
Created By:	Sohaib Ahmed	**Last Updated By:**	Sohaib Ahmed
Date Created:	15th June,2020	**Last Revision Date:**	15th June,2020
Actors:	User		
Description:	When player joined game and can move freely in map, there he/she can interact with different of items like trees, tunnel, Fence etc.		
Trigger:	Interaction with environment.		
Preconditions:	Player must already joined game.		
Post conditions:	Successful viewing to player and move freely there.		
Normal Flow:	Actor	System	
	While moving he/she can interact with different things	Show the things present in the map.	
Alternative Flows:	May be server down or lost internet connection.		
Exceptions:			

Use Case 17: Attacking using guns

Table 2.19 Use Case 17

Use Case ID:	UC17		
Use Case Name:	Attacking using guns		
Created By:	Usman Yaqoob	**Last Updated By:**	Usman Yaqoob
Date Created:	15th Feb,2020	**Last Revision Date:**	15th Feb,2020
Actors:	User		
Description:	Player can shot to other players who joined the game using different type of guns as stated as Cz805, Sniper SVD and MP05		
Trigger:	Damage the health of player whom shots got hit.		
Preconditions:	Player must join server and other player must be in the server in order to get show when one player shoot at them.		
Post conditions:	Successful damaging the health of player.		
Normal Flow:	Actor	System	
	Shot the fire.	Decrease health of player who got hit.	
Alternative Flows:	Shot may be missed.		
Exceptions:			

Use Case 18: Left movement.

Table 2.20 Use Case 18

Use Case ID:	UC18		
Use Case Name:	Moving Character.		
Created By:	Sohaib Ahmed	**Last Updated By:**	Sohaib Ahmed
Date Created:	17 October 2019	**Last Revision Date:**	10th January,2020
Actors:	Player		
Description:	When player join server, he can control character to left.		
Trigger:	Character movement as press a for left .		
Preconditions:	Server must be joined.		
Post conditions:	Player should move.		
Normal Flow:	Actor	System	
	Motion of character seen be player in the map.	System should the motion of the character to the player.	
Alternative Flows:	Show network error message.		
Exceptions:			

Use Case 19: Forward movement.

Table 2.21 Use Case 19

Use Case ID:	UC19		
Use Case Name:	Moving Character.		
Created By:	Sohaib Ahmed	**Last Updated By:**	Sohaib Ahmed
Date Created:	17 October 2019	**Last Revision Date:**	10th January,2020
Actors:	Player		
Description:	When player join server, he can control character to forward.		
Trigger:	Character movement as press w for forward.		
Preconditions:	Server must be joined.		
Post conditions:	Player should move.		
Normal Flow:	Actor	System	
	Motion of character seen be player in the map.	System should the motion of the character to the player.	
Alternative Flows:	Show network error message.		
Exceptions:			

Use Case 20: Backward movement.

Table 2.22 Use Case 20

Use Case ID:	UC20		
Use Case Name:	Moving Character.		
Created By:	Sohaib Ahmed	**Last Updated By:**	Sohaib Ahmed
Date Created:	17 October 2019	**Last Revision Date:**	10th January,2020
Actors:	Player		
Description:	When player join server, he can control character to right, left, forward and backward.		
Trigger:	Character movement as press s for backward.		
Preconditions:	Server must be joined.		
Post conditions:	Player should move.		
Normal Flow:	Actor	System	
	Motion of character seen be player in the map.	System should the motion of the character to the player.	
Alternative Flows:	Show network error message.		
Exceptions:			

2.5 System Sequence Diagram

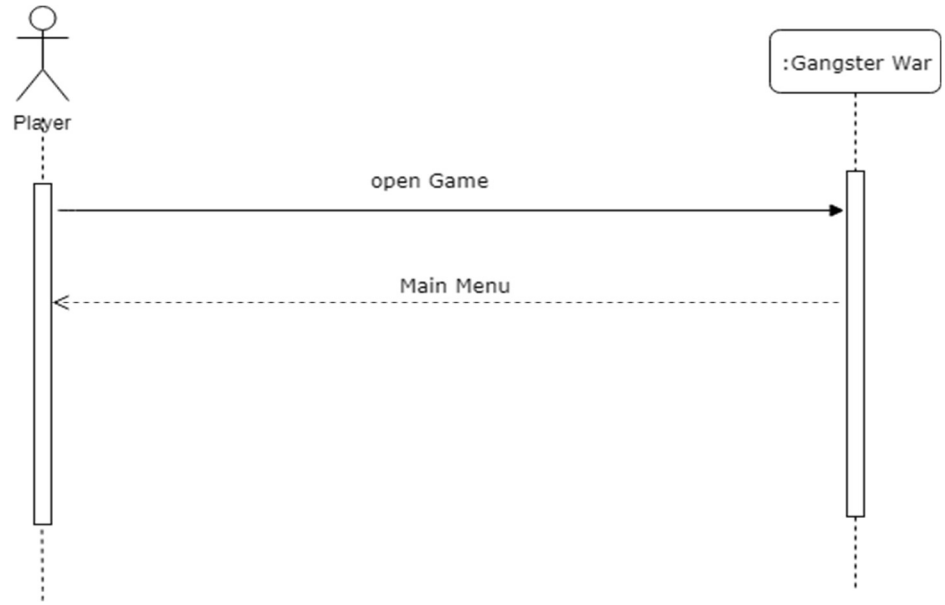

Figure 2-2: SSD (Open Game)

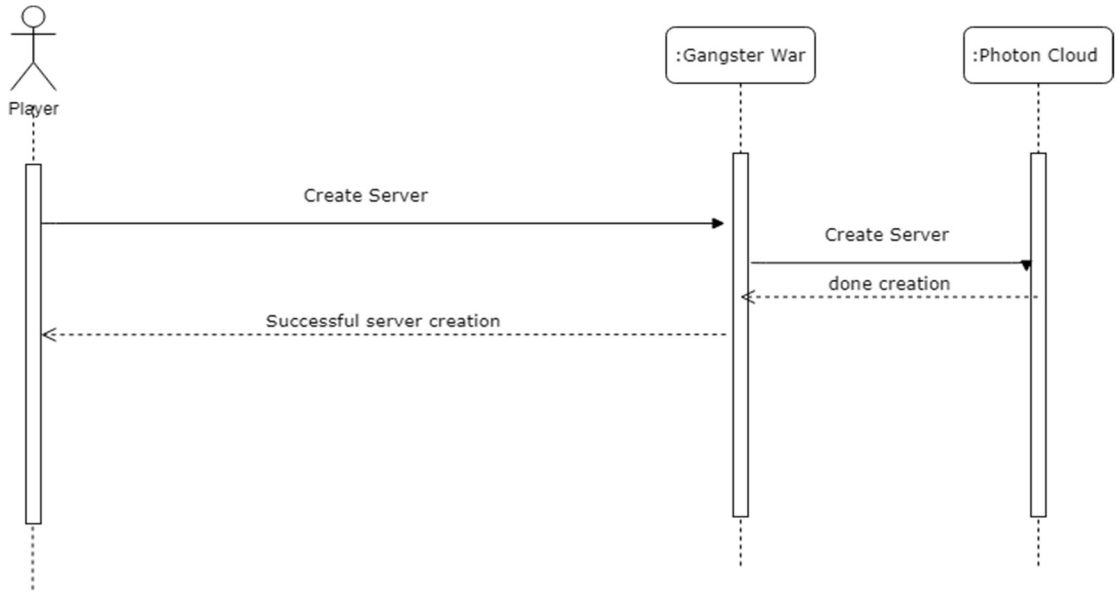

Figure 2-3: SSD (Create Server)

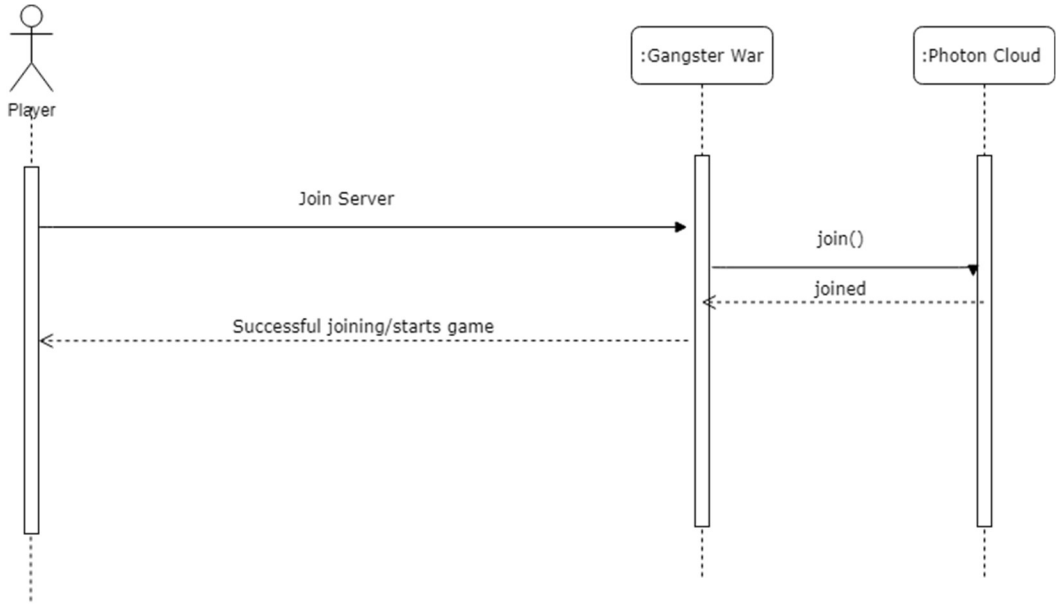

Figure 2-4: SSD (Join Server)

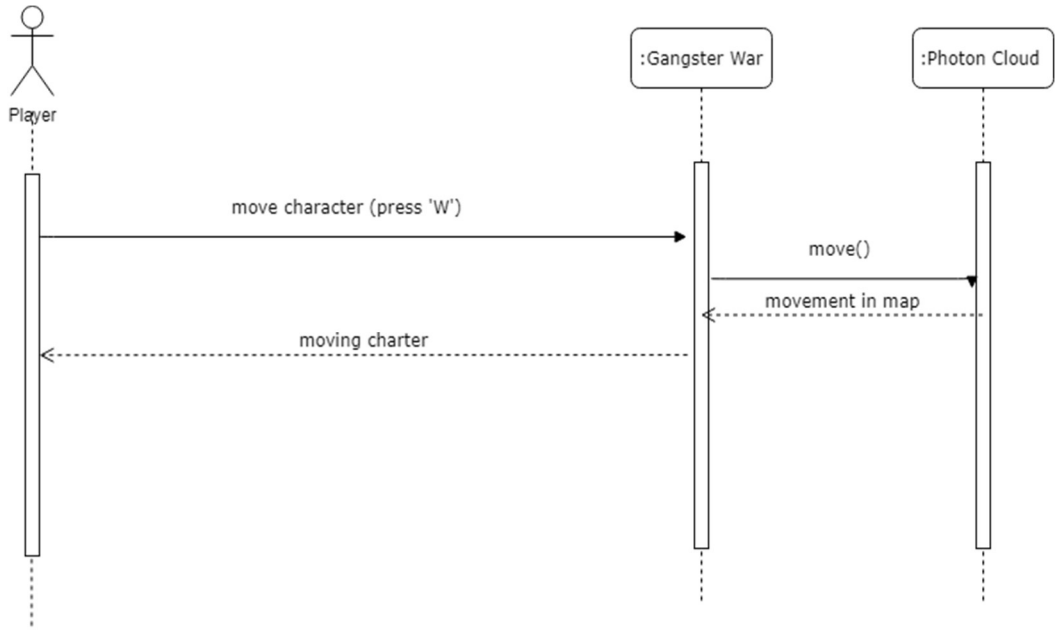

Figure 2-5: SSD (Moving Character forward)

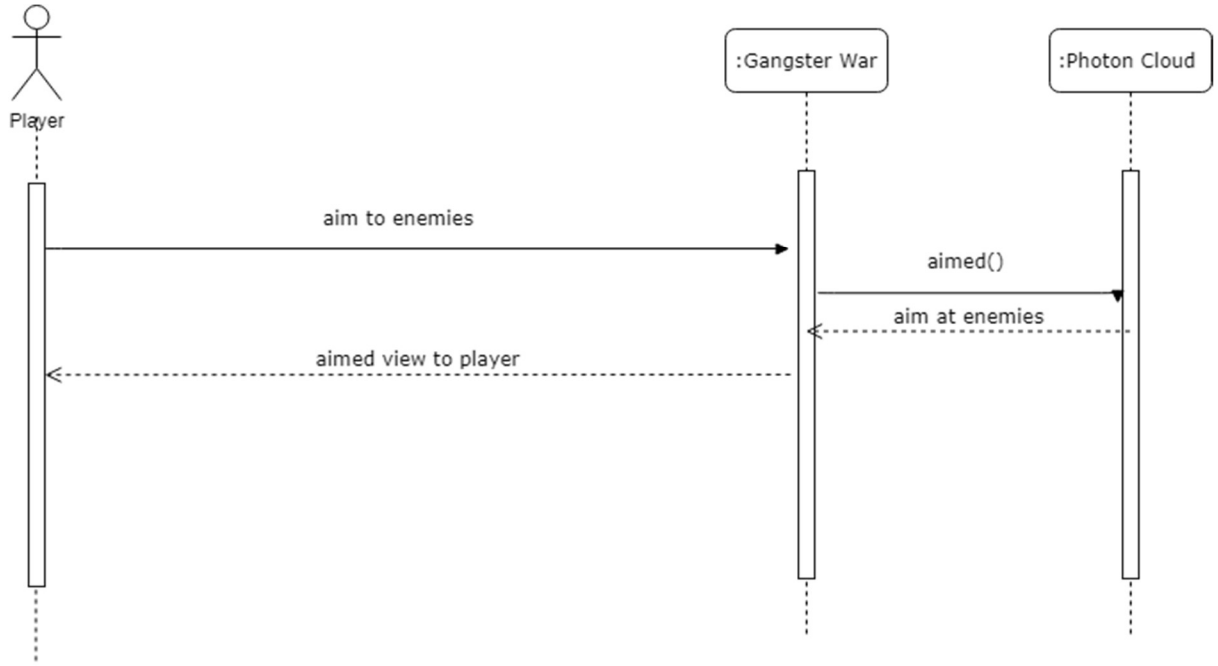

Figure 2-6: SSD (Aiming At Enemies)

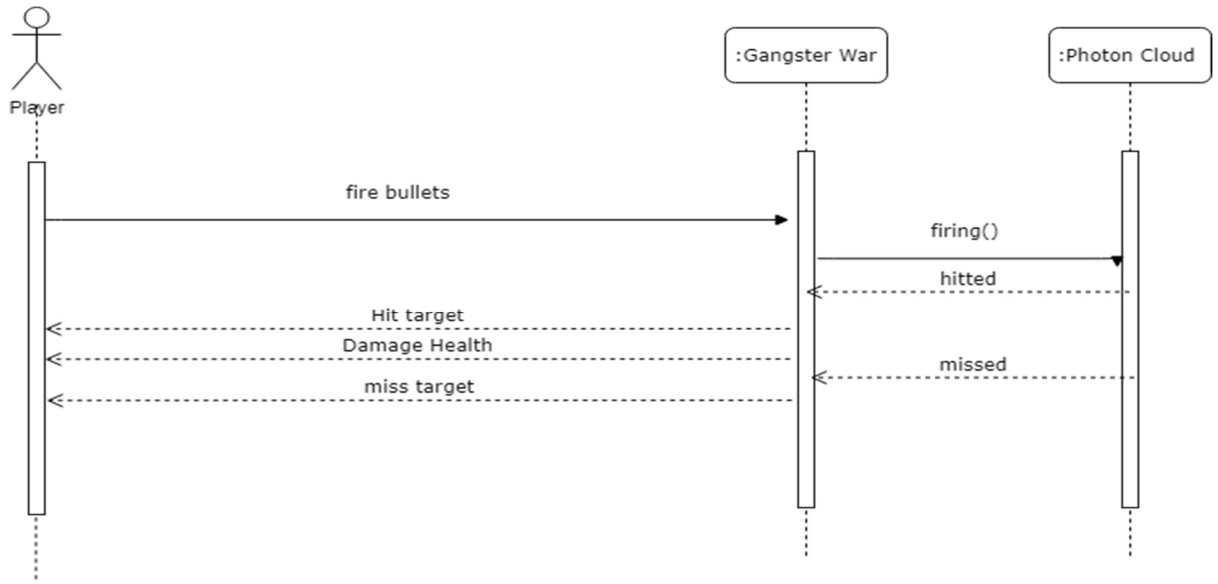

Figure 2-7: SSD (Firing At Enemies)

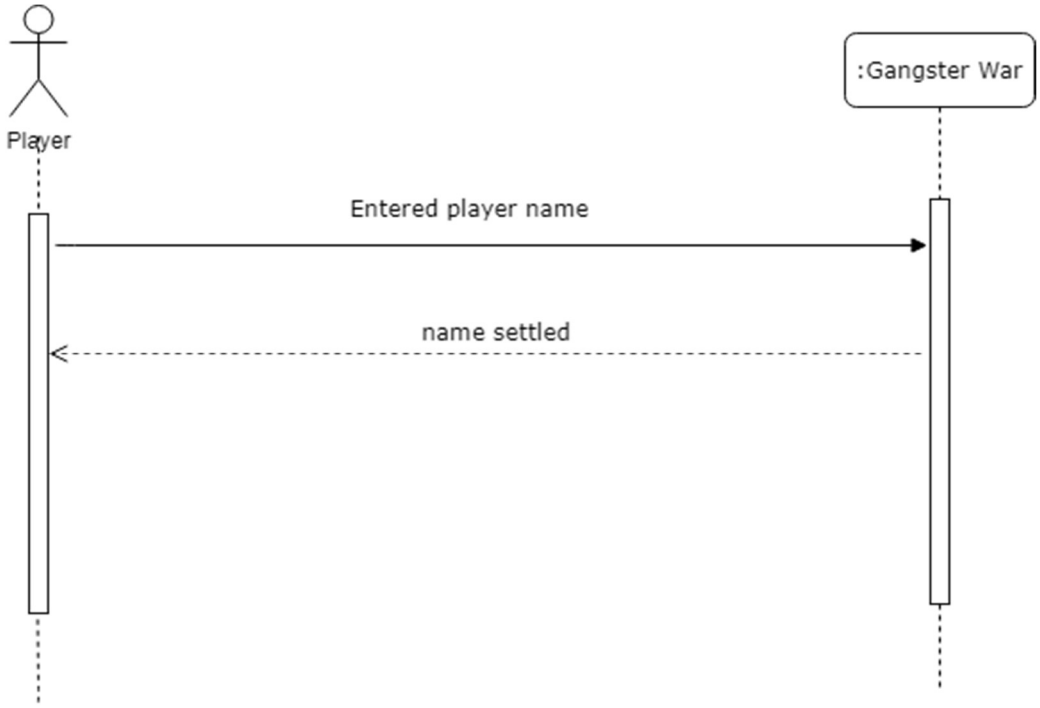

Figure 2-8: SSD (Enter Player Name)

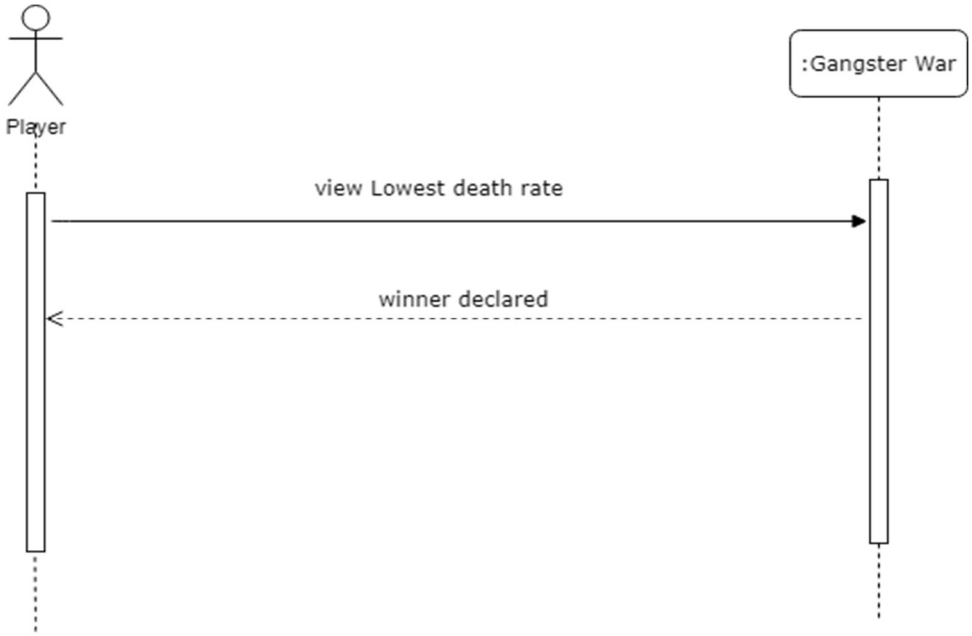

Figure 2-9: SSD (View Lowest death)

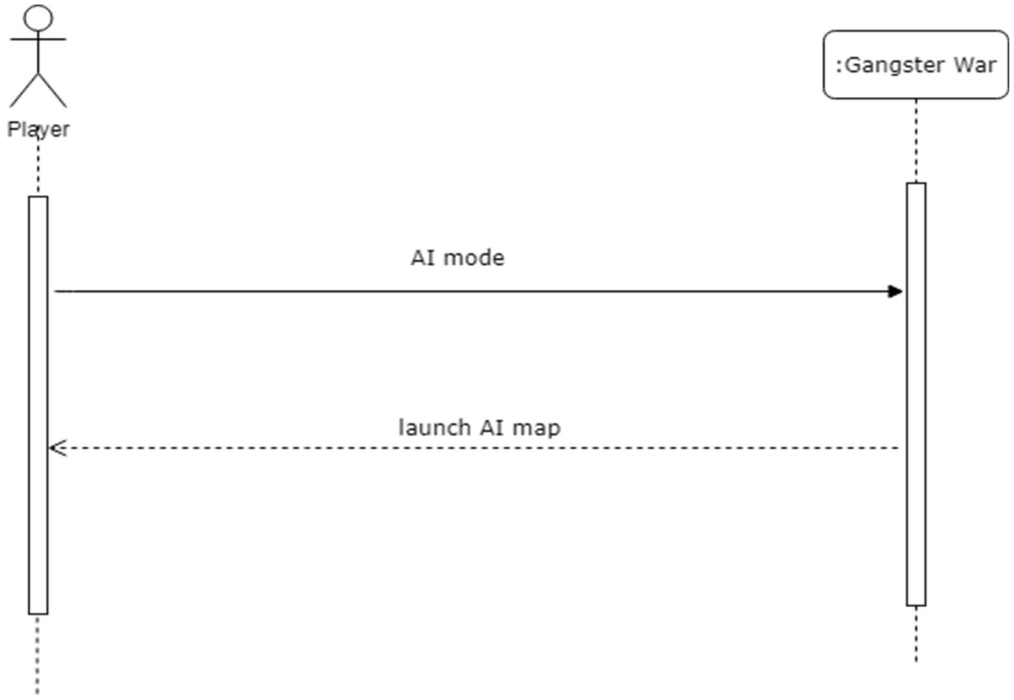

Figure 2-10: SSD (AI Mode)

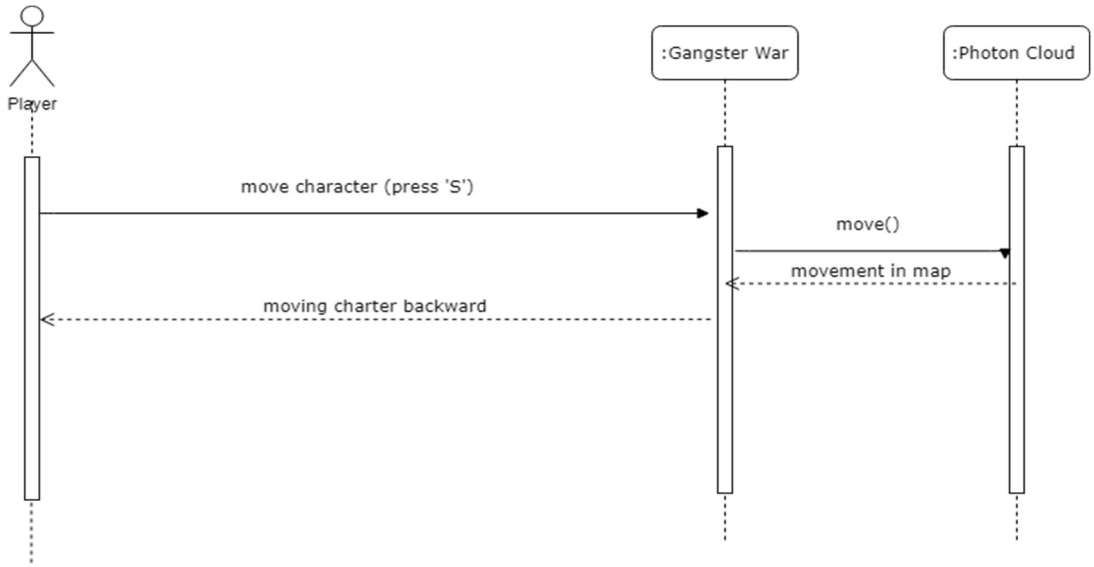

Figure 2-11: SSD (Move Backward)

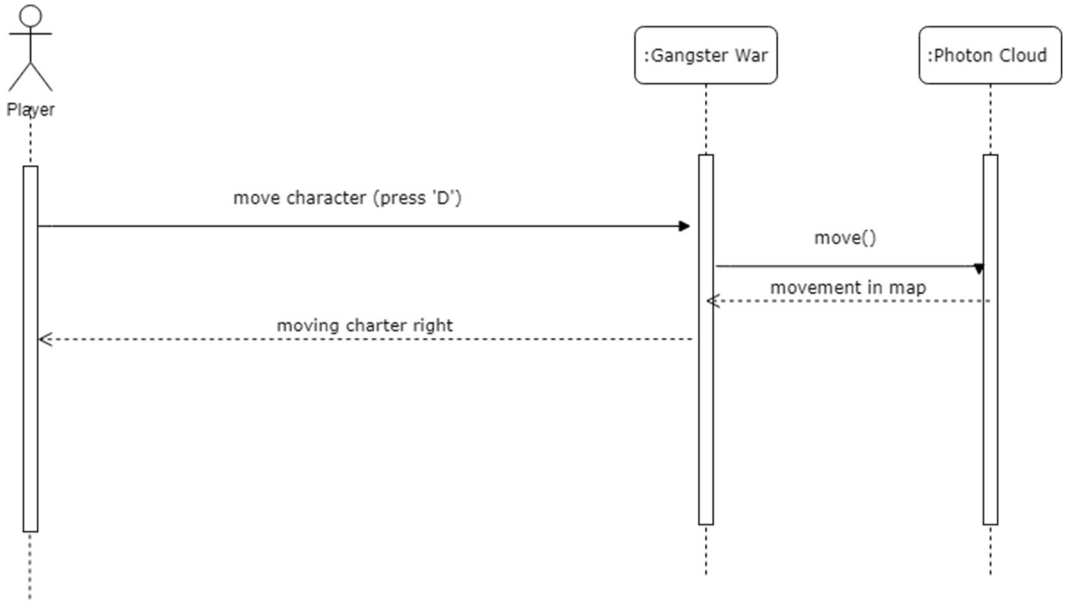

Figure 2-12: SSD (Move Right)

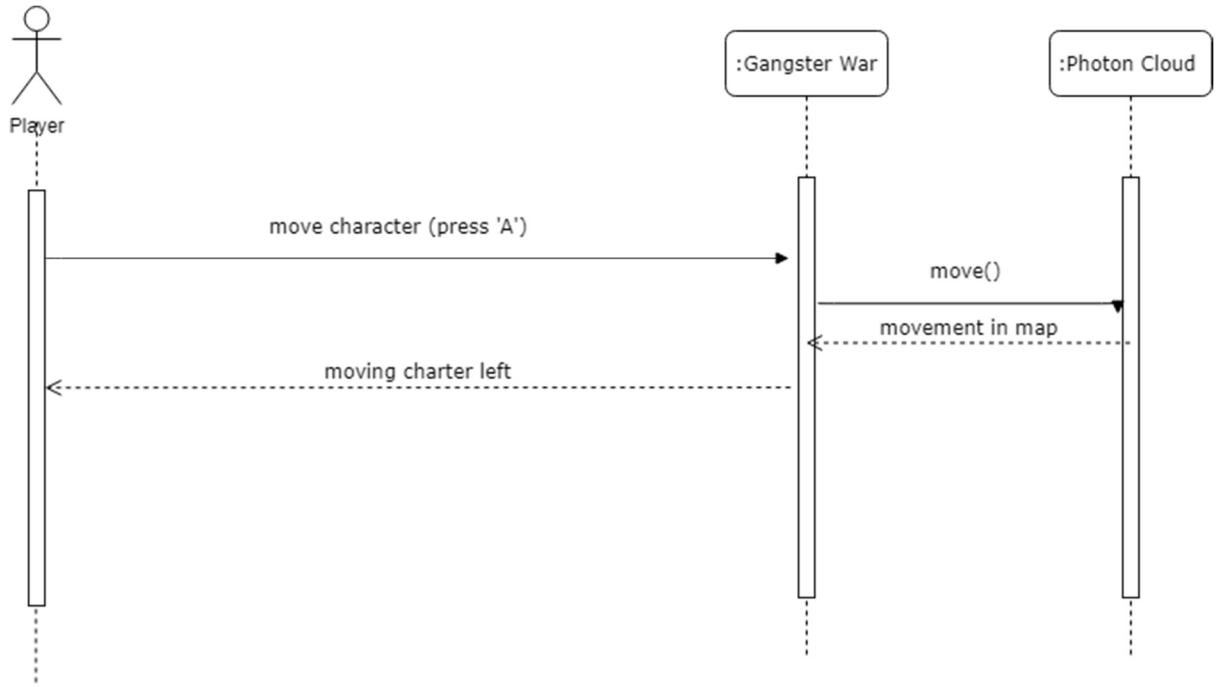

Figure 2-13: SSD (Move Left)

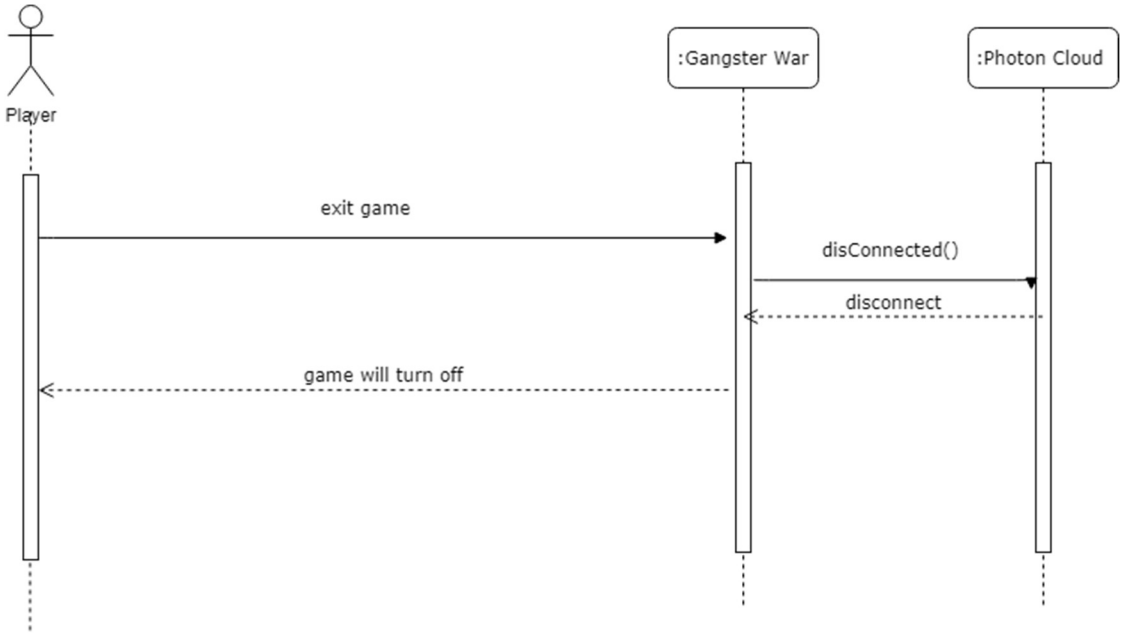

Figure 2-14: SSD (Exit Game)

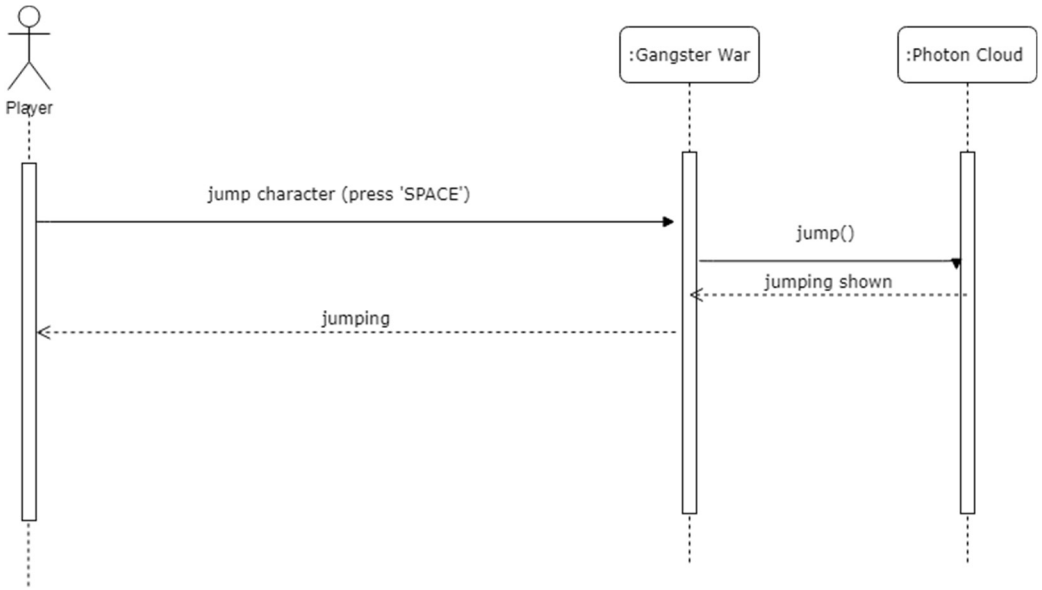

Figure 2-15: SSD (Jumping)

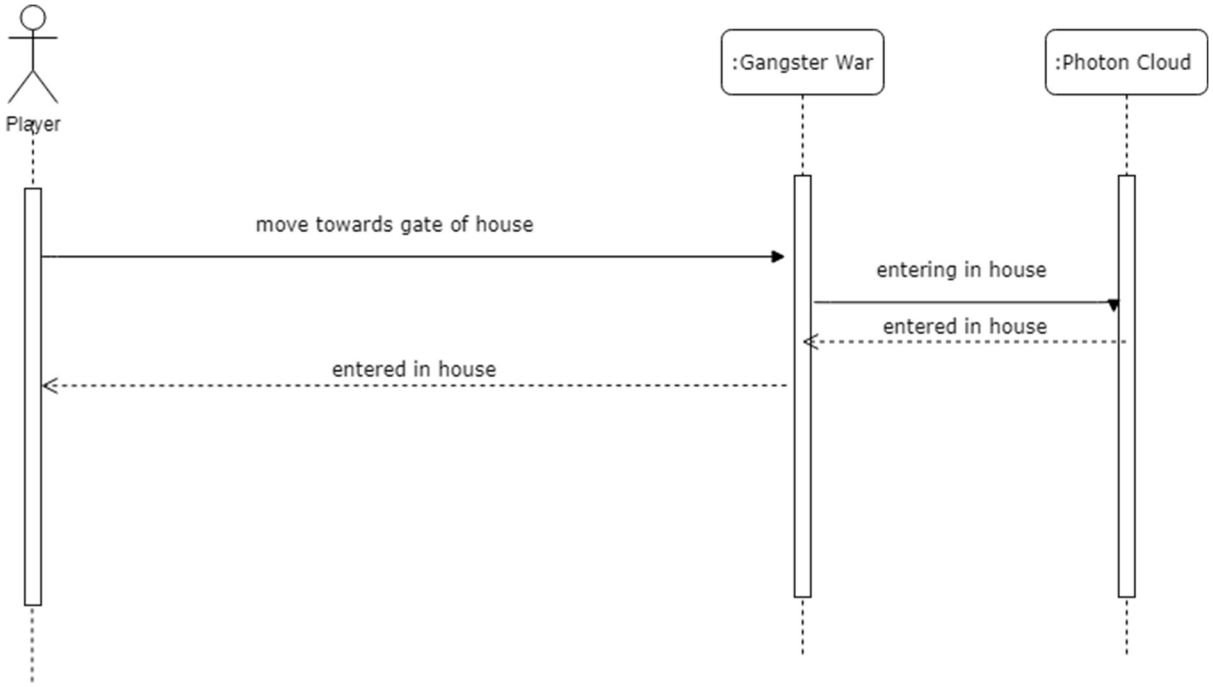

Figure 2-16: SSD (Enter In House)

.

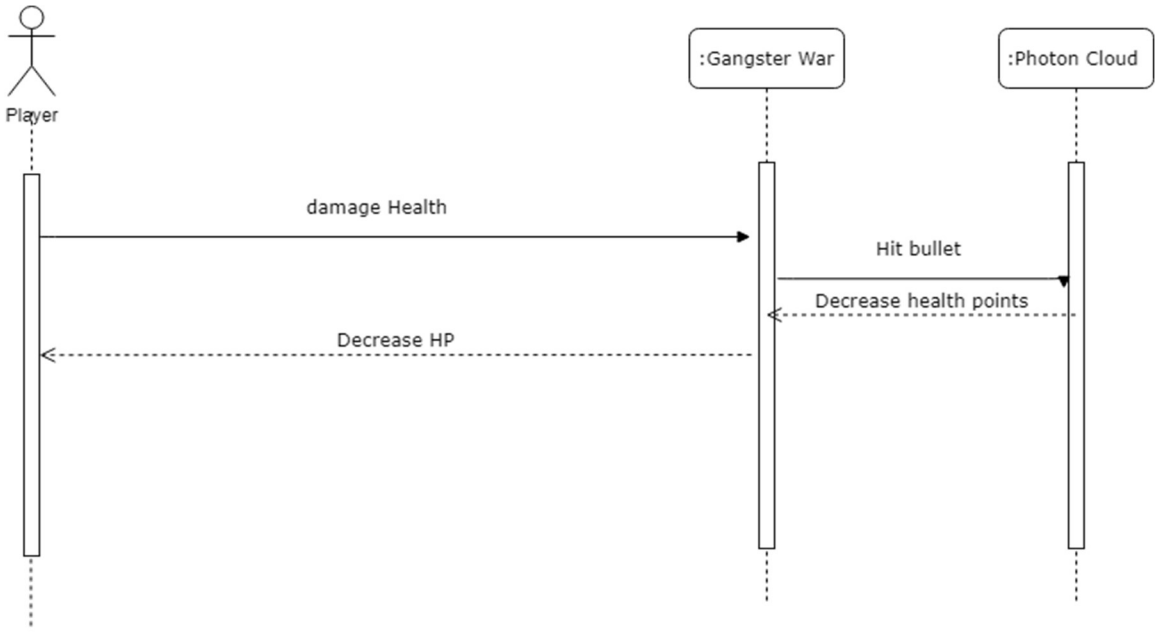

Figure 2-17 SSD (Damage Health)

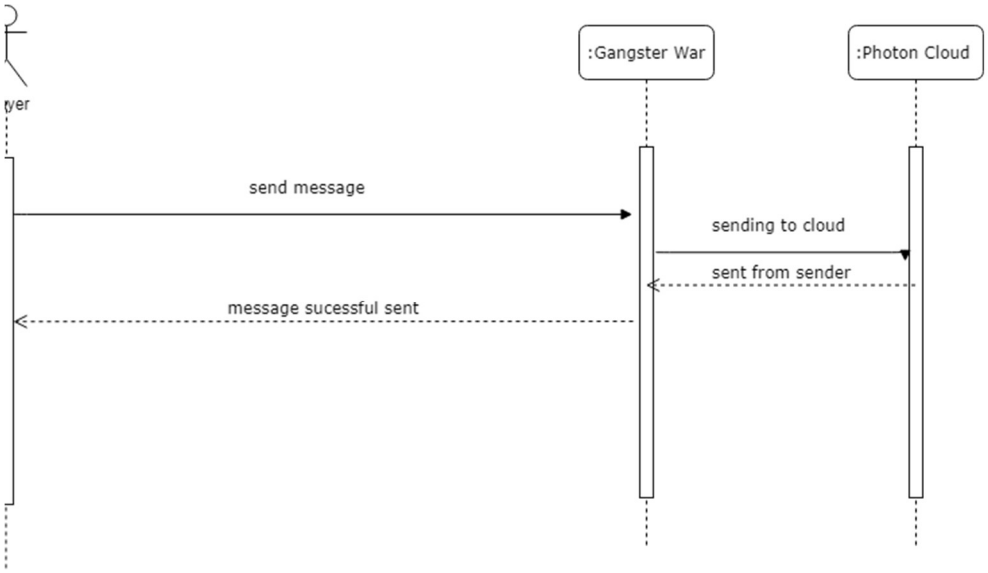

Figure 2-18 SSD (Sending Message)

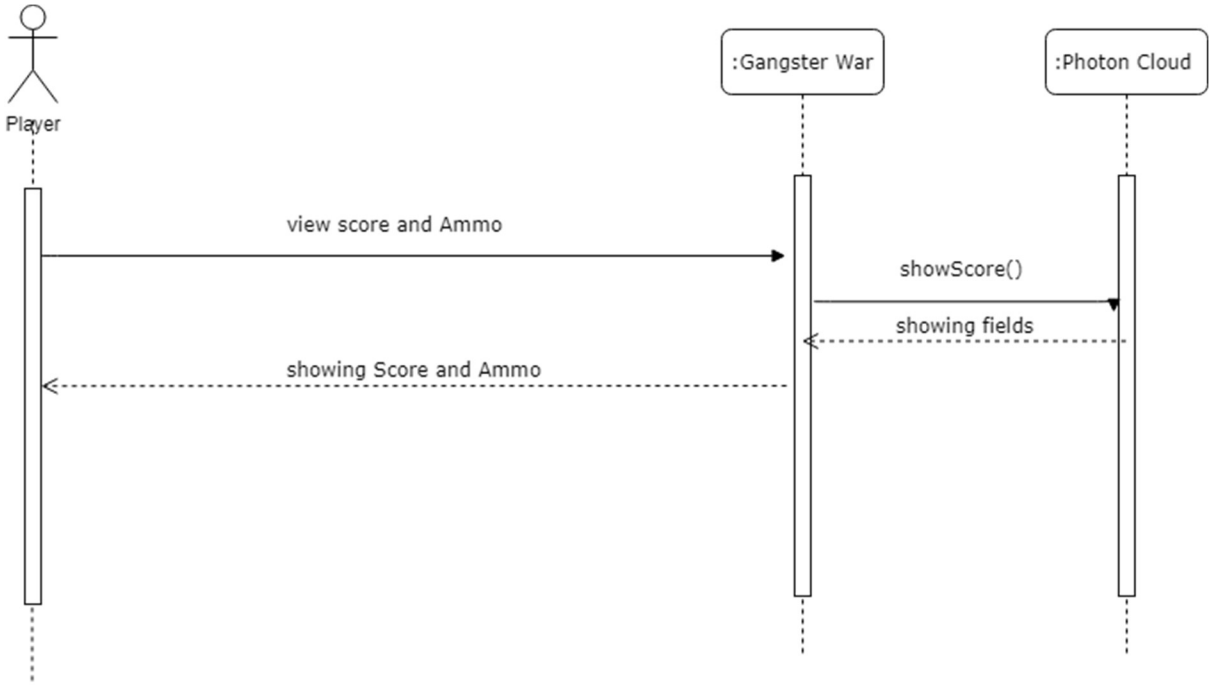

Figure 2-19 SSD (Viewing Score and Ammo)

Chapter 3

System Design

3.1 Software Architecture

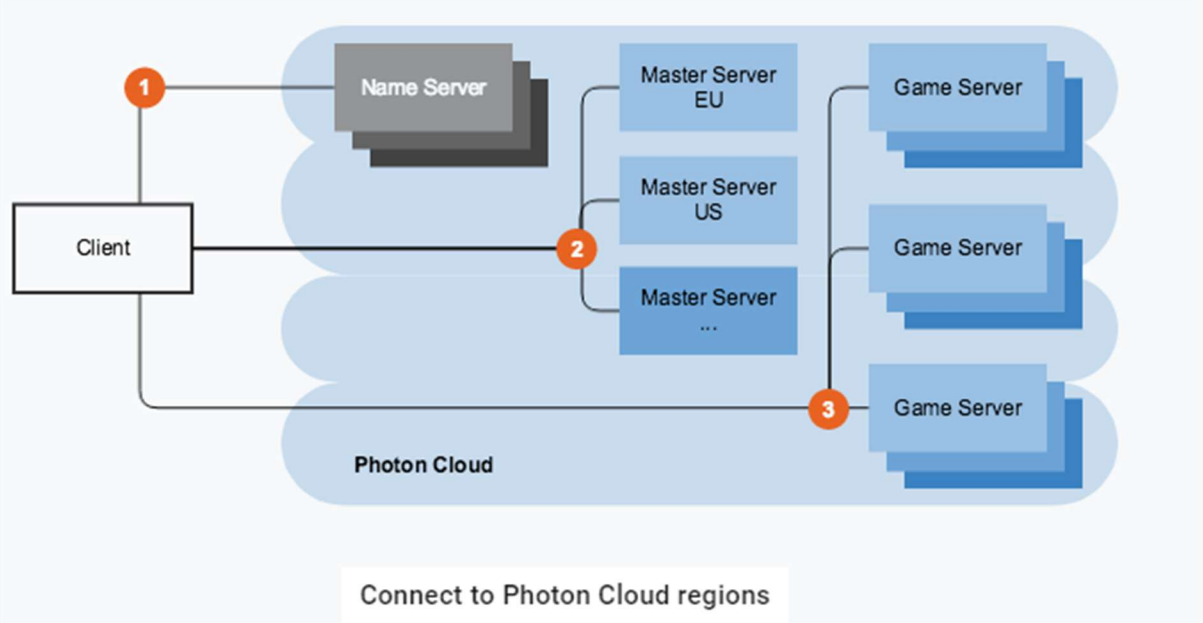

Figure 3-1: Software Architecture Diagram

3.2 User Interface Design

1. Main Menu:

Figure 3-2 Main Menu

Description:

This screen will be shown when player open the game for the first time.

2: Controls:

Figure 3-3 Controls interface

Description:

This interface will be shown when User clicked on the control option in the setting interface. User can change controls by editing and entering new controls.

3: House:

Figure 3-4 House

Description:

These houses shown in the map in which the player can go inside and outside.

4: Board:

Figure 3-5 Board

Description:

Bill boards shown in map having the logo of "Gangster's war".

5: Tunnels:

Figure 3-6 Tunnel

Description:

The character can move in and out of the tunnel.

6: swing

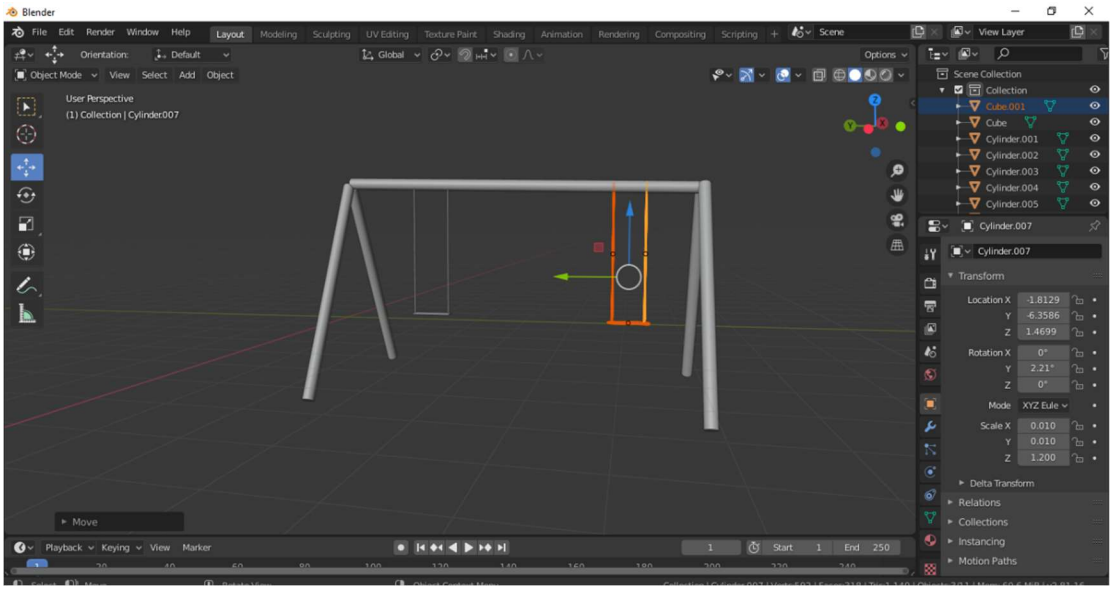

Figure 3-7 Swing

Description:

Swing will show in the map but cannot be used by the player in the game.

Chapter 4
Software Development

4.1 Coding Standards

- Meaningful and descriptive words will be used while coding the project.
- Variables and method names will be concatenated all in one if there would be several words against a name.
- Lowercase will be used for the first word and each next word would be starting from uppercase.

4.2 Development Environment

Unity, Visual Studio, Make Human and Blender are the tools that used in the project.

- Gangster's war Game will be online that will able to play across the globe.
- C# programming is used during making this project so of the code is given below.

4.3 Software Description

- **Cameras:**

```
Using System. Collections;
Using System.Collections.Generic;
Using Unity Engine;

public class ThirdPersonCamera : MonoBehaviour
{
   [System.Serializable]
   public class CameraRig
   {
      public Vector3 CameraOffset;
      public float CrouchHeight;
      public float Damping;

   }

   // 0.7 8.0 -7
   // 5
   [SerializeField]
```

```
    CameraRig defaultCamera;

    [SerializeField]
    CameraRig aimCamera;

    Transform cameraLookTarget;
   Player localPlayer;

    void Awake(){
        GameManager.Instance.OnLocalPlayerJoined += HandleOnLocalPlayerJoined;
    }
    void HandleOnLocalPlayerJoined(Player player)
    {
        localPlayer = player;
        cameraLookTarget = localPlayer.transform.Find("AimPivot");

        if (cameraLookTarget == null)
            cameraLookTarget = localPlayer.transform;
    }
    void LateUpdate()
    {
        if (localPlayer == null)
            return;

        CameraRig cameraRig = defaultCamera;

        if (localPlayer.PlayerState.WeaponState == PlayerState.EWeaponState.AIMING ||
localPlayer.PlayerState.WeaponState == PlayerState.EWeaponState.AIMEDFIRING)
            cameraRig = aimCamera;


        float targetHeight = cameraRig.CameraOffset.y + (localPlayer.PlayerState.MoveState ==
PlayerState.EMoveState.CROUCHING ? cameraRig.CrouchHeight : 0);

        Vector3 targetPosition = cameraLookTarget.position + localPlayer.transform.forward *
cameraRig.CameraOffset.z +
            localPlayer.transform.up * targetHeight +
            localPlayer.transform.right * cameraRig.CameraOffset.x;

        Vector3 collisionDestination = cameraLookTarget.position + localPlayer.transform.up * targetHeight -
localPlayer.transform.forward  *.5f ;

       // print("Target Position   :" + targetPosition);
       // print("Collision destiation   :" +collisionDestination);
      //  print(collisionDestination);
        Debug.DrawLine(targetPosition, collisionDestination, Color.blue);
        HandleCameraCollision(collisionDestination,ref targetPosition);
```

```
        transform.position = Vector3.Lerp(transform.position, targetPosition, cameraRig.Damping * Time.deltaTime);

        transform.rotation = Quaternion.Lerp(transform.rotation, cameraLookTarget.rotation, cameraRig.Damping *
Time.deltaTime);
    }

    private void HandleCameraCollision(Vector3 toTarget, ref Vector3 fromTarget)
    {

        RaycastHit hit;

        if (Physics.Linecast(toTarget, fromTarget, out hit))
        {
            Vector3 hitPoint = new Vector3(hit.point.x + hit.normal.x * .2f, hit.point.y, hit.point.z + hit.normal.z * .2f);
            fromTarget = new Vector3(hitPoint.x, fromTarget.y, hitPoint.z);



        }
    }
}
```

AmmoPickup:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AmmoPickup : PickupItem
{
    [SerializeField] EWeaponType weaponType;
    [SerializeField] float respawnTime;
    [SerializeField] int amount;
    public override void OnPickup(Transform item)
    {
        var playerInventory = item.GetComponentInChildren<Container>();
        GameManager.Instance.Respawner.Despawn(gameObject, respawnTime);
        playerInventory.Put(weaponType.ToString(), amount);

        item.GetComponent<Player>().PlayerShoot.ActiveWeapon.Reloader.HandleOnAmmoChanged();

    }
}
```

PlayerShoot:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[RequireComponent(typeof(CharacterController))]
[RequireComponent(typeof(PlayerState))]
[RequireComponent(typeof(PlayerHealth))]
public class Player : MonoBehaviour
{
   [System.Serializable]
   public class MouseInput {
      public Vector2 Damping;
      public Vector2 Sensitivity;
      public bool LockMouse;

   }


   [SerializeField]
   SwatSolider settings;

   [SerializeField] MouseInput MouseControl;
   [SerializeField] AudioController footsetps;
   [SerializeField] float minimalMoveTreshold;

   public PlayerAim playerAim;


   Vector3 previousPosition;

   private CharacterController m_MoveController;
   public CharacterController MoveController {
      get {
         if (m_MoveController == null)
            m_MoveController = GetComponent<CharacterController>();
         return m_MoveController;

      }

   }

   private PlayerShoot m_PlayerShoot;
   public PlayerShoot PlayerShoot {
      get {
         if (m_PlayerShoot == null)
            m_PlayerShoot = GetComponent<PlayerShoot>();
         return m_PlayerShoot;

      }
```

```
}



private PlayerState m_PlayerState;
public PlayerState PlayerState
{
   get
   {
      if (m_PlayerState == null)
         m_PlayerState = GetComponentInChildren<PlayerState>();
      return m_PlayerState;

   }

}


private PlayerHealth m_PlayerHealth;
public PlayerHealth PlayerHealth
{
   get
   {
      if (m_PlayerHealth == null)
         m_PlayerHealth = GetComponentInChildren<PlayerHealth>();
      return m_PlayerHealth;

   }

}



InputController playerInput;
Vector2 mouseInput;

void Awake()
{
   playerInput = GameManager.Instance.InputController;
   GameManager.Instance.LocalPlayer = this;

   if (MouseControl.LockMouse) {
      Cursor.visible = false;
      Cursor.lockState = CursorLockMode.Locked;

   }


}
```

```
    void Update()
    {
        if (!PlayerHealth.IsAlive)
            return;

        Move();
        LookAround();

    }



    void Move()
    {
        float moveSpeed =settings.RunSpeed;

        if (playerInput.IsWalking)
         moveSpeed =settings.WalkSpeed;

        if (playerInput.IsSprinting)
            moveSpeed = settings.SprintSpeed;

        if (playerInput.IsCrouched)
            moveSpeed = settings.CrouchSpeed;

        if (PlayerState.MoveState == PlayerState.EMoveState.COVER)
            moveSpeed = settings.WalkSpeed;


        Vector2 direction = new Vector2(playerInput.Vertical *moveSpeed, playerInput.Horizontal* moveSpeed);

        MoveController.SimpleMove(transform.forward * direction.x + transform.right * direction.y);

        if (Vector3.Distance(transform.position, previousPosition)>minimalMoveTreshold)
            footsetps.play();

        previousPosition = transform.position;
    }


    void LookAround(){
mouseInput.x = Mathf.Lerp(mouseInput.x, playerInput.MouseInput.x, 1f / MouseControl.Damping.x);
        mouseInput.y = Mathf.Lerp(mouseInput.y, playerInput.MouseInput.y, 1f / MouseControl.Damping.y);

        transform.Rotate(Vector3.up* mouseInput.x* MouseControl.Sensitivity.x);


        playerAim.SetRotation(mouseInput.y * MouseControl.Sensitivity.y);

}

}
```

PlayerHealth:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerHealth : Destructable
{
   [SerializeField]
   Ragdoll ragDoll;


   public override void Die()
   {
      print("Die");
      base.Die();
      ragDoll.EnablRagdoll(true);



   }
}
```

Destructable:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;


[RequireComponent(typeof(Collider))]
public class Destructable : MonoBehaviour
{
   [SerializeField] float hitPoints;

   public event System.Action OnDeath;
   public event System.Action OnDamageReceived;

   float damageTaken;

   public float HitPointsRemaining
   {

      get
      {
         return hitPoints - damageTaken;

      }
```

```
    }
    public bool IsAlive
    {
        get
        {

            return HitPointsRemaining > 0;

        }


    }
    public virtual void Die()
    {


        if (OnDeath != null)
            OnDeath();



    }


    public virtual void TakeDamage(float amount)
    {
        //   print(hitPoints*5);

        if (!IsAlive)
            return;


        damageTaken += amount;

        if (OnDamageReceived != null)
            OnDamageReceived();

        if (HitPointsRemaining <= 0)
        { Die(); }


    }
    public void Reset()
    {
        damageTaken = 0;
    }

}
```

Move:

```
void Move()
   {
      float moveSpeed =settings.RunSpeed;

      if (playerInput.IsWalking)
       moveSpeed =settings.WalkSpeed;

      if (playerInput.IsSprinting)
         moveSpeed = settings.SprintSpeed;

      if (playerInput.IsCrouched)
         moveSpeed = settings.CrouchSpeed;
      if (PlayerState.MoveState == PlayerState.EMoveState.COVER)
         moveSpeed = settings.WalkSpeed;


      Vector2 direction = new Vector2(playerInput.Vertical *moveSpeed, playerInput.Horizontal*
moveSpeed);

      MoveController.SimpleMove(transform.forward * direction.x + transform.right * direction.y);

      if (Vector3.Distance(transform.position, previousPosition)>minimalMoveTreshold)
         footsetps.play();

      previousPosition = transform.position;
   }
```

Chapter 5

Software Testing

Software Testing is the most crucial part of Software Development Process. It is the investigation or evaluation of a software component, improving them, and finding bugs and defects. Testing is usually done by executing a system in such a way that it identifies any gaps, errors, or missing requirements in contrary to the actual requirements. This chapter provides a description about the adopted testing procedure. This includes the adopted testing methodology, test suite and the test results of the developed software.

5.1 Testing Methodology

We have used black box testing in our testing phase. Black box contains certain benefits that include following:

- Black box testing examines the functionality of application without peering into its internal structure.
- This method can be applied to every level of the software testing which include unit, integration, system and acceptance.

Black Box unit testing is used in our project. Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules, together with associated control data, usage procedures and operating procedures are tested to determine whether they are fit for purpose.

5.2 Testing Environment

5.2.1 Test Case: Moving Character Forward

Table 5.1 Moving Character Forward

Date: 04 February, 2020	
System: Gangster's War.	
Objective: Character movement.	*Test ID:*1
*Version:*1	*Test Type:* Unit testing
Input: **Press W to move character forward**	
Expected Result: **Moving character forward.**	
Actual Result: passed	

5.2.2 Test Case: Moving Character Backward

Table 5.2 Moving Character Backward

Date: 04 February, 2020	
System: Gangster's War.	
Objective: Character movement.	*Test ID:*2
*Version:*1	*Test Type:* Unit testing
Input: **Press S to move character backward**	
Expected Result: **Moving character backward.**	
Actual Result: passed	

5.2.3 Test Case: Moving Character Right

Table 5.3 Moving Character Right

Date: 04 February, 2020	
System: Gangster's War.	
Objective: Character movement.	*Test ID:*3
*Version:*1	*Test Type:* Unit testing
Input: **Press D to move character backward**	
Expected Result: **Moving character Right side.**	
Actual Result: passed	

5.2.4 Test Case: Moving Character Left

Table 5.4 Moving Character Left

Date: 04 February, 2020	
System: Gangster's War.	
Objective: Character movement.	*Test ID:*4
*Version:*1	*Test Type:* Unit testing
Input: **Press A to move character backward**	
Expected Result: **Moving character Left side.**	
Actual Result: passed	

5.2.5 Test Case: Jumping

Table 5.5 Jumping

Date: 04 February, 2020	
System: Gangster's War.	
Objective: Character movement.	*Test ID:*5
*Version:*1	*Test Type:* Unit testing
Input: **Press SPACE to jump by character.**	
Expected Result: **Jumping of character.**	
Actual Result: passed	

5.2.6 Test Case: Crouching

Table 5.6 Crouching

Date: 04 February, 2020	
System: Gangster's War.	
Objective: Character movement.	*Test ID:*6
*Version:*1	*Test Type:* Unit testing
Input: **Press SHIFT to move character while crouching**	
Expected Result: **Crouching.**	
Actual Result: passed	

5.2.7 Test Case: Creating Server

Table 5.7 Creating Server

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **creating server.**	*Test ID:*7
*Version:*1	*Test Type:* Unit testing
Input: **Enter in online mode to create server**	
Expected Result: **Created server**	
Actual Result: passed	

5.2.8 Test Case: Joining Server

Table 5.8 Joining Server

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **creating server.**	*Test ID:*8
*Version:*1	*Test Type:* Unit testing
Input: **Press on server to join server**	
Expected Result: **Joined server.**	
Actual Result: passed	

5.2.9 Test Case: Viewing Map

Table 5.9 Viewing Map

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **creating server.**	*Test ID:*9
*Version:*1	*Test Type:* Unit testing
Input: **Entered in map.**	
Expected Result: **Shown map.**	
Actual Result: passed	

5.2.10 Test Case: Firing

Table 5.10 Firing

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*10
*Version:*1	*Test Type:* Unit testing
Input: **Left Click from mouse.**	
Expected Result: **Shoot fire.**	
Actual Result: passed	

5.2.11 Test Case: Moving in the houses

Table 5.11 Moving in the houses

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*11
*Version:*1	*Test Type:* Unit testing
Input: **Move towards the houses.**	
Expected Result: **Moved In the house.**	
Actual Result: passed	

5.2.12 Test Case: Joining AI Mode

Table 5.12 Joining AI Mode

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*12
*Version:*1	*Test Type:* Unit testing
Input: **Click AI mode option.**	
Expected Result: **AI mode joined.**	
Actual Result: passed	

5.2.13 Test Case: Opening Scope

Table 5.13 Opening Scope

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*13
*Version:*1	*Test Type:* Unit testing
Input: **Press right mouse button while holding sniper**	
Expected Result: **Show zoomed view.**	
Actual Result: passed	

5.2.14 Test Case: Damaging Health

Table 5.14 Damaging Health

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*14
*Version:*1	*Test Type:* Unit testing
Input: **Press left mouse button to shoot.**	
Expected Result: **Health low to player who got hit.**	
Actual Result: passed	

5.2.15 Test Case: Getting damage by bots

Table 5.15 Getting damage by bots

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*15
*Version:*1	*Test Type:* Unit testing
Input: **Get shot by bot (PC based players).**	
Expected Result: **Get damaged and got killed**	
Actual Result: passed	

5.2.16 Test Case: Aiming

Table 5.16 Aiming

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*16
*Version:*1	*Test Type:* Unit testing
Input: **Hold pointer towards enemy**	
Expected Result: **X mark shows where is now aimed the gun**	
Actual Result: passed	

5.2.17 Test Case: Moving Character

Table 5.17 Moving Character

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*17
*Version:*1	*Test Type:* Unit testing
Input: **Press any key from W,S,A,D to move character**	
Expected Result: **Movement shown on screen as player press key.**	
Actual Result: passed	

5.2.18 Test Case: Sending Messages

Table 5.18 Sending Messages

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*18
*Version:*1	*Test Type:* Unit testing
Input: **Press TAB button from keyboard to open chat dialog from bottom left side.**	
Expected Result: **Type any message to press ENTER to broadcast that message to whole server.**	
Actual Result: passed	

5.2.19 Test Case: Show Score

Table 5.19 Show Score

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*19
*Version:*1	*Test Type:* Unit testing
Input: **Score can be shown on top right corner while enemies will show on his head.**	
Expected Result: **Viewing score on screen**	
Actual Result: passed	

5.2.20 Test Case: Show Deaths

Table 5.20 Show Deaths

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*20
*Version:*1	*Test Type:* Unit testing
Input: **Deaths can be shown on top right corner while enemies will show on his head.**	
Expected Result: **Viewing Deaths on screen**	
Actual Result: passed	

5.2.21 Test Case: Show Ammo

Table 5.21 Show Ammo

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*21
*Version:*1	*Test Type:* Unit testing
Input: **Ammo can be shown on top right corner while enemies will show on his head.**	
Expected Result: **Viewing ammo on screen**	
Actual Result: passed	

5.2.22 Test Case: Enter Player Name

Table 5.22 Enter Player Name

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*22
*Version:*1	*Test Type:* Unit testing
Input: **While creating server also enter your name to use during game.**	
Expected Result: **Name set successful.**	
Actual Result: passed	

5.2.23 Test Case: Interact with the environment

Table 5.23 Interact with the environment

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*23
*Version:*1	*Test Type:* Unit testing
Input: **After joining server player can move in map freely and move to and fro.**	
Expected Result: **Go in map.**	
Actual Result: passed	

5.2.24 Test Case: Attach Using Guns

Table 5.24 Attach Using Guns

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*24
*Version:*1	*Test Type:* Unit testing
Input: **Players can attack on each other using different kind of guns as sniper and assault.**	
Expected Result: **Attacking successful.**	
Actual Result: passed	

5.2.25 Test Case: Leaving Server

Table 5.25 Leaving Server

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*25
*Version:*1	*Test Type:* Unit testing
Input: **Press ESC button to open further options to select disconnect.**	
Expected Result: **Leaving server**	
Actual Result: Failed	

5.2.26 Test Case: Firing gun while running in AI mode

Table 5.26 Firing gun while running in AI mode

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*26
*Version:*1	*Test Type:* Unit testing
Input: **Press Mouse left button to fire while running.**	
Expected Result: **Firing done.**	
Actual Result: Failed	

5.2.27 Test Case: Multiple Players

Table 5.27 Multiple Players

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*27
*Version:*1	*Test Type:* Unit testing
Input: **Joining server by different players in online mode.**	
Expected Result: **Joined**	
Actual Result: Passed.	

5.2.28 Test Case: Jump on containers

Table 5.28 Jump on containers

Date: 04 February, 2020	
System: Gangster's War.	
Objective: **Firing**	*Test ID:*28
*Version:*1	*Test Type:* Unit testing
Input: **When player wants to jump on containers present in the map.**	
Expected Result: **Jumped on them.**	
Actual Result: Failed.	

Chapter 6

Software Deployment

6.1 Installation / Deployment Process Description

The downloading of game can be done by using this link:

https://mega.nz/file/YbYHjKoJ#rkbS03qNuyXeym1975d5x4L7V8vGKT3LlNzsIZBBOCE

To RUN:

- Download RAR file as:

Figure 6-1 Deployment

Having size of 867.7 MB

- Then UNZIP this file as Gangsters War**.rar** in any folder.
- Open EXE file as located in Gangster's war \Full.exe

Then game will run and can be played by any player around the globe.

Chapter 7

Project Evaluation

This chapter includes the examiner's evaluation report, including the points to be revised/included along with the selected requirements in the next iteration.

7.1 Project Evaluation Report

Examiner Name:	
S. No.	**Suggestion**

Other Comments (If any):

_____

_____

_____

_____

_____

Signature