**Software Testing**

**Test Planning Project**

**Test Plan Template:**

**Prepared by** Steven Joyce

**Date:** 03/05/2020

**TABLE OF CONTENTS**

1.0 INTRODUCTION

2.0 OBJECTIVES AND TASKS

2.1 Objectives

2.2 Tasks

3.0 SCOPE

4.0 Testing Strategy

4.1 Unit Testing

4.2 System and Integration Testing

4.3 Performance and Stress Testing

4.4 User Acceptance Testing

4.5 Batch Testing

4.6 Automated Regression Testing

4.7 Beta Testing

5.0 Test Schedule

6.0 Control Procedures

7.0 Features to Be Tested

8.0 Features Not to Be Tested

9.0 Resources/Roles & Responsibilities

10.0 Schedules

11.0 Risks/Assumptions

12.0 Tools

**1.0 INTRODUCTION**

The product being tested is The Pixel Wizard. It is a 2D side scrolling platformer. The gameplay will have at least 3 levels of ever-increasing difficulty. The player will have the option to save the game to store the progress they have already acquired. The saved game can then be loaded in the main menu. The player will have a health mechanism that will be decreased by taking damage from the enemy, the player can also gain health from health pick up items. Enemies also have a health mechanism that will vary with all the different enemy types and different levels.

**2.0 OBJECTIVES AND TASKS**

**2.1 Objectives**

The main objective of the project is to create a 2d side-scrolling platformer type game. The game is a single player application.

The project’s main objectives

* The game is able to save the user progress
* The user can load up that saved game in the main menu
* The game will have multiple levels – at least 3 different levels
* The user and enemy will have a set amount of damage it can take before being killed
* The game will have customizable features

**2.2 Tasks**

All aspects of the game are tested and functions successfully

* The game loads up when the Play Game button is clicked
* The player can cause damage and take damage
* The player can be killed when health icons run out
* The enemy can be killed when health is empty
* Health pickups give the player more life
* The advancement of the game between levels works
* The player can pause the game
* All he options in the pause screen allow user to access the correct screen
* The player can quit the game and return to the main menu
* The player can save the progress achieved
* Settings menu can be accessed using the main menu
* All Settings can be changed correctly by the player
* Load Game Button can be clicked, and user can load up a saved game of their choosing
* The Delete Game Button allows user to choose which game file to delete when clicked
* When user clicks Exit Game, the application is closed
* The approval of test cases will be communicated via work email.

**3.0 SCOPE**

**General**

The purpose of this test is to ensure basic deformities are removed from the game before the

Next step in the project is undertaken.

**Tactics**

To accomplish this the test will need:

* To make sure all menu options work correctly
* That the game runs smoothly
* That the player and enemy both take damage and can be killed
* The game can be saved
* The game can be paused and bring up the pause menu
* The options all work to allow the player to customize the game
* The game will exit when Exit game is chosen

**4.0 TESTING STRATEGY**

**4.1 Unit Testing**

**Definition:** Unit testing will be carried out using the white box testing technique. This means the tester will have in depth knowledge of the project.

**Participants:** Mark Wallace

**Methodology:** The developer will systematically test each function of the project. All aspects of the project that is set out by the client as a system requirement will be tested individually to ensure that they function correctly. The tests will be written by Mary Burns.

|  |  |
| --- | --- |
| **Unit** | **Component** |
| Unit 1 | Front End |
| Unit 2 | In Game Menus |
| Unit 3 | Control Mechanisms |

**4.2 System and Integration Testing**

**Definition:** With the individual components of the system working correctly, they will now be tested when integrated with the other components.

**Participants:** Jenny Hill

**Methodology:** The parts will be tested with 2 components linked at a time until all components are able to work when integrated together. The tests will be written by Tim Jennings.

|  |  |
| --- | --- |
| **Test** | **Components** |
| Test 1 | Front End &Control Mechanisms |
| Test 2 | Front End & In Game Menus |
| Test 3 | In Game Menus & Control Mechanisms |
| Test 4 | Front End & Control Mechanisms & In Game Menus |

**4.3 Performance and Stress Testing**

**Definition:** The tester will check if the game crashes or freezes when the performance on the game is pushed to its limits. Will the number of enemies on screen have a limit to where if reached the game crashes? Does the player attacking constantly cause issues? What causes failures to the program, can the game crash?

**Participants:** Arron Thompkins

**Methodology:** The tester will run the game several times testing out several different scenarios. Each scenario will be unique. The tests will be written by Mary Burns.

|  |  |
| --- | --- |
| **Test** | **Scenario** |
| Test 1 | The number of enemies will reach its max number. |
| Test 2 | The player will kill all enemies apart from the boss |
| Test 3 | The player will constantly perform attacking actions |
| Test 4 | The player will pick up health items even when health is full |
| Test 5 | The game will run with no end |
| Test 6 | The player will stay on the edge of the screen |
| Test 7 | The game will stay paused for 2 minutes |

Test 1 – It will test how many enemies can appear until it crashes

Test 2 – Tests how many enemies can be killed until it crashes due to the level not being finished

Test 3 – It will test if constant attacks cause any issues

Test 4 – It tests how many pick up items can be picked up even when health is full

Test 5 – It will test how long can the game run until any errors occur

Test 6 – It tests does the user being on the edge of the screen make any difference to the application

Test 7 – Does the pause menu being active for a long period of time cause problems?

**4.4 User Acceptance Testing**

**Definition:** The client will test out the application to see if what has been created meets the criteria set out at the beginning of the project. Has all the client’s needs been met?

**Participants:** Harry Burke

**Methodology:** The client will test the game fully. The game shown to the client should be a near final version, the only changes left are ones that are found in this round of testing. The client will test all the features of the application. The tests will be written by Tim Jennings.

**4.5 Batch Testing**

**Definition:**

**Participants:** Arron Thompkins

**Methodology:** The tests will be written by Mary Burns.

**4.6 Automated Regression Testing**

**Definition:**

**Participants:** Mark Wallace

**Methodology:** The tests will be written by Tim Jennings.

**4.7 Beta Testing Participants:** A select group of computing software students from across the country, no more than 40 per college. The number of participants will be 1000.

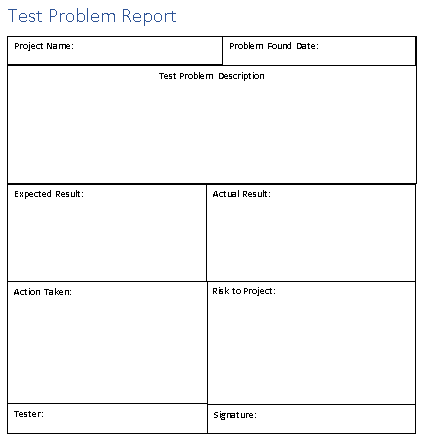
**Methodology:** The students will get the game via a link to a downloadable game to run of their PC’s or laptops. The beta will last 3 days. At the end of the beta the participants will get an online questionnaire to fill out to give the developers feedback on the game. All participants will get 25% of the game when it launches for participating in the project. The tests will be written by Mary Burns.

**5.0 TEST SCHEDULE**

**6.0 CONTROL PROCEDURES**

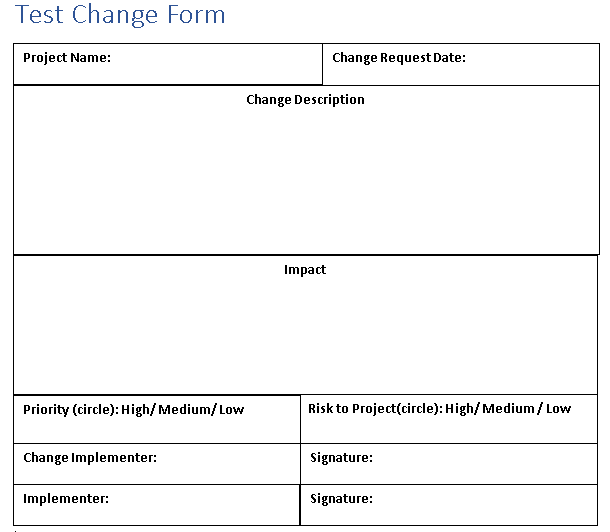
**Problem Reporting**

Form to send a test problem report to Test manager

****

**Change Requests**

Form to request changes to Test manager



**7.0 FEATURES TO BE TESTED**

* All menu options
* The player can take damage and be killed
* The enemy can take damage and be killed
* The health pickup items add health to the player
* The game can be paused
* The game controls function correctly

**8.0 FEATURES NOT TO BE TESTED**

**9.0 RESOURCES/ROLES & RESPONSIBILITIES**

|  |  |  |
| --- | --- | --- |
| Roles | Staff Member | Responsibilities |
| Test Manager | John Smith | * Provide managerial oversight of the testing * Apply a clear direction * Create a good team camaraderie * Build a team of professionals that are assigned to the correct roles of their skill set. |
| Test Designer | Tim Jennings  Mary Burns | * Design the tests for the application * Send all test cases to the test manager for review * Assess and verify the Test approach * Identify the appropriate techniques and supporting tools |
| System Tester | Arron Thompkins  Jenny Hill  .Mark Wallace | * Will run the tests * Will give the developers feedback on the game * Will be unbiased in the implementation of the tests |
| Test System Administrator | Jennifer Swan | * Give the team support in the development and running of the tests * Makes sure that the environment to run the tests is managed and maintained |
| Project Manager | Harry Burke | * The person who gave the test to this team * He will be testing the product towards the end of the testing process using the instructions provided by this test team |

**10.0 SCHEDULES**

* **Test Plan –** The final version of the test plan needs to be submitted by May 14th.
* **Test Cases Plan –** The final version of the test cases needs to be submitted by May 12th.
* **Test Incident Reports Plan –** The final version of the incident reports plan needs to be submitted by May 9th.
* **Test Summary Reports Plan –** The final version of the summary reports plan needs to be submitted by May 10th.

**11.0 RISKS/ASSUMPTIONS**

**Project Risks**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Probability** | **Impact** | **Mitigation Plan** |
| Change in requirements | High | High | * Weekly meetings with client for the developer team * After each stage of development,   the client sees the progress |
| Software used to develop tests during build | Low | High | * Make sure no updates are set for automatic updates * The software used to test is the latest stable version and the testers have a good understanding of the software |
| Natural Disaster/Crisis | Low | High | * Set up a Microsoft team * Set up an online code repository to continuously update project as a team using software like GitHub |
| Testing delayed due to problems/Issues | Low | High | * Make sure all tests are correct * All tests test the correct components and all features that are important |
| Testing Schedule is tight | High | High | * Testing team |

**12.0 TOOLS**