Test Plan

Software Testing

Joe Hallinan

G00267801

Table of Contents

1.0 Introduction………………………………………………………………………………………………………………………………2

2.0 Objectives & Tasks…………………………………………………………………………………………………………………… 3

2.1 Objectives…………………………………………………………………………………………………………………………… 3

2.2 Tasks…………………………………………………………………………………………………………………………………….4

3.0 Scope……………………………………………………………………………………………………………………………………….4

3.1 General…………………………………………………………………………………………………………………………………4

3.2 Tactics…………………………………………………………………………………………………………………………………..5

4.0 Testing Strategy……………………………………………………………………………………………………………………….5

4.1 Unit Testing………………………………………………………………………………………………………………………….5

4.2 System and Integration Testing……………………………………………………………………………………………5

4.3 Performance and Stress Testing…………………………………………………………………………………………..6

4.4 Acceptance Testing………………………………………………………………………………………………………………6

4.5………………………………………………………………………………………………………………………………………………

4.6……………………………………………………………………………………………………………………………………………….

4.7……………………………………………………………………………………………………………………………………………….

**1.0 Introduction**

Test plan documents the strategy that will be used to verify and ensure that the product meets its design specifications and other requirements. This document contains guidelines and direction that will assist designated staff and personnel involved in testing in completing their task.

The product that will be tested is a 2D left to right side scrolling platform game, like Nintendo’s Super Mario Brothers and Sega’s Bomber games.

The levels and characters will be made primarily using Aseprite which allows you to create 2D sprites and animations. The main character that the player will control will be a wizard who you guide from one level to the next while using your magic potions to defeat the enemy. As the wizard makes his way through the level, he can collect various pickups, such as health, which can be used to boost his health back to 100%. At the end of each level the wizard will need to defeat an enemy or a boss to progress.

The game will feature game statistics, such as highest score or best time, enemies will increase in difficultly as you progress through the levels and have various other obstacles that the player must defeat.

**2.0 Objectives and Tasks**

The objectives for this project are to test every criteria for the game. This will enable us to achieve our goals that we are setting out at the objectives. The purpose of this is to find any possible software defects before the game is released to the public.

**2.1 Objectives**

**Objectives to Test:**

* **Main Menu:** Play Game, Settings, Load Game, Delete Game, Exit Game.

Play Game will bring the player to the first level.

Settings will allow the player to change some settings within the game such as music.

Load Game will allow the player to load the last level the player saved, or any previous levels saved.

Delete Game will allow the player to delete the game.

Exit Game will allow the player to exit the game.

* **Pause Menu:** Saved Game, Settings, Exit.

Once the player pauses the game, they will be presented with various options. These will include saving the current level that they are playing.

Settings will allow the player to change some options within the game, such as turning the music on/off, muting the sound.

Exit will allow the player to exit the game and return to the main menu.

* **Control Mechanisms:** Allows the player to move in any direction within the game.

If playing on a mobile device the player will be able to control the wizard with directional arrows.

If playing on a PC, the player will be able to use certain keys on the keyboard to control the wizard.

* **The Game:**

When you begin the game, you will be presented with three options, to play the game, access the settings and finally to exit the game. When the player starts the game, level one will load, and you take control of the wizard. You then guide the wizard through the many obstacles and enemies and try to defeat the boss at the end to progress to level two. The game has three levels, each getting more difficult as you progress. When the player has completed the game, they will be given the option to exit the game or to restart the game.

**2.2 Tasks**

1. Access the game on mobile device or PC.
2. Start level one
3. Check that all player control buttons are working, wizard can move in any direction.
4. Press pause and check settings.
5. Mute/Unmute sound within the game.
6. Unpause the game and you should be brought back to current level.
7. Let the wizard be attacked by the enemy to see if health level goes down.
8. Attack enemy to see if you can defeat them.
9. Press pause and save the current level.
10. Exit the game.
11. Access the game.
12. From main menu load the level previously saved.
13. From the saved level exit the game.
14. From the main menu delete the level saved.
15. Access the game.
16. From the main menu load the game.
17. Has the previously saved level been deleted?
18. Access the game.
19. Play levels one to three.
20. After level three are you given the option to restart the game or exit the game.

**3.0 Scope**

**3.1 General:**

The purpose of this test plan is to achieve our targets in the objectives and tasks sections. These will be achieved over a period yet to be determined. The purpose of these objectives and tasks is to ensure that when the game is released to the general public it will be have no errors or glitches within the game and will be of the highest quality possible.

**3.2 Tactics:**

As stated above in the general section, we aim to achieve our targets over a period yet to be determined. All features of the game will be tested robustly and in a structured approach before being released to the public.

**4.0 Testing Strategy**

The purpose of this test strategy is to provide an outline of the testing approach that we are going to undertake. This will also ensure that we follow a structure of what testing needs to be done in the various steps and any defects that we find in the testing phase will be solved before we release the product.

**4.1 Unit Testing**

**Definition:**

A unit test is a way of testing a unit, which is the smallest piece of code, for example, a function, method or a property. Unit testing involves breaking your program into smaller pieces and subjecting each piece to a series of tests. This should mean that if any set of input is fed to function or procedure it should result in an expected output. Defects are found at a very early stage by the developer, stress is being placed on making the code inter dependent, which in turn reduces the chances of it affecting other sets of code.

**Participants:**

In unit testing this role is performed by the developer/the development team. This is to ensure that when a developer makes changes to the source code, new changes will still pass the existing tests.

**Methodology:**

Unit testing is the first level of testing and is performed by the developers. Developers in a test-driven environment will write and run the tests before the software or the features being tested are passed along to the test team. Unit testing can be performed manually but automating the tests will speed up the delivery cycles.

**4.2 System and Integration Testing**

**Definition:**

System Integration Testing is performed to establish the interactions between the modules of a software system. It is used with the validation of the low- and high-level software requirements that are described in the software requirements specification and the software design document. The aim of system and integration testing is to confirm that all the software modules are functioning correctly and that the integrity is maintained between the separate modules.

**Participants:**

System and Integration Testing is a role that is mostly performed by testers but can sometimes also be performed by developers who would write integration tests.

**Methodology:**

System Integration Testing is the procedure of comprehensive testing that is performed on the application software along with the overall system. The main aim of performing this way of testing is to ensure all the functional characteristics of the software and the hardware systems are in sync with each other.

**4.3 Performance and Stress Testing**

**Definition:**

Stress testing is a testing technique that is performed as part of performance testing. During the stress test the system is monitored after exposing the system to overloading to ensure that the system can maintain the stress. Stress testing tries to break the system under test by overwhelming its resources. The purpose of this is to verify that the system fails and can recover.

**Participants:**

Performance and Stress Testing is mainly performed by testers who try and overwhelm the software during stress tests to check if it breaks and can recover.

**Methodology:**

Performance and Stress Testing will be performed by our testers. The purpose of this is to overload the system and see how it performs when it gets to breaking point. The aim here is that the game is pushed to its maximum limit and see how it responds to a full recovery.

**4.4 User Acceptance Testing**

**Definition:**

An acceptance test is a formal description of the behaviour of a software product, generally expressed as an example. Once the system and integration testing is complete by the testing time and is passed onto the next step, the product is handed over to the customer to test for its acceptability, in other words, the product should meet both its critical and major business requirements. Even though system and integration testing has successfully completed, the acceptance test is required by the customer to gain confidence in the product, to ensure that the product is working as it should be and to ensure that the product matches the market standards.

**Participants:**

Acceptance testing is performed by either internal or external teams. Internal acceptance testing is usually performed by members of the product management, sales and/or customer service. External acceptance testing is performed by the customer or the end user.

**Methodology:**