Team Project

SOFTWARE REQUIREMENTS SPECIFICATION

for

Escort The President

Prepared by Team A1

James Birch, Ahmed Bhallo, Edward Dean, Brendan Hart, Kwong Hei Tsang

March 21, 2017

Contents

1 Introduction

1.1 Purpose

We are team A1, designing and implementing the game "Escort The President". The purpose of this System Requirements document is to specify the overall system requirements that will govern the development and implementation of our game.

1.2 Game Concept

Escort The President is a multiplayer game where each rounds consists of Assassins attempting to assassinate the recently-elect president. Assassins carefully plot their move to kill the president, knowing there's a $\$2,000,000$ bounty awaiting. The president has been followed inside his car and he receives a tip that Assassins are patiently waiting to make their move... Gunshot! An Assassin has shot at the vehicle and the president's Escorts with the aid of Police Officers attempt to escort him to the helicopter *alive*. To make matters worse, this is in public streets and there are civilians everywhere!

1.3 Research

1.3.1 Target Audience

Our target audience are teenagers from the age of 12 - 18 who enjoy fast-paced action shooter games. Because of this, our game must be engaging and aesthetically suited to meet our audiences' preferences.

1.3.2 Perspective

We have decided to go for a 2.5D top-down perspective, but not a direct birds-eye view in order to give depth to boundaries such as walls [1]. This will grant players the freedom to move in 8 directions and feel in control of their unit. The game will only have 2 layers of depth (in order to not be confusing) – The ground tiles (the floor), and the obstacle tiles (think of boxes and walls placed on the ground that are the size of a character).

1.3.3 Character Movement

When the creator of Super Mario Bros, Shigeru Miyamoto, was asked why the Mario franchise was so popular, he talked about the way they implemented movement [2]. One element of Mario's movement was his slide (he doesn't stop moving exactly when a button is released). This is something we are planning to implement in our own game to make movement feel more natural, making our players feel immersed into the game.

1.3.4 Sprites

Unit rendering will require multiple different sprites depending on the direction they are facing. 1 base sprite for animation will be created, which will be used for every unit. In order to give units their uniqueness, sprites will programatically be modified on load (for example, changing the colour of clothes/skin, adding hair/hats).

1.3.5 Sound and Music

The game must have arcade-sounding sound effects and chip-tune music to give the game a nostalgic arcade feel.

1.3.6 Networking

Both TCP and UDP will be used as the transfer protocol on the server and client implementation in order to make the game work in most network environments. The player will be allowed to select which protocol to use during log on.

TCP TCP provides a mechanism to guarantee any messages sent between server and the client reaching each other. Libraries for encryption are also ready for use to secure the data sent between the server and client. It is useful if the client is connecting through an insecure network such as public WiFi or a restrictive network which prevents UDP from working properly.

UDP UDP does not guarantee each packet can reach each side as packet loss are not handled by the protocol. The benefit from it is not all lost messages have to be resent as they may be out of date such as the game state. This allows server and client to selectively resend important lost information only, increasing efficiency. At the same time, it reduces handshaking to reduce the delay. It is useful if the player is far away from the server or is connecting through a trusted network.

Other parts of the game will only use MessageControl interface to send or receive objects of the Message class to avoid interacting directly with the networking subsystem. If the client and server is communicating via UDP, the networking module will determine whether specific message is critical or not to decide whether to resend the message when the other side fails to receive it.

1.4 Glossary and Acronyms

- HP: Health Points
- Ammo: Weapon ammunition
- Mags: Ammunition magazines
- Combat activity: Combat activity occurs when a unit attempts to attack (by shooting for throwing a grenade).
- HUD: Head-up display. Visually relaying information as part of the user interface.

1.5 References

[1] Brandon Dollar: Types of Game Perspectives,
`https://mozzastryl.wordpress.com/2013/01/20/types-of-game-perspectives/`

[2] Shigeru Miyamoto: Miyamoto on World 1-1: How Nintendo made Mario's most iconic level,
`https://www.youtube.com/watch?v=zRGRJRUWafY`

2 System Features

We have documented the features of our game in the form of functional requirements.

2.1 Units

A unit is a "character" of our game. Units in our game are the President, Escorts, Assassins (including Menaces), and Police Officers.

2.1.1 Unit assignment

REQ 1.1.1: The lobby system must randomly assign all human players a role of either Escort or Assassin at the beginning of each round.

REQ 1.1.2: The lobby system must ensure that a ratio of 1 Escort for every 2 Assassins is maintained. If there are not enough human players to form assassins, the numbers must be made up with AI players.

REQ 1.1.3: The lobby system must assign around 20 AI players to civilians.

2.1.2 President (AI only)

REQ 1.2.1: The president AI must remain stationary in his position until an Escort asks to follow. This is done by an Escort by walking up near to the president and pressing the Action key.

REQ 1.2.2: When following an Escort, the president must find the path from the president's current position to the Escort's current position. The path must be recomputed every time the Escort makes a new move.

REQ 1.2.3: Damage dealt by Escorts have no effect on the president.

2.1.3 Escort (Human only)

It is the Escorts' duty to ensure the safety of the president. They must fend off and kill Assassins while escorting the president to the save zone.

REQ 1.3.1: Damage dealt by Escorts have no effect on other Escorts.

REQ 1.3.2: All Escorts win when the president has safely reached the safe zone.

2.1.4 Assassins (Human and AI)

The Assassins' goal is to be the one to kill the president. Only the assassin who dealt the killing blow to the president will win the game.

REQ 1.4.1: When an Assassin deals the killing blow to a civilian, they become a menace for 20 seconds.

REQ 1.4.2: Damage dealt by Assassins have no effect on other Assassins.

REQ 1.4.3: Damage dealt by Menaces *do* have an effect on Assassins.

2.1.5 Menace (Human and AI)

A Menace is an Assassin who has recently killed a civilian. Menaces return back to normal Assassins after a set period of time (20 seconds). It is beneficial for Assassins to kill a Menace to relieve the competition of being the one to kill the president.

REQ 1.5.1: Damage dealt by Assassins *do* have an effect on Menaces.

REQ 1.5.2: Menaces may now attack other Assassins until their time is up.

2.1.6 Police Officers (AI only)

The sole purpose of a police officer is to find and kill all Assassins and ultimately protect the president at all costs.

REQ 1.6.1: A police must reload their gun if they have magazines left and are not currently involved in combat.

REQ 1.6.2: A police must run around the map, but staying within a certain distance from the president.

REQ 1.6.3: If assassin combat activity is detected near a police officer, they must run there are shoot at all Assassins that they encounter.

REQ 1.6.4: If there is a line of sight between a Police Officer and an Assassin, the police must wait 100-200ms (decided randomly) and shoot a small random offset away from the target angle. Random values are used for realism.

2.1.7 Civilians (AI only)

Civilians are neutral units who do not attack but can be attacked by other units.

REQ 1.7.1: If combat activity occurs near a civilian, the civilian must run in the opposite direction for 5 seconds.

REQ 1.7.2: If no combat activity has occurred near a civilian in the last 5 seconds, civilians walk around as normal.

2.2 Combat System

2.2.1 Machine Gun

REQ 2.1.1: Machine Guns are a fully-automatic weapon with a fire rate and cooldown of 0.15 seconds. This means that holding down shoot will only fire one bullet per 0.15 seconds. Continuously pressing and releasing the shoot button will only shoot one bullet per 0.15 seconds.

REQ 2.1.2: Machine Guns' bullets deal 40 damage to the unit that it hits.

REQ 2.1.3: Machine Guns have 12 rounds per magazine.

REQ 2.1.4: Reloading a partially-full magazine automatically transfers these bullets to the next magazine.

2.2.2 Pistol

REQ 2.2.1: Pistols are a semi-automatic weapon with a cooldown of 0.3 seconds. This means that the fastest a unit can fire is once every 0.3 seconds.

REQ 2.2.2: Pistols' bullets deal 50 damage to the unit that it hits.

REQ 2.2.3: Pistols have 8 rounds per magazine.

REQ 2.2.4: Pistols have unlimited magazines.

2.2.3 Grenades

REQ 2.3.1: Grenades deal 60-0 HP depending on how close the damaged unit is close to the grenade.

REQ 2.3.2: After a grenade is thrown, it will explode after 5 seconds.

2.2.4 Blast Shields

REQ 2.4.1: Blast shields are only given to Escorts to aid mitigate and block damage. Blast shields can absorb 500HP worth of damage until they deteriorate and disappear.

REQ 2.4.2: The holder of a blast shield cannot run and must walk.

REQ 2.4.3: Blasts shields will black and absorb 100% of bullet damage.

REQ 2.4.4: Blast shields will absorb 80% if grenade damage for each unit, if the straight line between that unit and the grenade intersects a blast shield.

2.2.5 Death

REQ 2.5.1: A unit dies when they have 0 or less HP.

REQ 2.5.2: A dead unit drops all of his magazines (if they had any), which can be picked up by another player (walking on top of the corpse).

REQ 2.5.3: Dead players must wait until they respawn. Until then, they may spectate players on their same team. i.e. Assassins may spectate other Assassins, Escorts may spectate other Escorts or Police Officers.

REQ 2.5.4: If there are no players to spectate the area that the player died in is shown.

2.2.6 Game Balancing

The following table shows default values for each unit and their IDs. All values in this table are subject to change during the implementation and user testing phase.

Unit	HP on spawn	Max HP	Mags on spawn	Max mags	Gren* on spawn	Max Gren*	Machine gun	Pistol	Blast shield?	Respawn time (sec)
President	500	500	0	0	0	0	No	No	No	N/A
Escort	100	100	5	6	2	3	Yes	Yes	Yes	5
Assassin (+ Menace)	75	100	4	6	2	3	Yes	Yes	No	10
Police Officer	50	50	3	6	0	0	Yes	Yes	No	10
Civilian	50	50	0	0	0	0	No	No	No	N/A

*Gren = Grenade

2.2.7 Game Rounds

The game is played in rounds.

REQ 2.7.1: A round begins with all units in their spawn points.

REQ 2.7.2: A round ends when the president is killed (the Assassin who killed the president wins), or the president reaches the safe zone (all Escorts win).

REQ 2.7.3: After a round has ended, all players return to the lobby and the creator of the lobby can start a new game (as well as being able to perform other lobby functionality).

REQ 2.7.4: If an Escort leaves a game, a new Police Officer is added.

REQ 2.7.5: If an Assassin (or Menace) leaves a game, they are replaced by an AI.

REQ 2.7.6: If all players leave a game or lobby, the lobby terminates.

2.2.8 Maps

The game can be played on multiple maps.

REQ 2.8.1: All maps must have an Assassins' spawn point (same as their start point).

REQ 2.8.2: All maps must have an Escorts' spawn point (same as their start point).

REQ 2.8.3: All maps must have basic and super power-up spots.

REQ 2.8.4: All maps must have the president's goal point.

2.2.9 Power-up Spots

Power-up spots are located around the map. There are two types of power-up spots - basic and super. Super power-up spots are less common.

REQ 2.9.1: Power-ups are collected when a unit walks over it. Power-ups cannot be walked on by police officers, the president or civilians.

REQ 2.9.2: A Power-up is restored 30 seconds after it has been picked up. Its new item is generated randomly but within the same group, i.e. a basic power-up spot will only ever regenerate basic power-ups.

REQ 2.9.3: **Basic:** Basic power-ups are consumed as soon as they are walked on. The following basic power-ups must be implemented:

(a) **Affordable care** – Restores 50 HP to the unit.

(b) **Right to bear arms** – Gives the unit 2 additional magazines.

(c) **Bombs away** – Gives the unit 1 extra grenade.

REQ 2.9.4: **Super:** Super power-ups are consumed by pressing the Power-Up button. The following super power-ups must be implemented:

(a) **Spray and pray** – The unit who uses this power up does not use up any ammunition whilst shooting. Grenades are not affected. This effect lasts for 15 seconds.

(b) **Diplomatic immunity** – If an Assassin (or Menace) picks this up, they are invincible. If an Escort picks this up, the president is invincible. This effect lasts for 10 seconds.

(c) **Bald eagle** – When used by an Assassin, all Escorts are visible on the radar to all Assassins. When used by an Escort, all Assassins are visible on the radar to all Escorts. This effects lasts for 20 seconds.

REQ 2.9.5: If a unit walks on a super power-up without using their old one, the old one is replaced with the new one.

2.3 Unit Control

2.3.1 Input handling

REQ 3.1.1: All input keys (except the mouse buttons) can be changed in the settings panel. Below is a table of defaults.

Action	Default		Action	Default		Action	Default
Move up	W		Jump	Space bar		Switch to machine gun	1
Move down	S		Roll	Shift + Move		Switch to pistol	2
Move left	A		Settings menu	Escape		Switch to blast shield	3
Move right	D		Chat	T		Scroll between weapons	Scroll wheel
Shoot	Left click		Reload	R		View scoreboard	Tab
Throw grenade	Right click		Use power-up	F		Switch camera mode	C

2.3.2 Movement and Actions

REQ 3.2.1: All units (except Civilians) are constantly running while moving.

REQ 3.2.2: All units stop running and start walking when the following actions are performed: Shooting, reloading, holding a blast shield in their current weapon slot.

REQ 3.2.3: When a unit stops moving, they start sliding for a very short distance in order to mimic weight.

REQ 3.2.4: Units can roll, which allows them to move fast for a very short period.

REQ 3.2.5: A unit cannot roll again less than 5 seconds after they have already rolled.

REQ 3.2.6: During rolling and jumping, a unit cannot perform any other actions such as shooting or reloading.

REQ 3.2.7: A player can rotate by moving their mouse around the screen. A player's direction will dictates which way they will shoot or throw grenades.

REQ 3.2.8: A player can reload by pressing the Reload button.

REQ 3.2.9: A player can shoot by pressing the Shoot button. If there is a bullet left in their current magazine, the gun will shoot in the direction that they are facing.

REQ 3.2.10: If a player shoots without any bullets in their current magazine, they will attempt to reload. If there are no magazines left, nothing happens.

2.4 Graphics

2.4.1 Camera

REQ 4.1.1: Players will have three options in how the camera will be handled: Locked, unlocked and semi-locked. They can switch between the three at any time.

REQ 4.1.2: **Locked:** The camera always follows the player's character automatically. If the play is near an edge/corner, the camera stops following the player until they have moved away from the corner.

REQ 4.1.3: **Unlocked:** The camera never moves automatically. The player must move their mouse to the edges of the screen to tell the camera where to go.

REQ 4.1.4: **Semi-locked:** A combination of locked and semi unlocked. The camera follows the player, but some portions of the map near the player can be viewed by moving the mouse to the edges of the screen.

2.4.2 HUD

REQ 4.2.1: The HUD will consist of the mini map (radar), the chat, the players' scores, the players weaponary, all players' HP bars and the scoreboard.

REQ 4.2.2: The mini map will be a simplified version of the map, where players' location and their allies will be shown. Enemy location will also be shown if they have recently been responsible for any combat activity or the Bald Eagle power-up has been activated. The president's location is shown at all times to all players.

REQ 4.2.3: Players may communicate via the chat. Any message sent will be shown to all players in the game.

REQ 4.2.4: The players' own kill/death score will be displayed on the screen.

REQ 4.2.5: The players' weaponry, their ammunition state and the current selected weapon slot is displayed.

REQ 4.2.6: All players have a HP bar which is displayed on top of the unit's sprite. This is viewable to all players.

REQ 4.2.7: Pressing the scoreboard tab will allow players to view all players' kill/death score and their usernames.

2.4.3 Sprites and Animation

REQ 4.3.1: The base unit sprite must be colour customisable.

REQ 4.3.2: The base sprite will need to have the following animations: Running, shooting, jumping, rolling, reloading, throwing grenade, taking bullet damage, taking grenade damage (being airborne), and dying.

REQ 4.3.3: The grenade sprite will need to have the following animations: Being thrown in the air, travelling, explosion.

3 Project Planning

3.1 Project Schedule

A = Ahmed, B = Brendan, E = Edward, J = James, K = Kwong

Assignee	Task	09/1	16/1	23/1	30/1	06/2	13/2	20/2	27/2	06/3	13/3	20/3
A	Camera functionality											
A	Sprites and animation											
A	Player movement											
A	Input handling											
A	Map loading											
A	User interface											
K	User properties											
J	Weapon switching											
J, A	Player shooting											
J, A	Grenade throwing											
K	Sound engine											
E	Units receiving damage											
E	Damage mitigation											
E	Unit death											
A	Scoreboard display											
E	Calculating KDR											
B	Spectator system											
B	Respawning											
A, J	Projectile logic											
B	Power-ups											
B	Menace System											
All	Map creation											
K	TCP (TLS)											
K	UDP (Basic comm.)											
K, E	UDP packet loss handling											
K, E	Connecting to server											
K	Lobby management											
J	Unit assignment											
J	Returning to lobby											
B	Path finding											
B, E, J	AI Combat System											
B, E, J	Civilian AI											
B, E, J	President AI											
B, E, J	Police AI											
B, E, J	Assassin AI											
J	Entity collision detection											
All	System integration											
K, E	Server load testing											
K	Improve network security											
All	Game logic unit testing											
All	Acceptance testing											

3.2 Risk Analysis

Risks	Likelihood	Methods to mitigate the risk
Members not understanding what they should be doing.	Likely towards the start of the project.	Have a discussion about all of the tasks that need to be carried out and assign them to people. Create a Gantt chart to help.
People falling behind on tasks.	Likely towards middle to the end of the project.	Try to make tasks well defined and make time estimates realistic. Build some slack in to the time.
Components not integrating easily.	Quite likely.	Spend more time on the design and on the interfaces in which components are intended to communicate via.
Duplication of work.	Quite unlikely.	If the tasks are well-defined and assigned accordingly then there should not be too much overlap.
People falling ill.	Unlikely.	Pick up other tasks while the person is ill. Hopefully it will not be severe and they will be able to do other tasks on return.
People not turning up for meetings.	Unlikely.	Seek assistance from the lecturer. May have to pick up other tasks on behalf of the missing person.
Some members doing more work than others.	Quite likely.	Try as best as possible to try and plan tasks so that the workload is balanced. If a task is unexpectedly short for someone then that person should seek out more tasks to help.

4 Use Cases

Use case name	**User login**
Description	The user wants to be able to connect to a server with a desired username and proceed to the main menu.
Primary actor(s)	The user
Secondary actor(s)	Client, Server
Triggering event	The game first opens.
Preconditions	The user is able to run the game. The user has an internet connection.
Postconditions	The user is connected to the server.
Main success scenario	**Goal:** User connected to user and enters main menu. 1. The setup menu is displayed on the client. 2. The user fills in their desired username, server details and connection method (UDP/TCP). 3. The user clicks the "Connect" button. 4. The client sends a message to the server with this information. 5. The server responds with a unique ID if the username was not already in use. 6. The user is then presented with the main menu. 7. The client requests a lobby list refresh from the server every 5 seconds, if the user is not currently in a game.
Alternate flow	**Branching action:** The user missed a required field or has gave invalid input. 3 a. Prompt user to correct the form. Go to step 2. **Branching action:** Connection to server cannot be established. 4 a. The user is informed that the connection failed. Go to step 2. **Branching action:** Username was not unique. 5 a. The user is informed that the username was not unique. Go to step 2.

Use case name	**User creates lobby**
Description	The user wants to create a lobby so other players can join, ready to play a game.
Primary actor(s)	The user
Secondary actor(s)	Client, Server
Triggering event	The user clicks the "Create Lobby" button.
Preconditions	The user is connected to the server.
Postconditions	A lobby is created which other players can join, optionally with a password.
Main success scenario	**Goal:** User creates a lobby, optionally password-protected. 1. The user is prompted for a name for the lobby and an optional password by the client. 2. The user enters a name and a password for the lobby. 3. The client sends this information to the server. 4. The server creates a lobby. 5. The server sends the lobby ID back to the client. 6. The client now refreshes the list of lobbies.
Alternate flow	**Branching action:** The user has missed out the password field. 3 a. A lobby is created without a password. Go to step 4.
Exceptions	**Branching action:** Connection to the server is not present. 4 a. The user is informed that a connection is not present. **Branching action:** Lobby creation unsuccessful by server. 5 a. The user is informed that lobby creation was unsuccessful.

Use case name	**User joins lobby**
Description	The user wants to be able to join an already created lobby in order to play the game.
Primary actor(s)	The user
Secondary actor(s)	Client, Server
Triggering event	The user selects a lobby to join from the lobby menu.
Preconditions	The user is connected to the server.
Postconditions	The user has joined the desired lobby.
Main success scenario	**Goal:** User joins their selected lobby. 1. The user is asked for the password for the selected lobby by the client. 2. The server verifies the password is correct. 3. The server checks if the user has been kicked from the lobby in the last 30 minutes. 4. The user then presented with the lobby by the client.
Alternate flow	**Branching action:** The selected lobby does not have a password. 1 a. If the lobby is not password protected, the player can join the lobby without the password needing to be validated, go to step 3.
Exceptions	**Branching action:** Incorrect password entered. 2 a. The user is denied entry to the lobby. **Branching action:** User has been kicked from this lobby in the last 30 minutes. 3 a. The user is denied entry to the lobby.

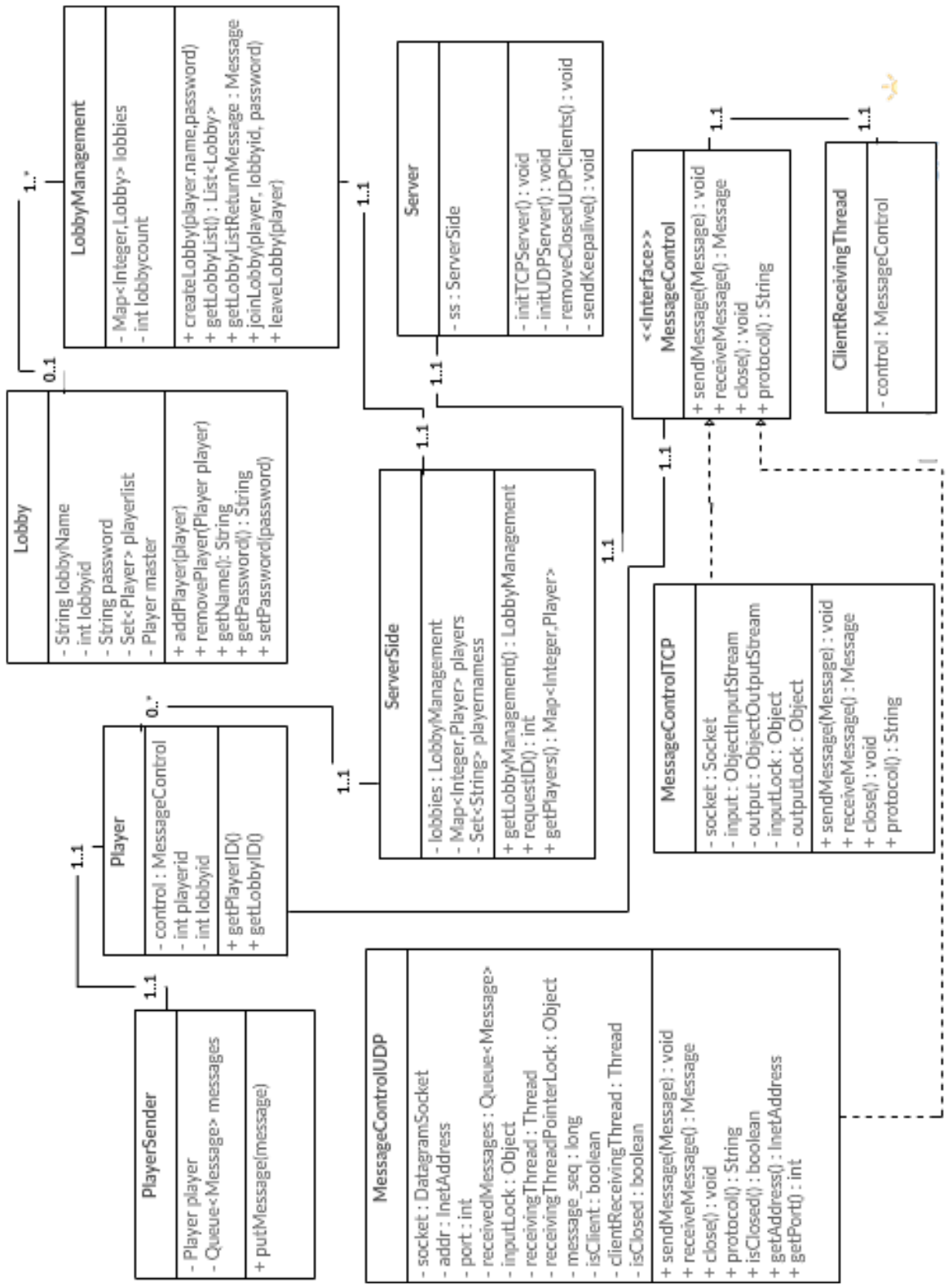
Use case name	**Unit fires gun**
Description	When a unit fires a gun, a bullet is produced with the intention of doing damage.
Primary actor(s)	Unit who fired the gun
Secondary actor(s)	Another unit, Server
Triggering event	A unit fires a gun.
Preconditions	The unit possesses a gun and has it selected as their current weapon slot.
Postconditions	The produced bullet collides with an object and has the appropriate effect.
Main success scenario	**Goal:** The bullet is fired and travels until contact with an object. 1. The unit fires a gun. 2. A bullet is produced. 3. The bullet travels in the direction it was fired. 4. Once the bullet collides with a unit, the appropriate damage is dealt to that unit and the bullet is removed from the game.
Alternate flow	**Branching action:** The bullet collides with a wall or blast shield. 4 a. The bullet has no effect and is removed. **Branching action:** Bullet collides with a friendly unit. 4 b. The bullet has no effect and is removed. **Branching action:** Bullet collides with a blast shield. 4 b. Damage is dealt to the blast shield, the unit holding the blast shield is protected.
Exceptions	**Branching action:** The ammo has ran out for the weapon. 2 a. A bullet is not produced.

Use case name	**Unit throws grenade**
Description	When a unit throws a grenade, it explodes 5 seconds after.
Primary actor(s)	The unit who released the grenade
Secondary actor(s)	Another unit, Server
Triggering event	A unit throws a grenade.
Preconditions	The unit has at least 1 grenade.
Postconditions	The grenade has exploded.
Main success scenario	**Goal:** User joins their selected lobby. 1. The grenade is released by a unit. 2. The grenade travels in the direction is was released, eventually slowing to a stop due to friction. 3. After 5 seconds, the grenade explodes. 4. Damage is applied to any surrounding units (depending on the distance they are from the grenade at the time of explosion) which are able to be damaged by the unit who released the grenade.
Alternate flow	**Branching action:** Grenade collides with a wall or similar object. 2 a. The grenade is reflected. **Branching action:** Blast shield near explosion. 4 a. If a blast shield is between a unit and the explosion, the blast shield absorbs 80% of the damage and the unit 20% of the damage the unit would normally receive without the blast shield protecting it.
Exceptions	**Branching action:** Unit has no grenades. 2 a. No grenade is thrown.

5 Class Diagrams

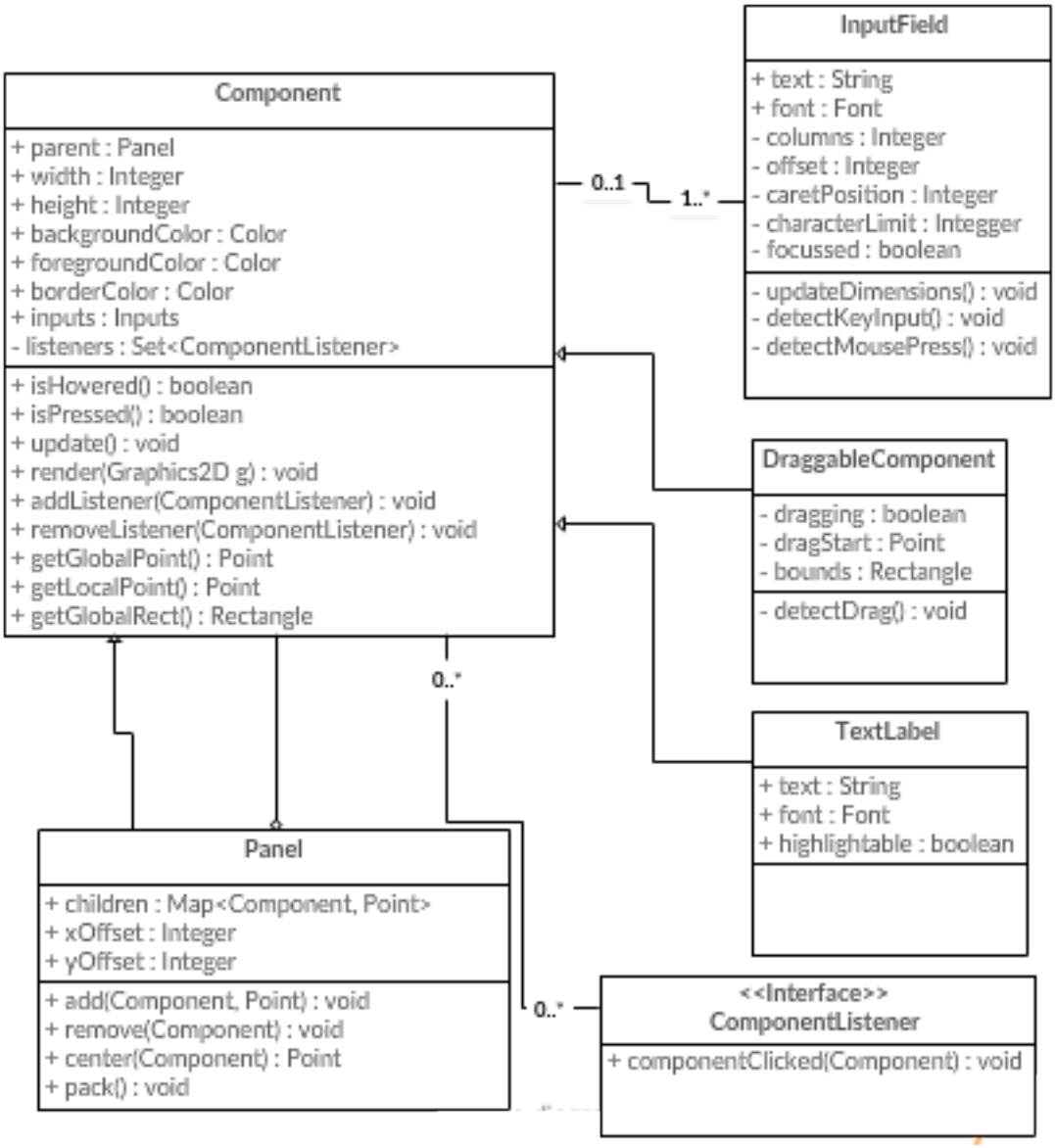
5.1 Server-side Class Diagram

We have created a class diagram to illustrate the server-side classes used for networking, and their relations.



5.2 User Interface Class Diagram

We have also created a class diagram to illustrate the classes of our user interface components.



6 Software Architecture

We have created a component diagram to model the static implementation view of the system, and to illustrate how components should communicate with other components in the system.