**Software Requirements Specification**

**For**

**CS4398 Online Multiplayer Game**

**Version 1.0**

Prepared by Group 4398\_SM\_5

Raul Zuniga

Dayana Joseph

Denise S. Gan

Firdaus Botirzoda

Jie Zhou

2/14/2016

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Raul Zuniga | 2/14/2016 | Initial version. | 1.0 |
| Firdaus Botirzoda | 2/15/2016 | Completed section 5.6 | 1.1 |
| Raul Zuniga | 2/15/2016 | Added Section 1.1 | 1.2 |
| Firdaus and Jie | 2/15//2016 | Added section to section 3 | 1.3 |
| Denise Gan | 2/15/2016 | Added sections 3.3-3.5 to section 3. | 1.4 |
| Dayana Joseph | 2/15/2016 | Added Section 1.6 | 1.5 |
| Denise Gan | 2/15/2015 | Added sections 3.8-3.10 to section 3. | 1.6 |
| Dayana Joseph | 2/15/2015 | Added Section 1.7 | 1.7 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

The reason for this document is to present a detailed description of a new online first-person shooter multiplayer game. It will explain the purpose and features of the system, its interfaces, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for the stakeholders, which consists of Dr. Rodion Podorozhny, the project manager and software engineers who will develop the system.

## **Product Scope**

The system shall provide a fun online multiplayer first-person shooter experience for one or two players. The game will be free to play. The game will be provided by a host server which will contain all the project files, a game engine, and game controller using a real-time connection. Players will play in a map in which there are enemies roaming around that can kill the player. The players will also have the ability to shoot each other and cause health damage. The game will end for the player when the exit for the map is found. The player will be able to play multiple levels. As the player progresses through the levels of the game the difficulty of the game will get progressively harder. The system will not save the state of the game when the player quits the game. When the user plays the game again, the player will start at the beginning of the first level.

## **Document Conventions**

Every detailed requirement statement will have its own priority. This priority will not be inherited by higher-level requirements.

## **Intended Audience and Reading Suggestions**

The audience for this SRS is Dr. Rodion Podoronzhny, the project manager, software and test engineers of the system. It is written in natural English language so a broad audience

of stakeholders can understand it. Section 1 of the SRS is an introduction to the overall system.

Section 2 describes the external interfaces of the system. Section 3 gives a description of the functional requirements of the system. Section 4 gives a description of the informal requirements of the system. Section 5 is the traceability matrix. Section 6 and 7 are the Appendixes.

## **References**

## **Product Perspective**

The software product will be interface with the game's graphical user interface where users can operate can operate all the provided functionality. A domain and host provider will host the website where all the files will be stored. A script on the server will act as a controller of the flow of data between the client(s) and the server. A player will be able to join and quit the game whenever they want. Up to two players will be able to join a game session at a time.

## **Product Functions**

* Ability to create game.
* Ability to start game.
* Ability to enter room.
* Ability to choose player.
* Ability to display game state along with other information for playing purposes.
* Ability for player to kill as many enemies to make it to the exit portal.
* Ability to end game.
* Ability to be a multiplayer game.
* Ability to play over the Internet.
* Must allow player to control their actions.
* Must allow player to access a user menu.
* Must allow player to access a player selection menu.

**Operating Environment**

Version 1.0 of the game software will be developed for use with Windows Operating System (Vista, 7, 8, 8.1, 10) or a mobile device connected to a high bandwidth internet connection. In subsequent versions there will be added support for Chrome, Firefox, Internet Explorer, Opera and Safari browsers and UNIX environments. The system, having been designed first for Windows, will be most compatible with a Windows PC and all hardware that is able to run Windows. Minimum Windows requirements should be suitable for use with this game. If there is no HTML5 browser support, then the user will get a pop-up message in the browser saying to update the browser. Video output is required for functional gameplay with a minimum display resolution of 1024x768. Audio output is required.

Color is used to differentiate several in-game mechanics and as such it will be challenging to create a mode for use with users who are colorblind. This will require a design change that may be implemented in later versions.

## **Design and Implementation Constraints**

A minimum of one player and maximum 2 players will be allowed per game session. The game must allow multiple connections. The game must allow multiple one and two game sessions.

## **Assumptions and Dependencies**

The user must have a high-bandwidth internet connection. The host server can handle multiple connections from many different clients. If the user is playing on a mobile device, it must be a touch capable smart-phone.

# **External Interface Requirements**

## **User Interfaces**

The game shall work and be tested on the following browsers: Internet Explorer, Firefox, Google Chrome, Apple Safari, and Opera. The game shall work and be tested on different devices such as a desktop, tablet with 10-inch screen, and different mobile phones.

## **Communication Interfaces**

Transfer Protocol (HTTP), which is designed to enable communications between clients and servers, will be used. A game controller will interface with a game engine. The game controller will be transferring information between the browser back and forth using a real-time connection.

# **Functional Requirements**

## **User Screen**

User Name





Figure 1 User screen

### A User screen shall be the first screen displayed in the browser when the user types in the game URL and presses enter.

### In the character select screen there shall be an input box for a username.

### In the User screen the input box for username will have system created username by default.

### The User screen shall have a selector to create a room for a one-player game.

### The User screen shall have a selector to join a room for a two-player game.

### Creating a room takes the player to the Selection screen.

## Joinin a room takes the player to the Selection screen

### In the User screen the player starts a one-player game by selecting the selector for creating a room.

### After the selector for creating a room is selected the Selection screen appears.

## Joining a room takes the player to the Selection screen

### In the User screen two players can start a two-player game by selecting the selector to join a room.

### After the selector for creating a room is selected the Selection screen appears.

**Selection Screen** 

1P 2P





Character Character

name name

Figure 2 Selection Screen

### The Selection screen shall have a character select screen for each player.

### The character select screen shall have an image of each character that can be selected by using an arrow button.

### In the character select screen the image of the character will appear in color if there exists another player.

### In the character select screen the image of the character will appear in black and white if there doesn’t exist another player.

### In the Selection screen there shall be a button called “ready” to start the game.

### In the Selection screen there shall be a “ready” button that will light up green after it’s clicked.

## **Starting a 1-player game**

### In the Selection screen the user starts a one-player game by pressing the “ready” button.

## **Starting a 2-player game**

### In the Selection screen the user starts a two-player game by pressing the “ready” button for a two-player game.

## **Creating a game session**

### Once the game is started by pressing the “ready” button for either a one-player or two-player game, the game engine loads the level map.

### After the game session has been initiated and the level map loaded, the dependent game libraries are loaded.

### The main game screen shall briefly display “Level 1” which indicates the current level of the game.

### The game shall display the health indicator of the player during game play.

### The player character(s) shall appear at the start of the map with default weapon loaded

### (pistol).

### The game engine will load enemy sprites.

### Interaction events between enemy sprites and players will wait to be triggered by player movement.

## **Enemy Movement**

### Enemy classes shall move enemy sprites within the game.

### Enemies will have autonomous movement.

### The game shall allow enemy movement dependent on enemy type.

### Riflemen-type enemies shall not be allowed to move.

### Soldier-type enemies shall be allowed to move (run and jump) toward players.

### Static-type enemies shall be allowed to change direction of their weapon toward a player

### but othewise may not move.

## **Enemy Combat**

### The game shall allow enemies to use various weapons but only one at a time.

### The game shall allow the weapon type: pistol.

### The game shall allow the weapon type: shotgun

### The game shall allow the weapon type: assault rifle

### The tame shall allow the weapon type: laser.

### Each weapon will have different capabilities (bullets per shot, distance bullets travel,

### firing rate).

### The enemy weapon shall be dropped upon enemy death.

### The game shall allow enemies to kill a player.

## **Player Movement**

### An individual Player class shall move a player within the game.

### Players will have autonomous movement within the game.

### The player shall have the ability to run.

### Each player shall have the ability to jump.

### Each player shall have the ability to crouch.

### The player shall die if player falls into a hazard such as a pit.

## **Player Combat**

### The default weapon for a player will be a pistol.

### A player shall not possess more than one weapon at a time.

### The PC user shall be able to shoot a weapon by pressing a key on the keyboard.

### The mobile device user shall be able to shoot a weapon by touching a spot on the screen.

### The player shall be able to kill an enemy by shooting it.

### The player shall die if touched by enemy.

### The player shall die if shot by an enemy with one bullet.

## **Player versus Player Combat**

### The game shall allow players to attack other players by ranged attack (shooting).

### Player health shall be decrease if attacked by another player.

### The player’s health bar will decrease by one bar (max capacity is 3) per shot that makes contact with the player.

### The player will lose a life if their health bar is fully drained.

### The player will respawn at the same spot the player died if killed by other player.

## **Player completes mission**

### The end of a map level shall be designated by a door or portal.

### A player shall finish a map level by finding the exit portal and entering it.

### A new map level will be loaded when a player finishes a map level.

### A player shall be placed at the start of a map level when starting a new map level.

3.14.5 A player will start a new level with only the default weapon (a pistol).

## **Game input devices**

### On a personal computer the user shall be able to move the player using a keyboard.

### On a personal computer the user shall be able to move the player using a joystick.

### On a tablet the user shall be able to move the player using the touchscreen.

### On a mobile phone the user shall be able to move the player using the touchscreen.

### On a mobile phone the user shall be able to move the player using the keyboard.

## **Pausing the game**

### The game will have a pause system to temporarily halt game progress.

### For one-player games, the pause system shall have no timeout duration.

### For two-player games, players shall not have the ability to pause.

# **Nonfunctional Requirements**

## **Performance requirements**

## **Response time**

### The response time from the server shall be under 100 milliseconds.

### The server shall be able to handle a maximum of 100 real-time connections.

### The game should allow a minimum of 1 one-player and 1 two-player game to be played simultaneously.

### The game shall support a minimum of two users at the same time.

## **Software Quality Attributes - TBD**

## **Business Rules**

### This game is built strictly for demonstration purposes. This game is not licensed for use in any commercial, non-profit, or revenue-generating business activities by its developers.

### The game may not be distributed through any commercial online app marketplace.

### The game may not be copied by anyone without authorization by its creators.

# **Traceability Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technical Requirement ID** | **Test Case ID** | **Functional Requirement Description** | **Priority** | **Type** |
| 3.1.1 | 1 | In the User screen the input box for username will have system created username by default. |  | Interface |
| 3.1.2 | 2 | A User screen shall be the first screen displayed in the browser when the user types in the game URL and presses enter. |  | Functional |
| 3.1.3 | 3 | In the User screen the input box for username will have system created username by default. |  | Functional |
| 3.1.4 | 4 | The User screen shall have a selector to create a room for a one-player game. |  | Functional |
| 3.1.5 | 5 | The User screen shall have a selector to join a room for a two-player game. |  | Functional |
| 3.1.6 | 6 | Creating a room takes the player to the Selection screen. |  | Functional |
| 3.2.1 | 7 | In the User screen the player starts a one-player game by selecting the selector for creating a room. |  | Functional |
| 3.2.2 | 8 | After the selector for creating a room is selected the Selection screen appears. |  | Functional |
| 3.3.1 | 9 | In the User screen two players can start a two-player game by selecting the selector to join a room. |  | Functional |
| 3.3.2 | 10 | After the selector for creating a room is selected the Selection screen appears. |  | Interface |
| 3.4.1 | 11 | The Selection screen shall have a character select screen for each player. |  | Functional |
| 3.4.2 | 12 | The character select screen shall have an image of each character that can be selected by using an arrow button. |  | Functional |
| 3.4.3 | 13 | In the character select screen the image of the character will appear in color if there exists another player. |  | Interface |
| 3.4.4 | 14 | In the character select screen the image of the character will appear in black and white if there doesn’t exist another player. |  | Functional |
| 3.4.5 | 15 | In the Selection screen there shall be a button called “ready” to start the game. |  |  |
| 3.4.6 | 16 | In the Selection screen there shall be a “ready” button that will light up green after it’s clicked. |  | Functional |
| 3.5.1 | 17 | In the Selection screen the user starts a one-player game by pressing the “ready” button. |  | Functional |
| 3.6.1 | 18 | In the Selection screen the user starts a two-player game by pressing the “ready” button for a two-player game. |  | Interface |
| 3.7.1 | 19 | Once the game is started by pressing the “ready” button for either a one-player or two-player game, the game engine loads the level map. |  | Interface |
| 3.7.2 | 20 | After the game session has been initiated and the level map loaded, the dependent game libraries are loaded. |  | Interface |
| 3.7.3 | 21 | The main game screen shall briefly display “Level 1” which indicates the current level of the game. |  | Interface |
| 3.7.4 | 22 | The game shall display the health indicator of the player during game play. |  | Interface |
| 3.7.5 | 23 | The player character(s) shall appear at the start of the map with default weapon loaded |  | Functional |
| 3.7.6 | 24 | The game engine will load enemy sprites. |  | Interface |
| 3.7.7 | 25 | Interaction events between enemy sprites and players will wait to be triggered by player movement. |  | Interface |
| 3.8.1 | 26 | Enemy classes shall move enemy sprites within the game. |  | Interface |
| 3.8.2 | 27 | Enemies will have autonomous movement. |  | Interface |
| 3.8.3 | 28 | The game shall allow enemy movement dependent on enemy type. |  | Interface |
| 3.8.4 | 29 | Riflemen-type enemies shall not be allowed to move. |  | Functional |
| 3.8.5 | 30 | Soldier-type enemies shall be allowed to move (run and jump) toward players. |  | Functional |
| 3.8.6 | 31 | Static-type enemies shall be allowed to change direction of their weapon toward a player but othewise may not move. |  | Functional |
| 3.9.1 | 31 | The game shall allow enemies to use various weapons but only one at a time. |  | Interface |
| 3.9.2 | 32 | The game shall allow the weapon type: pistol. |  | Functional |
| 3.9.3 | 33 | The game shall allow the weapon type: shotgun |  | Interface |
| 3.9.4 | 34 | The game shall allow the weapon type: assault rifle |  | Functional |
| 3.9.5 | 35 | The tame shall allow the weapon type: laser. |  | Interface |
| 3.9.6 | 36 | Each weapon will have different capabilities (bullets per shot, distance bullets travel, firing rate). |  | Functional |
| 3.9.7 | 37 | The enemy weapon shall be dropped upon enemy death. |  | Interface |
| 3.9.8 | 38 | The game shall allow enemies to kill a player. |  | Functional |
| 3.10.1 | 39 | An individual Player class shall move a player within the game. |  | Functional |
| 3.10.2 | 40 | Players will have autonomous movement within the game. |  | Functional |
| 3.10.3 | 41 | The player shall have the ability to run. |  | Functional |
| 3.10.4 | 42 | Each player shall have the ability to jump. |  | Functional |
| 3.10.5 | 43 | Each player shall have the ability to crouch. |  | Functional |
| 3.10.6 | 44 | The player shall die if player falls into a hazard such as a pit. |  | Functional |
| 3.11.1 | 45 | The default weapon for a player will be a pistol. |  | Functional |
| 3.11.2 | 46 | A player shall not possess more than one weapon at a time. |  | Interface |
| 3.11.3 | 47 | The PC user shall be able to shoot a weapon by pressing a key on the keyboard. |  | Interface |
| 3.11.4 | 48 | The mobile device user shall be able to shoot a weapon by touching a spot on the screen. |  | Interface |
| 3.11.5 | 49 | The player shall be able to kill an enemy by shooting it. |  | Interface |
| 3.11.6 | 50 | The player shall die if touched by enemy. |  | Functional |
| 3.11.7 | 51 | The player shall die if shot by an enemy with one bullet. |  | Functional |
| 3.12.1 | 52 | The game shall allow players to attack other players by ranged attack (shooting). |  | Functional |
| 3.12.2 | 53 | Player health shall be decrease if attacked by another player. |  | Functional |
| 3.12.3 | 54 | The player’s health bar will decrease by one bar (max capacity is 3) per shot that makes contact with the player. |  | Functional |
| 3.12.4 | 55 | The player will lose a life if their health bar is fully drained. |  | Functional |
| 3.12.5 | 56 | The player will respawn at the same spot the player died if killed by other player. |  | Functional |
| 3.13.1 | 57 | The end of a map level shall be designated by a door or portal. |  | Functional |
| 3.13.2 | 58 | A player shall finish a map level by finding the exit portal and entering it. |  | Functional |
| 3.13.4 | 59 | A new map level will be loaded when a player finishes a map level. |  | Functional |
| 3.13.5 | 60 | A player will start a new level with only the default weapon (a pistol). |  | Functional |
| 3.14.1 | 61 | On a personal computer the user shall be able to move the player using a keyboard. |  | Functional |
| 3.14.2 | 62 | On a personal computer the user shall be able to move the player using a joystick. |  | Functional |
| 3.14.3 | 63 | On a tablet the user shall be able to move the player using the touchscreen. |  | Functional |
| 3.14.4 | 64 | On a mobile phone the user shall be able to move the player using the touchscreen. |  | Functional |
| 3.14.5 | 65 | On a mobile phone the user shall be able to move the player using the keyboard. |  | Functional |
| 3.15.1 | 66 | The game will have a pause system to temporarily halt game progress. |  | Functional |
| 3.15.2 | 67 | For one-player games, the pause system shall have no timeout duration. |  | Functional |
| 3.15.3 | 68 | For two-player games, players shall not have the ability to pause. |  | Functional |
| 4.2.1 | 69 | The response time from the server shall be under 100 milliseconds. |  | Functional |
| 4.2.2 | 70 | The server shall be able to handle a maximum of 100 real-time connections. |  | Functional |
| 4.2.3 | 71 | The game should allow a minimum of 1 one-player and 1 two-player game to be played simultaneously. |  | Functional |
| 4.2.4 | 72 | The game shall support a minimum of two users at the same time. |  | Functional |

# **Appendix A: Use Cases**

## Use Case 1

**Title:** Create a 1P game

**Actor:** Customer

Scenario:

1. Create a game session (room) in a room selection screen.
2. The system will generate the name of the room or session ID.
3. Character selection screen appears
4. User selects 1P
5. User selects character
6. User can input character name or use default name
7. User tells the system that he is ready to start the game.
8. Game session starts

## Use Case 2

**Title:** Create a 2P game

**Actor:** Customer

1. Create a game session (room) in a room selection screen.
2. The system will generate the name of the room or session ID.
3. Character selection screen appears
4. Users selects 2P
5. 1st player selects character
6. 2nd player selects character
7. Users can input character name or use default name
8. Users tell the system that they are ready to start the game.
9. Game session starts

## Use Case 3

**Title:** Create game session

**Actor:** Player

1. The game must load the map
2. The game must load all sprite classes
3. The game must load dependent game libraries (2D, sound, etc)
4. The user should see player character(s) appear to start the game.

## Use Case 4

**Title:** Killing an enemy

Prerequisite: game has started

**Actor:** Player

1. The default enemy can be killed by one bullet.
2. The default enemy can only run.
3. The shooting enemy can be killed by one bullet.
4. The shooting enemy can also shoot bullets.

## Use Case 5

**Title:** Enemy actions

**Actor:** Enemy

1. Enemy can run
2. Enemy can shoot
3. Enemy can jump (onto objects)

## Use Case 6

**Title:** Enemy actions

**Actor:** Enemy

1. Enemy can run
2. Enemy can shoot
3. Enemy can jump (onto objects)

## Use Case 7

**Title:** Killing a player

Prerequisite: game has started

**Actor:** Player

1. A player can be killed by one bullet
2. A player can be killed by an enemy touching him.
3. A player can be killed by falling into a pit.

## Use Case 8

**Title:** Player actions

**Actor:** Player

1. Player can run.
2. Player can shoot.
3. Player can jump (onto objects).
4. Player can pick up a dropped weapon.
5. Player can crouch.

## Use Case 9

**Title:** Player kills another player

**Actor:** Player

1. Player can shoot at other player
2. Player can be killed by three bullets from other player
3. A health bar will be displayed for each player during the game.
4. Health bar will decrease when shot.

## Use Case 10

**Title:** Player completes mission

**Actor:** Player

1. Player reaches the end of the map.
2. The end of the map is designated by a door or portal that he walks through.

## Use Case 11

**Title:** User action

**Actor:** User

1. User can move using a touch screen on a tablet
2. User can move using a keyboard on a personal computer.

## Use Case 11

**Title:** User can pause a game

**Actor:** User

1. User can pause a game for an unlimited time
2. User can then restart the game

# **Appendix B: Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Database | Collection of all the information monitored by this system. |
| Field | A cell within a form. |
| Member | A member of the Exe Club listed in the Exe database. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |