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…………………………………………………………………………………………………………………………………….3

3.0 Scope……………………………………………………………………………………………………………………………………….3

3.1 General…………………………………………………………………………………………………………………………………3

3.2 Tactics…………………………………………………………………………………………………………………………………..3

4.0 Testing Strategy……………………………………………………………………………………………………………………….4

4.1 Unit Testing………………………………………………………………………………………………………………………….4

4.2 System and Integration Testing……………………………………………………………………………………………4

4.3 Performance and Stress Testing…………………………………………………………………………………………..5

4.4 User Acceptance Testing………..……………………………………………………………………………………………5

5.0 Test Schedule………………………………………………………………………………………………………………………….6

6.0 Quality Control Procedures…….………………………………………………………………………………………………6

6.1 Document Defects…………..………………………………………………………………………………………………….6

6.2 Document Changes…………………………………………………………………………………………………………….6

7.0 Features to be Tested…………………………………………………………………………………………………………….6

8.0 Features not to be Tested……………………………………………………………………………………………………….

**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**

The Objective of this test is to verify that the functionality of the 2D game “The Pixel Wizard Version 1.0” works according to the specifications set out in this test plan.

The test will execute and verify the test scripts, identify, fix and retest all defects that are encountered during the testing cycle.

The objective is to have a product that is 100% defect free and ready for the client.

**2.2 Tasks**

* Unit Testing
* System and Integration Testing
* Performance and Stress Testing
* User Acceptance Testing

**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**

**Team Lead: John Ward**

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:**

Mike Johnson & Emer Reilly.

**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:**

Dan Brown & Jimmy Daniels.

**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:**

Darren Nestor, Fiona Joyce & Niamh Doherty.

**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 are 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:**

Ray Murphy, Sara Fitzgerald & Orla Ryan.

**Methodology:**

Our beta testers will test every aspect of the game, from the time it takes to load the game, functionality, response time, right through to any defects/bugs encountered playing the game and will report back to John Ward on their experience. This step can only be completed once all other steps have passed their objectives, as once we have completed the user acceptance testing the game is nearing completion to be released to the client.

**5.0 Test Schedule**

* Include any test milestones as per the software project schedule.
* Identify the schedule for each test and task milestone.
* List any additional test milestones that are required, include additional time if necessary.
* Testing and Bug Reporting.

**6.0 Quality Control Procedures**

The aim of quality control testing is to verify that the product meets performance and usability standards. These standards are set during the software development process, but quality control testing can tweak these standards if needed.

**6.1 Document Defects**

Document any defects that may arise during the testing process. If using an automated logging system, state name of application that was used and steps taken during this process.

**6.2 Document Changes**

Document any changes/modifications that have been applied to the product. State who has oratorized these changes/modifications and the criteria for applying them.

**7.0 Features to be Tested**

* **Check for background music and sound effects:** on/off sound and background music,

verify if sound effects are in sync with action, on/off device sound, check for vibration effect if present.

* **User Interface:** Font displayed, check for screen title, character should not move out of the screen/specified area, check scrolling, check the player controls, check animation of character, check in landscape/portrait mode.