Software Requirements Specification

For

**Beta Game**

Version 1.0 approved

Prepared by **Beta Group**

7/5/2018

# Table of Contents

[**Table of Contents**](#_c265avnyn13b) **1**

[**Revision History**](#_bahlwydrb6ot) **2**,3,4

[**DEFINITIONS**](#_308s4mqyow7v) **4**,5,6,7

[**1. Introduction**](#_1plbg0gf94w) **7**

[1.1 Purpose](#_j2xu8yc58mrb) 7

[1.2 Document Conventions](#_kewqk7snccj3) 7

[1.3 Intended Audience and Reading Suggestions](#_pip1l8gytzyc) 7

[1.4 Product Scope](#_d2pqviawzgyg) 7,8

[1.5 References](#_e2xxfyrw8vnw) 8

[**2. Overall Description**](#_6vif77pqjy07) **8**

[2.1 Product Perspective](#_3z12aasgavpx) 8

[2.2 Product Functions](#_pt7andygq0i1) 8,9

[2.3 User Classes and Characteristics](#_jr061rxzp4tt) 9

[2.4 Operating Environment - started by Joel Munoz](#_xoh1zbipa42r) 10

[2.5 Design and Implementation Constraints](#_ud1pvlc7kavv) 10

2.6 User Documentation 10

[2.7 Assumptions and Dependencies](#_kx8otlen1sdb) 10,11

[**3. External Interface Requirements**](#_rmlw5ujxctw2) **1**1

[3.1 User Interfaces](#_is9h0qq7saqz) 11,12,13,14

[3.2 Hardware Interfaces](#_4mrn2xmhpjnd) 14

[3.3 Software Interfaces](#_mltpkqpbg328) 14

[3.4 Communications Interfaces](#_8o6mwniyrldw) 14,15

[**4. System Features**](#_5by97k7h5hlk) **15,16**,17,18,19

[4.1 Detailed Feature Descriptions](#_uz6juob42zdy)

[4.1.1 - P1: Priority Features](#_hi1qz5ufpsqi) 19,20,21

[4.1.2 - P2: Priority Features](#_ihaaxpaxm7o3) 22,23,24,25

[4.1.3 - P3: Priority Features](#_lkkmnfcrroe4) 25,26,27

[**5. Other Nonfunctional Requirements**](#_gj3v20nl4p8w) **28**

[5.1 Performance Requirements](#_u3nj7c4eab68) 28

[5.2 Safety Requirements](#_n5uvmpm97j26) 28

[5.3 Security Requirements](#_13qw6ihqcgfq) 28

[5.4 Software Quality Attributes](#_4zbi36u09iez) 28

[5.5 Business Rules](#_w08tusr0rqqt) 28

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Prativa Kafley | 7/5/2018 | Wrote - **1.1** Purpose,  **1.2**-Document Conventions,  **1.3**- Intended Audience and Reading Suggestions. | 1.0 |
| Prativa Kafley | 7/6/2018 | Wrote**- 1.4** - Product Scope ,  **1.5** - References,  **2.2**-Product Functions ,  **2.3**- User Classes and Characteristics. | 1.0 |
| Samuel Ruiz | 7/6/2018 | Edited - 1.1 | 1.0 |
| Gordon Man | 7/6/2018 | Wrote foundation terminology in the **Definitions**  Edited **2.4** - Operating Environment | 1.0 |
| Joel Munoz | 7/6/2018 | * Wrote - **2.1** Product Perspective | 1.0 |
| Jonathan George | 7/6/2018 | Wrote- External Interface Requirements **3.1,3.2** | 1.0 |
| Joel Munoz | 7/6/2018 | * Wrote - **2.4** operating environment | 1.0 |
| Jonathan George | 7/7/2018 | Edited Product functions **2.2**  Wrote User documentation **2.6**  Assumptions and Dependencies **2.7** | 1.0 |
| Samuel Ruiz | 7/7/2018 | Wrote - **4.0** System Features  Wrote - [**4.1** Detailed Feature Descriptions](#_uz6juob42zdy)  Wrote - [**4.1.1**  P1: Priority Features](#_hi1qz5ufpsqi)  Wrote - [**4.1.2** P2: Priority Features](#_ihaaxpaxm7o3)  Wrote - [**4.1.3** P3: Priority Features](#_lkkmnfcrroe4)  Edited - Definition Section  Added page numbers | 1.0 |
| Prativa Kafley | 7/7/2018 | **Wrote- 2.5**- Design and Implementation Constraints,  **2.7**- Assumptions and Dependencies,  **3.4**- Communications Interfaces, | 1.0 |
| Joel Munoz | 7/7/2018 | * Started **Section 3.3** * Revised 2.4 | 1.0 |
| Jonathan George | 7/8/2018 | Edited **3.3, 4.0, 4.1**  Proofread document. Version 1.0 | 1.0 |
| Prativa Kafley | 7/8/2018 | Wrote**-5.1**- Performance Requirements,  **5.2** -Safety Requirements  Edited- Definition Section,  Edited - **3.1,** Section 4  **Managed Table of content**:referred sections to their appropriate page numbers. | 1.0 |
| Gordon Man | 7/8/2018 | Wrote **Definitions**  Wrote **5.3** - Security Requirements  Wrote **5.5** - Business Rules  Edited **3.1** - User Interfaces  Edited **4** - System Features  Automated **Table of Contents**  Proofread document | 1.0 |
| Joel Munoz | 7/8/2018 | * Edited **section 3.3** * Started **section 5.4** * Combed over paper to find errors and conduct edits * Proofread document | 1.0 |

# DEFINITIONS

|  |  |  |
| --- | --- | --- |
| **Terms** | **Definitions** | **Initials of editor** |
| Main Menu | The main menu is the central starting point of the software. Once Game is launched, the first thing the users interact with is the main menu. From there the user can select what they would like to start doing.(Play the game, view controls/overview, view high scores, and view credits. No matter what option the user chooses, an option or prompt to return back to the main menu will be possible with the use of easy to locate buttons described in finer detail in section 3.1 | **JG** |
| Game Mode | Game mode refers to when the user clicks the Play button on the main menu to start the actual game play. | **GM** |
| User | The person that is using the computer in order to play the game that we are developing will be called the user. | **GM** |
| Player Object | An image rendered on the screen during game mode`. The image is currently a black circle that can be controlled with the W-key for up, S-key for down,A-key for left and D- key for right on the keyboard and space to shoot bullets vertically going up. | **GM**  **Edited by Prativa** |
| Enemy Object | An image on the screen that spawns from the top of the window going down, that moves in various directions such as up down left and right, who has the ability to shoot bullets that can collide with the player object. | **GM** |
| Bullet | An image rendered on the screen during game mode where the image that spawns from the player or enemy object that fired it. Direction and speed of the bullet may vary, but generically the bullet should fire in a vertical straight line towards the top on the screen if the user object fired a bullet, and to the bottom of the screen if an enemy object fired the bullet. Bullets have the ability to remove objects from the game once collision is detected. Bullets will also have the ability to reduce life of objects. | **GM**  **Edited by Samuel**  **Edited by Jonathan** |
| Collide/  Collision | The instance where a player object, enemy object, or bullet image occupies the same (x,y) coordinate plain within the game window. | **Edited by Jonathan** |
| Life | The unit of measure for how much damage (harm caused by an attack) the player or enemy object can take. Once an object’s life drops to zero, the object will be destroyed. | **GM**  **Edited by Samuel** |
| Destroyed | When a player object or enemy object’s life drops to zero, the rendered image for that object will disappear from the screen and be deleted from memory. | **GM**  **Edited by Samuel** |
| Shooting/ shoot | When the player or enemy object spawns bullets in order to destroy the player or enemy objects, this action is called shooting. | **GM**  **Edited by Samuel** |
| Attack | When a player or enemy object is shooting. | **SR** |
| Game State | When the game is in a particular condition at a specific time:   1. **Running State**: The game is said to be in running state, if and only if, the game is not in the pause state or the menu state. 2. **Pause State**: The game is said to in paused state, if and only if, the game state is in running state and the user selects the pause button and all the objects inside the displays stop being updated constantly. 3. **Menu state**: The game is said to be in the menu state, if and only if, the user is not in the running state nor in the pause state but it is currently in the game menu. 4. **Menu option state**: The game is said to be in the menu option state, if and only if, the user selected on of the following options form the game mane: Help, Settings, Credits, High Score. | **SR** |
| setFocusable | A boolean flag to indicate whether a component can get  the focus. | **PK** |
| Username | The name in which the user will also be known as inside of the high score board.  An example would be:  “High Score Board”   1. Gordon Man - 113,543   In this scenario, Gordon Man will be the username for the user with the high score of 113,543. | **GM** |
| Game Over / High Score Board | refers to the state after game mode ends due to the player object being destroyed. When the game is over, it should display the list of the top 10 highest scores next to the corresponding username of the user that obtained that score. The list of high scores is referred to the High Score Board. | **GM** |
| The Screen | The screen refers to the window of the computer in which the game will be ran. If something is being displayed on the screen, that means inside of the window that the user is playing the game, something is being displayed on that window. | **GM** |
| IEEE​ | A standard set of procedures used to write an SRS document. | **PK** |

# 

# 1. Introduction

Beta Game is meant to be a fun game about skilled movement, shooting, and timing. It Is a derivative of existing top down bullet hell shooter games found on digital game markets such as Steam and old school arcades. Beta Game is rated E for everyone to enjoy.

## **1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the Beta Game, an airplane-based shooter type arcade game and its expected functionality. This document will provide a clear understanding of the game to guide for the development of its Version 1.0. It includes explanation of the game with its purpose and features, interface, and working functionalities. This game is rated E for everyone.

## **1.2 Document Conventions**

All standards adopted in this document preparation is based on the IEEE SRS template provided to the class and uploaded in the blackboard. Higher level requirements are assumed to be inherited by detailed requirements provided in sections 3 and 4.

## **1.3 Intended Audience and Reading Suggestions**

The intended audience are members of the group Beta, the Software Engineering I class students for Summer 2018 and Professor Hank Stalica. However, all the information written within this SRS is usable for all future game developers as well as whoever trying to learn about the Beta Game game. To anyone who is trying to know the technical requirement and implementation of the game, it is recommended to focus reading sections 3 and 4.

## **1.4 Product Scope**

A significant objective/goal of this game is to create a top down shooter, where the player is in control of a player object and must survive by destroying enemy objects to score points before a timer runs out, or players life reaches zero. This experience is meant to be used for entertainment to anyone interested in playing, and to fulfill the class requirements for the CS 4310 Software Engineering I at CSU East Bay in Summer 2018 Quarter.

## **1.5 References**

<https://bb.csueastbay.edu/bbcswebdav/pid-4370958-dt-content-rid-48770922_1/courses/20183_CS_4310_01SP_1/20183_CS_4310_01SP_1_ImportedContent_20180614044154/SRS-2.pdf>

SRS document for “Top Down Space Shooter Video Game”.

<https://bb.csueastbay.edu/bbcswebdav/pid-4370958-dt-content-rid-48770921_1/courses/20183_CS_4310_01SP_1/20183_CS_4310_01SP_1_ImportedContent_20180614044154/doc.pdf>

SRS document for “Parental Monitoring and Control Application”.

Both of the sample SRS documents posted on the class Blackboard portal for this course has been referenced and has been a source of inspiration while preparing this SRS document.

# 2. Overall Description

## **2.1 Product Perspective**

The **Beta game**  application will be new self-contained product. This game application gives the user the ability to interact with keyboard and computer monitor. The user will open the game application, in which they will be greeted with a menu screen. The menu screen will display five buttons ***Play***, ***Help***, ***High score***, ***Credits*** and ***Quit*** (Refer to section 2.2). The User will enter game mode in which they will play as the **user**, whom will be in shooting combat against an **enemy** object (Refer to DEFINITIONS). The game application will consist of Graphical User Interface Module(Refer to section 2.4)

## **2.2 Product Functions**

* User will be welcomed in the main menu window
* The main menu window will have option to press the button for **Play*, Help, High Score, Credits and Quit***: The functions of the appropriate buttons are described below:
  + **Play** -> When This button is pressed, user will be taken into the game play screen where they will be able to:

1: control a spaceship using W,S,A,D keep to move around the screen.

2: The user will be able to shoot the enemy object using bullets with the help of player object by pressing spacebar.

3: If the user is shoot by the enemy object, life of user will be taken off.

4: destroy enemy ships with bullets to score points.

5. View score, remaining life total, and remaining time total at the top of the screen.

6:Game will end when the user loses all life or time interval of 5 minutes gets elapsed.

* + **Help** -> By pressing the help button user will be able to view tutorial (instructions) for how to play the game such as keyboard keys for movement, or how to score points.
  + **High Score**-> By pressing the High Score button player will be able to view top 10 highest recorded scores of the game.
  + **Credits** -> when the credits button is pressed, it will display a screen with the names of the Beta game creators. A pixel image of each creator along with their contributions.
  + **Quit** ->Exits the Software.

## **2.3 User Classes and Characteristics**

There will be one **player class** which will receive input from keyboard. Specific keys W, S, A, and D will be used to move the player object.

There will be one **enemy class** which will have a fixed pattern of movement and shoot player at random intervals from three spawn points on top of the screen each in the right,middle, and left. The movement pattern will be back and forth in both horizontal and vertical directions making it hard for players to hit and prevent itself from being dodged from bullets from the enemy object. Successful shoot of an enemy object by the player is identified when x-coordinate of the released bullet matches the coordinate of the enemy object.

Enemy objects spawning and attacks will be fairly slow enough for player to dodge and counter attack. A successful game termination is defined as a moment when 60 enemy objects are destroyed. In the meantime, game termination is accounted on factors like destroy of enemy objects in a certain time frame otherwise it is considered a game loss. Total permitted time will be 5 minutes to destroy 60 enemy objects where three enemy objects are spawned each 5 seconds.

Characteristics of both the objects should be pretty predictable as most of the users have experience playing similar kind of games and will not require much technical expertise for end user to play it. It would however, accompanied with a general guide with screenshots if applicable to facilitate users to learn how to play this game. Game scoring will be simple as (time remaining x 100 + 100 per enemy object destroyed) which will be unknown to users unless they refer to the user guide or closely analyse their score while playing whichever applicable.

**2.4 Operating Environment - started by Joel Munoz**

2.4.1 **Hardware**

The user will need to have a monitor, computer hard drive, keyboard and mouse. This will enable interaction with the user and game application.

2.4.2 **Graphical user interface Module**

The Graphical User Interface Module allows the user to control the player object through interaction of the keyboards pressed, allowing control over player object in game mode.

2.4.3 **Operating System**

The operating system within the computer hard drive will need to run either windows ( 7, 8, 10) or MacOS.

2.4.4 **Java 8**

latest version of Java 8 installed within their system. The latest version can be downloaded or updated from javas website <https://java.com/en/download/> .

2.4.5 **Integrated development environment**

The Integrated development environment (IDE) that is capable of running Java code, example being NetBeans, Eclipse and Atom.

## **2.5 Design and Implementation Constraints**

The potential constraint identified for developers will be time. Due to time restrictions, we will incorporate basic features in the game where liability to extend or add more complex functionalities will be part of of any improvised versions in future if required or desired. Different level of proficiency in Java programming and knowledge of Java classes with their supported features among team players is certainly a constraint for developers which has been accounted under time factor because a learning curve in the beginning has been considered as a vital part of the application development.

**2.6 User Documentation**

The user documentation will be provided in game inside the **Help** menu found at startup once user selects this option. Once there the user can read which keyboard input perform specific ingame functions, like movement or firing, as well as a brief overview of what to do to score points and the overall objective of the game.

## **2.7 Assumptions and Dependencies**

This product will require either a Windows operating system (any of Windows 7, 8, and 10) or mac OS with Java version 8 to run. It will require keyboard and mouse as an input device, with a hard disk of at least 10 GB, and a monitor for display. It is assumed that the user will have knowledge on how a computer system boots and can be started. Furthermore, it is assumed that the user will follow documentation to know about the used keys for playing the game.

# 3. External Interface Requirements

## **3.1 User Interfaces**

This section will discuss the user interface for Beta Game.  
  
 At initial start of Beta game, the users will see a Main Menu screen.   
  
**I. Main Menu Screen:**

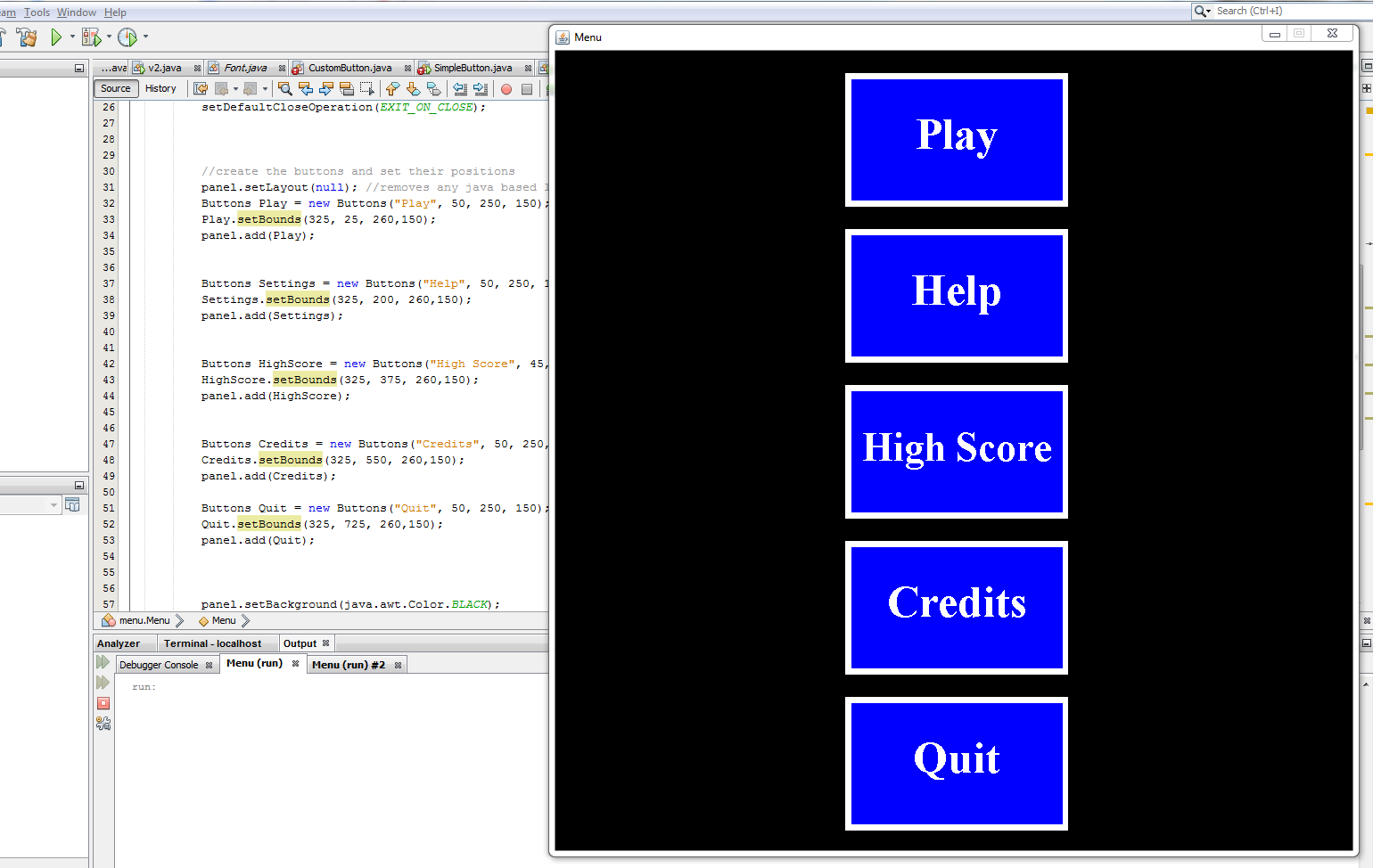
Upon opening the game, players will see the screen display with the name of the game, **Play** button, **Help** button, **High Scores** button, **Credits** button

And **Quit** button. *Below an example*

Beta Game  
 -> Play   
 Help

High scores  
 Credits

Quit



(screenshot courtesy of Gordon)

**II. Play Game screen**  
   
A) Game Screen

● On the Top Left Side of the Screen  
 ○Player life bar

● On the Top Right Side of the Screen  
 ○Current Score

● On the Top Middle Side of the Screen  
 ○Timer

● The rest of the screen will where the game play takes place

B) End Game Screen   
 ● Name entry field for user to write their achieved score to file  
 ● Prompt Screen  
 ○Main Menu option  
  
III. Help Screen  
  
A) Controls  
 ● displays controls(keyboard inputs for player object movement and bullet fire)  
  
B) objective   
 ● informs user of game objectives and ways to score points

C) return to main menu

● Menu Button (takes you back to Main Menu)

IV. High Score  
 ● Display on screen the names and scores of the 10th highest player scores, starting with highest on top and descending to lowest score at the bottom.   
 ● Menu Button (takes you back to Main Menu)  
  
V. Credit Screen  
 ● Display on screen the names of the creators of Beta game with graphics of there portraits next to their name and title.   
 ● Menu Button (takes you back to Main Menu)

User will be able to press the (W,S,A,D) keys to move the player object around the screen. Pressing spaceBar key will update graphics on the screen and display bullet projectiles that will originate from the layer object to , as well as pointing and clicking the mouse in the game window.

## **3.2 Hardware Interfaces**

This product supports any device that meets the following requirements:

● Hard drive - Must have at least 10GB of storage space to store the main game, custom graphics, sound, and txt files for high scores.

● Keyboard - Our game will be implementing java.awt.event.KeyAdapter, and

java.awt.event.KeyEvent libraries to translate keyboard actions into movement and action commands like bullet fire for the user so the java.awt.graphics may be updated.

● Monitor - This allows the user to view the game.

● Speakers(optional) - This allows the user to hear the game.

## **3.3 Software Interfaces**

**Beta Game** will interact with the software through a JFrame windows and a GUI created by Beta group. Netbeans IDE (see section 2.4.) can be used to launch the software. Javax.swing.JFrame header was used to import and enable the capability of JFrame window. Based on the Keys pressed the GUI will pass the user inputted information and process through the software to display the proper output to the Jframe.

**Menu**:

* JFrame window with four GUI buttons (see section 3.1). Each button will refresh the current window and take the user to a different part of the game listed below. .

**Play**:

* Updated window composed of GUI that enable interaction with user and player in game mode (See section 3.1).

**Help**:

* Updated window with printed information on game play instruction

**Score Board**:

* Updated window with printed results of top scoring users gaming results (see section --)

**Credit**:

* Updated window with information displayed on the **Beta Game’s** creators.

**Quit**:

Exits the Software.

## **3.4 Communications Interfaces**

This game will not require any of the online communication features and protocols. There will be a simple form for user to input his/her name for tracking scores in High Score if the user can achieve a high enough score which can be referenced in High Score menu (3.1 part IV). Communication between the user interface and the underlying disk file for scores will be based on I/O streams using Java classes.

# 4. System Features

**Priority**:

1. Essential will be displayed with P1 in red color (this means that software is not acceptable without meeting this requirement).
2. Conditional will be displayed with P2 in orange color (this means these requirements enhance the software, but software is still acceptable without them).

3. Optional will be displayed with P3 in blue color (this means that these requirements may or may not be worth doing).

|  |  |  |
| --- | --- | --- |
| **P1 – Priority Features** | **Description** | **Name** |
| Display Main menu | Display options: play, High Score, credits, help, and settings. | Jonathan |
| Game menu - Play option if pressed by mouse | If user presses the play button, then the game mode will start. | Jonathan |
| Allow for user mouse input | User can use mouse to select options in menu | Jonathan |
| Display game | Display the Health, Score, Timer, player object, and enemy objects in the screen. | Jonathan |
| Player Movement | User should be able to move player object with keyboard buttons: W,S,A,D | Jonathan |
| Enemy object movement | Enemies object will be able to move (up, down, left, right and diagonally) | Samuel |
| Destroy enemy object | Once the life of the enemy decrease to zero, remove enemy object from the window. |  |
| Enemy object attack | Enemies object will try to attack (by shooting or collision) with the player object. If the player object gets hit (by an enemy bullet or collision), decrease player object life. | Samuel |
| Score | In the top center of the screen the current score will appear as digits ranging from **0** - **2,147,483,647** depending on how well the players is doing | Jonathan |
| Timer | Countdown timer starting at 5 minutes down to zero seconds | Jonathan |
| Enemy Spawn | Once the player object destroys all the enemy objects on the screen, more enemy objects will appear on the screen from 3 specific locations on the top of screen (left, middle and right). Spawn will be of three enemy objects in every five seconds even if player has not destroyed all enemy objects. | Edited by Prativa |
| Death of player object | If the players object life decreases to zero.  The Player Object image will disappear from the Game Mode. | Edited by Gordon |
| Game Over | Display a message “Game Over” to the screen when the player object’s life decrease to zero and the death of player object has occurred. | Edited by Gordon |
| High Score Board | When the user clicks on the High Score Board Button on the main Menu, a list of the top ten user will be listed based on their scores, with the highest being number 1 and lowest being number 10. | Gordon |
| Username Prompt for High Score | When player object has been destroyed, Game Over message will be displayed, and if the user scored a score within the top ten highest scores, the user will be prompted to enter their username to be added into the high score board. | Gordon |

|  |  |  |
| --- | --- | --- |
| **P2 – Priority Features** | **Description** | **Name** |
| Move menu selector with arrow keys | User can select options in game menu using keyboard |  |
| Game menu - Scoreboard option | If the user selects High Score a list of players with the top scores will appear. |  |
| Game menu - Credits option | If the user selects credit a list of the creators that design the game application will appear, as well as their contribution to the Beta Game application | Edited by Joel |
| Game menu - Help option | If the user selects Help a tutorial page will appear explaining the user the functionality of each buttons are, and how they can be used during gameplay.  · Which keyboard are used for movement:  W - move upward up, S - move downward, A- move towards left and D - move towards the right.  · What the meaning of each item in the screen: life, score, level.  · Learn about the different types of enemies are their characteristics. | Edited by Joel |
| Restart game | After the player object is destroyed and Game Over is displayed, the game will then prompt user to restart game or go back to main menu via message buttons on the screen. | Edited by Gordon |
| Back to game menu | If the user select High Score, Help, Credits or Settings display a back button so user can go back to game menu. |  |

|  |  |  |
| --- | --- | --- |
| **P3 – Priority Features** | **Description** | **Name** |
| Two players | This feature allows two users to play in the same game at the same time as two separate player objects both with the same mission. | Edited by Gordon |
| Pause Game - Pause Menu | If the user presses the pause button then pause the game and display pause menu |  |
| Pause Game – Quit | If the user while the game is pause selects “Quit” then display the main menu |  |
| Pause Game – Resume | If the user while the game is pause selects “Resume” then unpause the game |  |
| Change Controllers | Allows the user to change the default control keys to whatever they like. |  |
| Change player sprite | The user can change the player sprite (default is an airplane) to anything they want by importing an image (.png or .jpg). |  |
| Sound Effect | When an enemy object is destroyed (meaning that the sprite is remove from the display because the life of the enemy object is equal to zero) then produce explosion sound.  When the player object is destroyed (meaning that the sprite is removed from the display because the life of the player object is equal to zero) then produce explosion sound. |  |
| Game menu - Settings option | Add an additional button in main menu, where If the user selects the setting options he/she will be able to change the music or mute the music. |  |
| Implementation of items | Allow enemy objects to drop items when destroyed so that the player object can collect these items to gain extra features such as strong and faster bullets or homing missiles that also shoot out of the player object to assist in destroying enemies. | Gordon |
| Implementation of special skills | Allows user to choose from multiple player objects, each with their own corresponding special skill such as faster movement but weaker bullets, or slower movement but stronger bullets, or an invincibility trait where if the user presses on a key on the keyboard, allows player object to not lose life points when hit or collides with enemy objects for X amount of seconds. | Gordon |

## **4.1 Detailed Feature Descriptions**

### **4.1.1 - P1: Priority Features**

4.1.1.1 - Display game

a) Description

The life, score, timer, player object and enemy objects will be rendered on the screen. The game will automatically start and enemy objects will start moving and shooting at the player object.

b) Stimulus/Response Sequence

After the user selects the “Play” option from the menu options the Game state will switch from menu state to game play state.

c) Functional Requirements

The user must have a keyboard and mouse to interact with the main menu or player object. If the keyboard or mouse is not connected the game will prompt the user “A keyboard is required to play the game” and wait until the keyboard is reconnected.

If the game fail to display sprites or the folder is not found, display to the user “Oops! Something went wrong. Unable to find X in folder Y.”

4.1.1.3 - Player object movement

a) Description

This allows the user to move the player object that is display on the screen while the game state is in running. The player will be able to move up with key “W”, down with key “S”, left with key “A”, and right with key “D”.

b) Stimulus/Response Sequence

The game will automatically start after the user press the play button form the main menu with use of a mouse and then the player object will spawn in the center of the screen. The user may choose to move either left, right, up or down, and shoot with use of the keyboard.

c) Functional Requirements

The user must have a keyboard to interact with the player object. If the keyboard is not connected the game will prompt the user “A keyboard is required to play the game” and wait until the keyboard is reconnected. The player object won’t be able to receive directions if the user doesn’t have a keyboard. The player object must begin moving in that direction with-in a 1/60 of a second to feel responsively to the player. If there is a lag then the game won’t feel natural.

4.1.1.4 - Enemy object movement

a) Description

This feature allows the enemy to move side to side, diagonally, up and down, square movement, right triangle movement and randomise movement. The enemies will constantly be in motion.

b) Stimulus/Response Sequence

Enemy objects that are hit by the bullets of the player object will diminish their life and eventually be removed and points will be awarded(score will be updated).

c) Functional Requirements

Enemy objects will be moving at different speeds and different directions. Once an enemy collides with player object their life will diminish.

4.1.1.5 - Score / Level

a) Description

In the top right of the screen the current score will appear.

b) Stimulus/Response Sequence

The score keeps incrementing as the player object keeps destroying the enemy objects.

c) Functional Requirements

The score and timer should increment, and decrement without any issues.

4.1.1.6 - Destroy enemy object

a) Description

This feature will allow the user object to increase the score once an enemy is destroyed. Depending on what type of enemy type was destroyed the player score could increase by 1, 5, 10 or 20.

b) Stimulus/Response Sequence

After the enemy sprite is removed from the screen because it was destroyed by the user, increase score by the amount of that enemy type.

c) Functional Requirements

The bullets of the destroyed enemy can affect the player object even though the player had already destroyed it.

4.1.1.7 - End Level

a) Description

This feature is what makes the game interesting. It is very important for a player to have a goal. This ensure that the player stays engaged in the game. The timer and enemy spawn rate will insure constant difficulty as the game will flood the screen with objects to destroy and timer to beat. The difficulty of enemy object’s should increase(speed of object, and life) after every 10 enemies are destroyed.

b) Stimulus/Response Sequence

if the user clears screen by destroying all the enemy objects in the game, then more spawn.

c) Functional Requirements

If the player object manages to destroying all the enemy objects , more enemy object will appear thus making the game always interactive and challenging

4.1.1.8 - Death of player object

a) Description

If there is collision between an enemy object bullet and the player object or an enemy object and a player object, if either or both of these actions are true, then player objects life will decrease. And life bar will update and shrink in size. If life ever reaches equal or below zero.then game over message will appear, game play will stop.

b) Stimulus/Response Sequence

The player object will disappear from the display and the enemy will be highlighted as a way to show the user how the player object life reached equal orbelowl zero.

c) Functional Requirements

The enemy should be displayed once the player object is destroyed.

4.1.1.9 - Game Over

a) Description

It will display a text message “Game Over” to the user when the user is destroyed.

b) Stimulus/Response Sequence

This will show after the player object’s life is equal to or below zero.

c) Functional Requirements

The text message “Game Over” should only be display once the user’s life equal zero. The player object should not be able to move and the sprite should disappear.

### 

### **4.1.2 - P2: Priority Features**

4.1.2.1 - Game menu

a) Description

When game state is currently in the menu state then display the following options: Play, High Score, Credits, Help, Quit, and settings(optional). Wait for user to move to the highlighted box by using the keyboard as input (available key options - up or down) or wait for user to hover over the button with their mouse and left click. The user can use the enter button to proceed with the desired option. The game will change its state depending on what the user selected.

b) Stimulus/Response Sequence

When the software(Beta Game) is launched, the default state is the menu state. Then the first option that is going to be pre-selected is play. During this state the game will be waiting for user keyboard or mouse action.

c) Functional Requirements

The user must have a keyboard and mouse to interact with the player object and menus. If the keyboard or mouse is not connected the game will prompt the user “A keyboard and mouse is required to play the game” and wait until the keyboard or mouse is connected.

4.1.2.2 - Move menu selector

a) Description

This feature allows the user to interact with the game menu by selecting the various options. The player will use the arrow keys to move up and down and the enter button to select the desired option form the menu or simply use their mouse to hover over and left click the option they want. .

b) Stimulus/Response Sequence

While the game state is in menu state the user can select among the different options. The default option that will be selected is the play button but the user can select any other option

c) Functional Requirements

The user must have a keyboard and mouse to interact with the menu options. If the keyboard or mouse is not connected the computer it will prompt the user “A keyboard and mouse is required to make a selection” and wait until the keyboard or mouse is re-connected.

4.1.2.3 - Game menu - Play option

a) Description

If the game state is menu state and the user selects play from the menu the game will automatically start and enemy object will start moving and shooting.

b) Stimulus/Response Sequence

Once the user clicks the option to play the game fromm the main menu then the health of the player object is 100, the score is 0, timer set to 5 minutes, player object and enemy objects will be rendered on the screen.

c) Functional Requirements

The user must have a keyboard to interact with the player object. If the keyboard is not connected the game will prompt the user “A keyboard is required to play the game” and wait until the keyboard is reconnected.

If the game fail to display sprites or the folder is not found, display to the user “Oops! Something went wrong. Unable to find X in folder Y.”

4.1.2.4 - Game menu - Scoreboard option

a) Description

Scoreboard is an option from the menu. It will display to the user the top ten scores. Furthermore it will provide the name of the person who actually got the score.

b) Stimulus/Response Sequence

The scoreboard information should be displayed once the user selects the scoreboard option from the menu options. When the user is in the scoreboard page the game state will be change to menu option state and then there will be a button that the user can select to go back to the Menu state.

c) Functional Requirements

The user should not have any problems/interruptions with displaying the help page on their monitor. It’s required that all options should display within 2 seconds of any key pressed from the keyboard or mouse.

4.1.2.5 - Game menu - Credits option

a) Description

Credits is an option from the menu. It will display to the user the developers that work on creating the game. Furthermore it will provide the name, a picture of the person and what part they did.

b) Stimulus/Response Sequence

The credits information should be displayed once the user selects the credits option from the menu options. When the user is in the scoreboard page the game state will be change to menu option state and then there will be a button that the user can select to go back to the Menu state.

c) Functional Requirements

The user should not have any problems/interruptions with displaying the credits page on their monitor. It’s required that all options should display within 2 seconds after option is selected from the main menu.

4.1.2.6 - Game menu - Help option

a) Description

Help is an option from the main menu. It will provide instruction to the user and how to play the game. Furthermore there will be a description of the key controls, enemies and their movement.

b) Stimulus/Response Sequence

The Help information should be displayed once the user selects the Help option from the menu options. When the user is in the help page the game state will be change to menu option state and then there will be a button that the user can select to go back to the Menu state.

c) Functional Requirements

The user should not have any problems/interruptions with displaying the help page on their monitor. It’s required that all options should display within 2 seconds after option is selected from the main menu.

4.1.2.7 - Game menu - Settings option

a) Description

Settings is an option from the main menu. It will provide the user control over the sounds, music playlist and change of controls.

b) Stimulus/Response Sequence

The setting’s option once the user selected from the menus options will display the default setting of the game if the user hasn’t already change them. If the user already change the default value then displays what that user selected previously. When the user is in the settings page the game state will be change to menu option state and then there will be a button that the user can select to go back to the Menu state.

c) Functional Requirements

The user should not have any problems/interruptions with displaying the settings page on their monitor. It’s required that all options should display within 2 seconds after option is selected from the main menu.

4.1.2.8 - Restart game

a) Description

After the game is over the player can restart the game.

b) Stimulus/Response Sequence

When the player object’s life is equal to or less than zero it means that the player was destroyed. Display to the user the options to return to main menu or restart the game.

c) Functional Requirements

The user should not have any problems/interruptions with displaying the restart game menu on their monitor.

4.1.2.9 - Back to game menu

a) Description

The user has the ability to go back to the main menu.

b) Stimulus/Response Sequence

If the game state is currently in menu option state the user has the ability to go back to the main menu.

c) Functional Requirements

The user should not have any problems/interruptions with displaying the back to game button on their top left display conner.

### **4.1.3 - P3: Priority Features**

4.1.3.1 - Two players

a) Description

This feature allows to player objects to the display on the screen and controlled by two different users. This can be set up in the settings page. The user can select between single player mode and multiplayer mode.

b) Stimulus/Response Sequence

If the user selects multiplayer mode and the user later press play the game state will switch from menu state to running and the new player will be ask to press the enter key. This will allow the second player to control the second player object. The game will continue if either player is alive.

c) Functional Requirements

The second player key controls must be different that the key controls from the first player. The programs has to be multithreaded in order to both player to play at the same time.

4.1.3.2 - Pause Game - Pause Menu

a) Description

If the game state is currently running state the user can change it from running to pause state by clicking the pause button.

b) Stimulus/Response Sequence

The game will remain in pause state for as long the user wants. While the game is in pause state a pause menu will appear. In this menu the user can either resume game and the game status changes from pause state to running state or the user can select quit and the game state will change from pause state to menu state. During this time the player’s life, level and score will not increase nor decrease and they user won’t be able to move the player object.

c) Functional Requirements

In the pause state the bullets, enemy and player object, the timer must stay still in the screen. The game must completely pause at this time.

4.1.3.3 - Pause Game – Quit

a) Description

This allow the user to end the game. This will change the game state from pause state to menu state.

b) Stimulus/Response Sequence

After the user has click the pause button the pause menu will appear display the user two options: Quit or Resume. If the player selects quit the screen will display the main menu and the game state will change to menu state.

c) Functional Requirements

If the player doesn’t make a selecting the game will wait until option is made by the user.

4.1.3.4 - Pause Game – Resume

a) Description

This will allow the player to unpause the game (meaning that the game state changes form pause state to running state).

b) Stimulus/Response Sequence

I the use clicks the button “Resume” the game will unpause and it will resume where it started. The bullets, enemy object and timer will continue updating or moving in the screen and the user regains back the controls for the player object.

c) Functional Requirements

The game state must change from pause state to running state allowing the user to control the player object.

4.1.3.6 - Change Controllers

a) Description

This feature will show under the settings page. This will allow the user to change the default control keys W-key for up, S-key for down, A-key for left and D- key to any they like.

b) Stimulus/Response Sequence

In this settings page the user will be asked to press the various keys for the player object movement (up, down, left, right). The instructions will be display on the screen.

c) Functional Requirements

The player can only select alphanumeric keys. The game should provide error validating to guide the user.

4.1.3.7 - Change player sprite

a) Description

This feature allow the user to upload their own images (.png and .jpg) and this will provide a more custom and joyful experience. I will be provided in the settings page.

b) Stimulus/Response Sequence

The user will add it own image with a specific name in a specific directory so that the game would load it for the user when he/she in running state.

c) Functional Requirements

The images must be a certain demainsion (32X32) in order to render correctly. The details and rules will be provided in the settings page.

4.1.3.8 - Sound Effect

a) Description

Produce sound when to notify user that the enemy was destroyed successfully. Produce sound the the player object’s life equals zero.

b) Stimulus/Response Sequence

When an enemy object is destroyed (meaning that the sprite is remove from the display because the life of the enemy object is equal to zero) then produce explosion sound.

When the player object is destroyed (meaning that the sprite is removed from the display because the life of the player object is equal to zero) then produce explosion sound.

c) Functional Requirements

If the game fail to load the sounds or the folder is not found, display to the user “Oops! Something went wrong. Unable to find X in folder Y.”

# 5. Other Nonfunctional Requirements

## **5.1 Performance Requirements**

Beta game will require a single user to start the game with no options for multiple players playing in parallel. The game will require optimal system requirements for operating systems either Windows 7/8/10 or mac OS with Java Runtime Environment (JRE). System will respond to user’s input keys almost promptly. But it still has performance of 99.9% leaving the remaining unknown despite using setFocusable(refer to DEFINITIONS ) for keystroke events in java frame, which is all limitations of the Java development platform.

## **5.2 Safety Requirements**

We do not have any specific safety requirement for user but we suggest them not to play indefinitely to make it addictive. While referring to the application itself, we suggest user not to delete the score text file created on their computer. A delete of that file will result in loss of historical log of user and the score achieved by them.

## **5.3 Security Requirements**

User should not modify product source code in any way.

User should know that the high score and username will be stored in a text file created on their computer. (created in the same folder as the application by default)

## **5.4 Software Quality Attributes**

Refer to section 2.4 for in depth details of hardware, operating system and latest Java required for the game. The **Beta group** will continue to build and example the games function before summer 2018 quarter is over. The **Beta Group** goal is to provide a simple 2D shooting game with easy to understand game play and keyboard functions. Hopes for the **Beta Group** is to have an application that the user will find fun and exciting.

## **5.5 Business Rules**

User should know how to operate a computer.

User should not enter profanity as their username.

User should refrain from playing obsessively.

The game is rated E for everyone.