The Pixel Wizard – Test Plan

**Test Plan Template:**

The Pixel Wizard

**Prepared by:**

Connor Brookfield

12/05/2020

**TABLE OF CONTENTS**

1. INTRODUCTION
2. OBJECTIVES AND TASKS
   1. Objectives
   2. Tasks
3. SCOPE
4. Testing Strategy
   1. Unit Testing
   2. System and Integration Testing
   3. Performance and Stress Testing
   4. User Acceptance Testing
   5. Batch Testing
   6. Automated Regression Testing
   7. Beta Testing
5. Test Schedule
6. Control Procedures
7. Features to Be Tested
8. Features Not to Be Tested
9. Resources/Roles & Responsibilities
10. Schedules
11. Risks/Assumptions
12. Tools
13. **INTRODUCTION**

The Pixel Wizard is a platformer game in which the player-controlled character must jump and climb between suspended platforms while avoiding obstacles and killing enemies and bosses to gain.

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.

**2.0 OBJECTIVES AND TASKS**

**2.1 Objectives**

The objective of this test play is to make sure that all areas fit the design specifications. Including:

* Menu Items
* Options menu
* Jumping mechanics
* Movement mechanics
* Damage mechanics
* Shooting mechanics
* Progression mechanics
* Health mechanics
* Enemy mechanics

**2.2 Tasks**

To using multiple testing strategies on the individual functions listed in the objectives, prioritising the player mechanics before prioritising enemy mechanics, then damage, shooting mechanics, health, progression, menu items, options menu and exit game.

**3.0 SCOPE**

**General:**

Testing the functions included in the objectives, tasks, and staying within the design specifications.

**Tactics**:

Prioritise the objectives as outlined in the tasks, in order.

**4.0 TESTING STRATEGY**

To ensure that all defined functionality in the design specifications work, using multiple testing methodologies, as defined below.

**4.1 Unit Testing**

**Definition:**

What is going to be included in the unit test.

Specify the minimum degree of comprehensiveness desired. Identify the techniques which will be used to judge the comprehensiveness of the testing effort (for example, determining which statements have been executed at least once). Specify any additional completion criteria (for example, error frequency). The techniques to be used to trace requirements should be specified.

**Participants:**

Jason Stateham, Senior Unit Tester

John Claud, Junior Unit Tester

**Methodology:**

If a test fails, the tester should stop the test, note it, escillate to proj man.

Describe how unit testing will be conducted, including a description of tests to be carried out. The Senior Tester will write the test scripts for the unit testing, what would be the sequence of events of Unit Testing and how will the testing activity take place onsite - unit testing department?

**4.2 System and Integration Testing**

**Definition:**

Does it work on dif platformz – where might there be potential concern/variation.

List what is your understanding of System and Integration Testing for your project.

**Participants:**

Niall Rogers – Senior Systems and Integration Tester

Sean Paul – Junior Systems and Integration Tester

**Methodology:**

Describe how System & Integration testing will be conducted, including a description of tests to be carried out who will write the test scripts for the unit testing, what would be sequence of events of System & Integration Testing, and how will the testing activity take place?

**4.3 Performance and Stress Testing**

**Definition:**

List what is your understanding of Stress Testing for your project.

**Participants:** Who will be conducting Stress Testing on your project? List the individuals that will be responsible for this activity.

**Methodology:**

Describe how Performance & Stress testing will be conducted, including a description of tests to be carried outWho will write the test scripts for the testing, what would be sequence of events of Performance & Stress Testing, and how will the testing activity take place?

**4.4 User Acceptance Testing**

**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. Users, timeframe, how their feedback is collated andanaly.

**Participants:** (Closed test with a selected group of people who fit the target market)

Who will be responsible for User Acceptance Testing? List the individuals' names and responsibility.

**Methodology:**

Describe how the User Acceptance testing will be conducted, including a description of tests to be carried out who will write the test scripts for the testing, what would be sequence of events of User Acceptance Testing, and how will the testing activity take place?

**4.5** **Batch Testing**

**4.6** **Automated** **Regression Testing**

**Definition:**

“Fixed something here, it broke something else in the process”

Regression testing is the selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still works as specified in the requirements.

**Participants:**

**Methodology:**

**4.7 Beta Testing**

**Participants:**

**Methodology:**

**5.0 TEST SCHEDULE**

Finishing each testing phase/type in order.

Include test milestones identified in the Software Project Schedule as well as all item transmittal events. Define any additional test milestones needed. Estimate the time required to do each testing task. Specify the schedule for each testing task and test milestone. For each testing resource (that is, facilities, tools, and staff), specify its periods of use.

**6.0 CONTROL PROCEDURES**

**Problem Reporting:**

Escalation process: Tester -> manger -> dev team -> testers. With feedback/surveys at each stage. Document the procedures to follow when an incident is encountered during the testing process. If a standard form is going to be used, attach a blank copy as an "Appendix" to the Test Plan. In the event you are using an automated incident logging system, write those procedures in this section.

**Change Requests:**

Any testers can raise a change request which may include changing features, functionality, structure etc. Manager signs off. Discuss changes, asses benefits, change and cost etc. Document the process of modifications to the software. Identify who will sign off on the changes (proj manager) and what would be the criteria for including the changes to the current product. If the changes will affect existing programs, these modules need to be identified.

**7.0 FEATURES TO BE TESTED**

Identify all software features and combinations of software features that will be tested.

**8.0 FEATURES NOT TO BE TESTED**

What has already been tested (that have passed). **READ THROUGH GAME FEATURES AND DECIDE WHAT SOUNDS LIKE A PAIN IN THE ASS.**

Identify all features and significant combinations of features which will not be tested and the reasons.

**9.0 RESOURCES/ROLES & RESPONSIBILITIES**

Specify the staff members who are involved in the test project and what their roles are going to be (for example, Mary Brown (User) compile Test Cases for Acceptance Testing). Identify groups responsible for managing, designing, preparing, executing, and resolving the test activities as well as related issues. Also identify groups responsible for providing thetest environment. These groups may include developers, testers, operations staff, testing services, etc.

**10.0 SCHEDULES**

Identify the deliverable documents. You can list the following documents:

-Test Plan

-Test Cases

-Project risks and assumptions

-Test Incident Reports

-Test Summary Reports

**11.0 RISKS/ASSUMPTIONS**

Identify the high-risk assumptions of the test plan. (Not meeting deadlines, unit testing does not identify key issues, making sure all issues have been addressed, testing is limited to the scope of the testing strategy, user acceptance on specific platforms, staying in scope) Specify contingency plans for each (for example, delay in delivery of test items might require increased night shift scheduling to meet the delivery date).

1. **TOOLS**

Selenium. JUnitList the Automation tools you are going to use. List also the Bug tracking tool here.