|  |
| --- |
| Software Testing  Test Planning Project |
| GitHub Repository  GitHub Repository  GitHub Repository  GitHub Repository  <https://github.com/LuqmanFarooq/Software-Testing-Test-Planning-Project>  <https://github.com/LuqmanFarooq/Software-Testing-Test-Planning-Project>  <https://github.com/LuqmanFarooq/Software-Testing-Test-Planning-Project>  <https://github.com/LuqmanFarooq/Software-Testing-Test-Planning-Project>  Muhammad Luqman  G00353385  Muhammad Luqman  G00353385  Muhammad Luqman  G00353385  Muhammad Luqman  G00353385 |



****

Table of Contents

[INTRODUCTION 3](#_Toc40381631)

[Objectives and Tasks 5](#_Toc40381632)

[Scope 5](#_Toc40381633)

[TESTING STRATEGY 7](#_Toc40381634)

[Description of Test Environment 7](#_Toc40381635)

[Unit Testing 7](#_Toc40381636)

[Main Menu Unit Tests 7](#_Toc40381637)

[Pause Menu Unit Tests: 10](#_Toc40381638)

[Gameplay Unit Tests: 11](#_Toc40381639)

[System and Integration Testing 13](#_Toc40381640)

[Integration Testing Definition: 13](#_Toc40381641)

[Methodology: 13](#_Toc40381642)

[System Testing Definition: 13](#_Toc40381643)

[Methodology: 13](#_Toc40381644)

[Performance and Stress Testing 14](#_Toc40381645)

[Performance Testing Definition: 14](#_Toc40381646)

[Methodology: 14](#_Toc40381647)

[Stress Testing Definition: 14](#_Toc40381648)

[Methodology: 14](#_Toc40381649)

[User Acceptance Testing 15](#_Toc40381650)

[Definition: 15](#_Toc40381651)

[Methodology: 15](#_Toc40381652)

[Beta Testing 15](#_Toc40381653)

[Methodology: 15](#_Toc40381654)

[TEST SCHEDULE 16](#_Toc40381655)

# INTRODUCTION

#### Project Overview

This game is a 2D side-scrolling platformer, inspired by the likes of ‘Salt and Sanctuary’, ‘Shovel Knight’, and ‘Fancy Pants’, with elements from ‘Skyrim’ (mainly in the way the player character and enemy characters attack). The artwork is inspired mainly by Shovel

Knight, which uses mainly pixel art to create its characters and world. The gameplay is inspired by ‘Salt and Sanctuary’ and ‘Dark Souls’ and ‘Skyrim’, which will see the player navigate progressively difficult levels with a wizard type character that uses magic a lá ‘Skyrim’. Each level has several enemies that the player must defeat to progress. Each level also has a boss that the player must defeat to progress to the next level. Each level contains pickups for the player, such as health pickups to replenish the player’s health.

The game will have three options on start-up: ‘Play’, ‘Settings’, and ‘Exit Game’.

Selecting ‘Play’ will take the player into the game and the player will begin at Level 1.

If a save system can be implemented, the player will begin at their last saved

point. ‘Settings’ will allow the player to edit game settings, such as sound level and

music level. ‘Exit Game’ will quit the application’. OPTIONAL: Include a ‘Load Save’,

‘Save Game’, and ‘Delete Save’ option. This will allow the player to create multiple

save files and be able to choose which one to load.

The game will include several options when the game is paused, similar to those

available on start up. The player will be able to resume the game, access settings,

restart the level and exit the game. The player can resume the game by selecting the

appropriate option or simply pressing the assigned button for pausing/resuming the

game. The player can access the settings and can adjust the sound level and music

level, for example. Choosing to restart the level will reset the entire level. This entails

resetting the player’s position to where they were when they first started, resetting

the enemies’ position and resetting any and all pickups.

Basic controls:

|  |  |  |
| --- | --- | --- |
| **Action** | **PC** | **Mobile** |
| **Move** **Forward** | Right arrow key/D | Arrow on screen |
| **Move Backwards** | Left arrow key/A | Arrow on screen |
| **Jump** | Up arrow key/W | Arrow on screen |
| **Attack** | Left mouse click/R | Dedicated button |
| **Pause/Resume** | Spacebar | Button in top right of screen |
| **Crouch** | C | Arrow on screen (hold) |

**The Game:**

Once the player has opened the game, they will be presented with three options:

‘Play’, ‘Settings’, and ‘Exit Game’. Selecting ‘Play’ will take the player into the game

and load the first level. The game will start immediately. From here, the player can

progress through the level and once completed and once completed, the next level

will load. At the start of the first level, text will appear on-screen informing the player

of the control screen. The game will feature at least three levels, with each increasing

in difficulty. This could range from having more and more enemies in the progressive

levels, enemies having more health, the player character having less enemies, etc.

Once the player has completed all the levels, the player will be presented with the

option to either start again from the first level or quit the game. If the player chooses.

‘Settings’ instead, they can instead adjust the sound level or music level. ‘Exit Game’ will close the game. Once inside the game, the player can pause the game at any time. From here, they can access a number of options: resume the game, access settings, restart the level, or exit the game.

# Objectives and Tasks

#### Objectives

The software test will ensure that our software is free from input errors and all the game logic will function as intended. For example:

After the game informs the player/user that there are no more waves of enemies, there should not be any more strains of enemies coming to attack the player

enemies and other entities on screen should not behave erratically and all positioning and attack scenarios should be accounted for

pickups work error free and functions as intended.

the player should only be able to “click on” certain elements in the user interface, with extraneous clicks not serving any purpose and performing no action.

# Scope

#### General

* Main Menu Functions.
  + art (world, texture etc.)
  + music, audio and the sound effect.
  + title screens.
  + Buttons and Functions.
* Pause Menu Functions.
  + art (world, texture etc.)
  + music, audio and the sound effect.
  + Buttons and Functions.
* Gameplay Functions.
  + animation (the like and quality of the realism, movement, and frame rate)
  + art (character model, terrain or world, texture, objects, etc.)
  + game logic and flow.
  + scene/level.
  + the action properties.
  + the player properties.
  + the situation to advance to the subsequent level.
  + increasing levels of difficulty
  + the scoring
  + the AI logic (for both offensive play and defensive play; player positioning and movement).
  + statistics (pre-game and in-game like high score and player statistics).
  + the gamepad.
  + the usability of the button functions.

#### 

# TESTING STRATEGY

## Description of Test Environment

The program will be compiled and tested on a PC with the Windows 10 Operating System as well as a Mobile running android. The compiler used will be the compiler included in the Unity cross-platform game engine. The test environment will be the same environment in which the software is guaranteed to operate - the only slight difference is that the end user will not need to compile and run their game out of Unity.

## Unit Testing

**Definition:**

Testing the unit components of the games such as play button, save game button and controls etc.

### Main Menu Unit Tests

**Participants:** Team 1

#### TEST 1

**Test Objective:** Test whether the "Play Game" button on the Main Menu functions appropriately.

**Test Description:** The "Play Game" button will be clicked by the tester and they will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the Game has loaded appropriately.

**Test conditions:** see test environment

**Expected Results:** This test will show the Game’s first level on the screen, text will appear on-screen informing the player of the control screen. The proper number of players lives, and boss lives displayed to the player at the start and the enemies will begin to assault the player as normally.

#### TEST 2

**Test Objective:** Test whether the "settings" button functions appropriately and that its contents function appropriately - i.e., the option to Control levels of sounds/music.

**Test Description:**  The "settings" button will be clicked by the tester and they will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the settings screen appears and contains the proper buttons, allowing the player to edit game settings, such as sound level and music level and ensure they function as desired.

**Test conditions:** see test environment

**Expected Results:** This test will show the contents of the settings screen, the button for the option for controlling levels of the music and sound effects.

**TEST 3**

**Test Objective**: Test whether the "Load Game" button functions appropriately when no prior game exists

**Test Description:** When the "Load Game" button is clicked on the Main Menu, the tester will ensure that " Load Game" does nothing when there IS NO prior saved data and the button will be grayed out.

**Test conditions:** see test environment

**Expected Results:** If no prior saved game data exists, then the "Load Game" button will be grayed-out and the button should serve no real purpose other than to inform the player that no prior saved game data exists.

#### TEST 4

**Test Objective:** Test whether the "load game" button functions appropriately when prior saved game data does exist

**Test Description:** The "Load game" button will be clicked by the tester and they will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the level that is presented will be the appropriate level as per the saved game data; note: this will vary depending on when the user's game is saved, so no game data is currently available. for this test, the user will first play the game, then have the game save her progress to the data file, and then engage in this test, test 4

**Test conditions:** see test environment

**Expected Results:** This test will show the appropriate level on the screen with the proper number of lives available to the player at the start and the enemies will begin to assault the kingdom as normally. the level is dependent upon the saved game data that is available at the time of this test.

#### Test 5

**Test objective:** Test that the “Exit Game” button on the main menu exits the game properly

**Test Description:** The tester will click the “Exit Game” button on the main menu and the game will exit without any errors

**Test conditions:** see test environment

**Expected Results:** Command will return once more to the operating system and the game will have exited with no errors.

**TEST 6**

**Test Objective**: Test whether the "Delete Game" button functions appropriately when no prior game exists

**Test Description:** When the " Delete Game" button is clicked on the Main Menu, the tester will ensure that " Delete Game" does nothing when there IS NO prior saved data and the button will be grayed out.

**Test conditions:** see test environment

**Expected Results:** If no prior saved game data exists, then the " Delete Game" button will be grayed-out and the button should serve no real purpose other than to inform the player that no prior saved game data exists.

#### TEST 7

**Test Objective:** Test whether the " Delete Game " button functions appropriately when prior saved game data does exist

**Test Description:** The " Delete Game " button will be clicked by the tester and they will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the level that is selected is deleted, so no game data is currently available. for this test, the user will first play the game, then have the game save her progress to the data file, and then engage in this test.

**Test conditions:** see test environment

**Expected Results:** This test will Delete saved game data that is available at the time of this test.

### Pause Menu Unit Tests:

**Participants:** Team 2

**TEST 1**

**Test Objective:** Test whether the " Save Game " button functions appropriately.

**Test Description:** The tester will ensure the game saves the player's progress correctly by playing the game's level, going back to the main menu, and selecting "Load game" button

**Test Conditions:** see test environment

**Expected Results:** The game data loaded when the player selects "Load Game" from the main menu will be the same game data (and the game will have been saved in the correct spot) that the player saved several moments earlier after selecting "Exit Game" from within the game's pause screen. this test will also be performed after the game has been shut off and then turned back on.

#### TEST 2

**Test Objective:** Test whether the "settings" button functions appropriately and that its contents function appropriately - i.e., the option to Control levels of sounds/music.

**Test Description:**  The "settings" button will be clicked by the tester and they will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the settings screen appears and contains the proper buttons, allowing the player to edit game settings, such as sound level and music level and ensure they function as desired.

**Test conditions:** see test environment

**Expected Results:** This test will show the contents of the settings screen, the button for the option for controlling levels of the music and sound effects.

#### Test 3

**Test objective:** Test that the “Exit Game” button on the main menu exits the game properly

**Test Description:** The tester will click the “Exit Game” button on the main menu and the game will exit without any errors

**Test conditions:** see test environment

**Expected Results:** Command will return once more to the operating system and the game will have exited with no errors.

### Gameplay Unit Tests:

**Participants:** Team 3

**TEST 1**

**Test Objective:** Test that all game animations and sound effects occur at the correct times

**Test Description:** The tester will play through the game and ensure that the player/enemy animations occur at appropriate times and character animations don't appear jerky; instead they are smooth animations

**Test Conditions:** see test environment

**Expected Results:** The enemies and player have animations that are smooth while they walk along the path, additionally, all sound effects occur at logical times, i.e., an attack sound is only heard when a player-character's attack animation is seen and it is only seen when an enemy is attacked.

**TEST 2**

**Test Objective:** Test that all control mechanisms work appropriately

**Test Description:** The tester will play through the game and ensure that the player moves according to the controls move forward, backward, jump, crouch, attack and pause/resume game.

**Test Conditions:** see test environment

**Expected Results:** The player moves forward with Right arrow key/D, backward with Left arrow key/A, jumps with Up arrow key/W, crouch with c and attack with Left mouse click/R and the game is paused or resume with spacebar.

**TEST 3**

**Test Objective:** Test that lives work appropriately

**Test Description:** The tester will play through the game and ensure that the player loses lives upon collision with enemy projectile. and gains lives when collides with health pickup.

**Test Conditions:** see test environment

**Expected Results:** The players lives increase by 1 diamond upon health pickup and decreases by 1 diamond upon collision with enemy projectile.

## System and Integration Testing

**Participants:** Team 4

### Integration Testing Definition:

Integration testing tests interfaces between components of the game, interactions to different parts of the game, file system for saving or loading games and hardware or interfaces between systems.

### Methodology:

For integration testing testes will use top down integration.

Starting with testing the game levels then down to pause menu and the lastly the main menu. Team of developers and testers will write test scripts.

**Test Description:** The tester will play through the game and ensure that the player loses lives upon collision with enemy projectile. and gains lives when collides with health pickup. Then tester will test the pause menu and then will test the main menu of the game.

**Expected Result:** The game works as intended in the game requirements.

### System Testing Definition:

Concerned with the behavior of the whole Game as defined by the scope of a project Test Scope above.

### Methodology:

In this all the functionalities of the game such as Main Menu functions, Pause Menu functions and Gameplay functions will be tested on a whole will see whether this module is working perfectly or not, by including all the control mechanisms. Team of developers and testers will write test scripts.

**Test Description:** The tester will play the game starting from main menu and complete all three levels to ensure the game as a whole works appropriately.

**Expected Result:** The game works as intended in the game requirements.

## Performance and Stress Testing

**Participants:** Team 5

### Performance Testing Definition:

Testing to determine the performance of a game e.g. resource utilization and response times.

### Methodology:

The focus of Performance Testing is to check the game’s

**Speed** - Determines whether the game responds quickly

**Scalability** - Determines maximum user load the game application can handle.

**Stability** - Determines if the game is stable under varying loads

**Test Description:** The Unity Performance Test Extension will be used to test the performance that can be run using the Unity Test Runner. And the Unity Performance Benchmark Reporter will be used to compare the performances.

**Expected Result:** The tester will get report of the performance.

### Stress Testing Definition:

Exercises the game beyond its maximum design load.

### Methodology:

Game will be stressed for a short period of time to know its withstanding capacity. A most prominent use of stress testing is to determine the limit, at which the system or Game or hardware breaks.

**Test Description:** The Unity Profiler is a tool you can use to get performance information about your application. Tester will connect it to devices on the network or devices connected to the machine to test how game runs on the intended release platform. Tester will also run it in the Editor to get an overview of resource allocation of the game. Team of developers and testers will write test scripts.

**Expected Result:** The game works flawless under stress.

## User Acceptance Testing

**Participants:** All Volunteered users.

### Definition:

The purpose of acceptance test is to confirm that the system is ready for operational use. During acceptance test, end-users (customers) of the system compare the system to its initial requirements.

### Methodology:

UAT testing will be conducted smoothly by offering hands-on training of the gameplay to the users invited on site and to those who volunteered and will then use user stories.

**Test Description:** The users will test the functionality validating fitness for purpose by playing the game.

**Expected Result:**  The Game behaves as desired and engages the user.

## Beta Testing

**Participants:** All volunteered users.

### Methodology:

Test conducted by cross-section of the users (live application of the s/w), who will install it, and use it under real-world conditions. The users will send records of incidents with the Game to the development organization where the defects will be repaired.

# TEST SCHEDULE

* Unit Testing on Main Menu Functions assigned to (Team 1) to be completed by 22/05/2020.
* Unit Testing on Pause Menu Functions assigned to (Team 2) to be completed by 21/05/2020.
* Unit Testing on Gameplay Functions assigned to (Team 3) to be completed by 24/05/2020.
* Integration Testing on Main Menu and Gameplay assigned to (Team 1 and Team 3) to be completed by 26/05/2020.
* Integration Testing on Pause Menu and Gameplay assigned to (Team 2 and Team 4) to be completed by 26/05/2020.
* Integration Testing on Main Menu, Pause Menu and Gameplay assigned to (Team 1, Team 2 and Team 3) to be completed by 28/05/2020.
* Performance and Stress Testing assigned to (Team 1) to be completed by 29/05/2020.
* User Acceptance Testing under supervision of (Team 2) to be completed by 10/06/2020.
* Beta Testing to be completed by 10/07/2020.

# CONTROL PROCEDURES

### Problem Reporting:

The following Procedure will be followed for problem reporting:

1. Summary line (short, summation of the problem)
2. The Actual Result (a description of what happened; the problem)
3. The Expected Result (what you thought should happen instead of the problem)
4. Steps to Replicate (how someone else can make the problem happen too)
5. Severity (how bad is the problem).

### Change Requests:

The following form will be used for Change Request and will be signed by the development team manager.

The CR form would normally contain the following entries:

1. The CR Id, which could be a serial number.
2. The CR description.
3. Date on which the CR is received.
4. Allocation details for Analysis including date of allocation, completion date, to whom it is allocated.
5. Allocation details for Approval of CR including date of allocation, completion date, to whom it is allocated.
6. Allocation details for Resolution of CR including date of allocation, completion date, to whom it is allocated.
7. Allocation details for peer review including date of allocation, completion date, to whom it is allocated.
8. Allocation details for regression testing including date of allocation, completion date, to whom it is allocated.
9. Status – open, closed or under analysis / approval / resolution / peer review / regression testing.
10. Date on which CR is closed.

# FEATURES TO BE TESTED