**The Pixel Wizard Test Plan**

**Prepared by:**

Sagheer Ahmad

09/05/2020

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 Alpha Testing (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 Hardware Requirements

6.0 Environment Requirements

6.1 Main Frame

6.2 Workstation

7.0 Test Schedule

8.0 Control Procedures

9.0 Features to Be Tested

10.0 Features Not to Be Tested

11.0 Resources/Roles & Responsibilities

12.0 Schedules

13.0 Significantly Impacted Departments (SIDs)

14.0 Dependencies

15.0 Risks/Assumptions

16.0 Tools

1. **INTRODUCTION**

**1.1 Project Purpose**

The purpose of this test plan is to showcase the testing approach and overall framework of The Pixel Wizard 2D game developed by Game Development International Ltd. This test document will showcase the test plan developed by the Testing Team.

The following test plan will describe:

* Test Strategy: This will showcase the rules for setting up a test and how they will be set up. The following functions are some of the things to be tested:
  + The front of the game.
  + The menus, both in game and starting menu.
  + The control mechanism.
  + The overall game itself
* Execution Strategy: This will showcase how the tests will be run and how to find defect that can be fixed.
* Management: This will showcase how to handle all the management required to set up a test plan e.g. communication, setting up teams and the risks included.

**1.2 Project Overview**

The Pixel Wizard is a 2D game that is developed by Game Development International Ltd. It is a 2D side scrolling platformer. The game has been inspired by such games as “Fancy Pants”, “Shovel Knight” and “Skyrim”. The game will mainly use pixel art to create the characters and the world. The aim of the game is to beat the enemies and the boss in the level to move onto the next level. The Wizard The levels will get more and more difficult as you progress through the game. The player will also be given powerups in the game like health and speed.

The game uses simple control mechanisms to move the player. The control mechanism will be the same as most games so that the players do not need to learn new controls. The game also includes menus that guide the player through the game.

**1.3 Roles**

The project teams will be responsible for carrying out the tests that are outlined in this document and track their progress.

The project manager will be responsible for looking at the progress being made by the project teams and reviewing the documents. They are also responsible for making sure the tests are being carried out according to the test plan and everything is being done according to the schedule.

The stakeholders can be involved in the test plan to make sure the test plan is accordance to what they wanted.

1. **OBJECTIVES AND TASKS**

# 2.1 Objectives

This test plan for The Pixel Wizard will support the following objectives:

* Define the activities required to prepare and conduct the System, Beta and User Acceptance testing.
* Communicate to all parties what the testing strategy is.
* Define the deliverables and parties/teams.
* Communicate to all parties the various jobs and tasks to be completed.
* Describe how tests are to be setup and how they will be documented.

**2.2 Tasks**

The following tasks are to be completed:

* Determine the scope and the risks that need to be tested and that are not to be tested.
* Identify the features that will and will not be tested.
* Identify the different methods and approaches to testing the game.
  + Integration Testing
  + Unit Testing
  + Regression Testing
  + Beta Testing
  + User Acceptance Testing
* Identify the different teams that will be working on the project and what testing they will be covering.
* Create a test schedule for the test plan.
* Create a document to reports problems with the game.
* Create a document for measuring the success of the testing.
* Create a document for writing the change requests.
* Automate the tests.
* Decide both the Entry and Exit Criteria.
* Evaluate the test estimate.
* Define the management process.

**3.0 SCOPE**

# General

The product being tested is The Pixel Wizard game created by Game Development International Ltd. The functions that will be tested is the Main Menu, Settings, Load Game, Delete Game, Exit Game, and the game itself. The main menu will be tested by seeing if each of the independent buttons do what they are intended to do.

The play button should take the user to the actual game where they are able to play level 1. In the settings menu the player should be able to adjust the game to match their preferences like edit the volume of the game and the music. The user should also be able to load the previous game that you saved. The delete game button should allow the user to delete the previously saved version of the game. The exit button should bring the user out of the game.

Another function that will be tested is the in-game menu. This will test if the user is able to pause and resume the game. The pause menu should allow the player to also access the settings option, save the game and leave the game to go back to the main menu.

The final feature that will be tested is the end screens that will appear when the player dies or when they win the game. There should be a separate death screen for both options.

**In Scope**

* Main Menu
* Controls
* Settings Menu
* Load Game
* Save Game
* Delete Game
* In Game Pause Menu
* Exit Game
* Win and Death Screen

**Out of Scope**

* Load Game
* Save Game
* Delete Game

The Load, Save and Delete game can be both in and out of scope because according to the document these options are optional. So, we do not know if these are going to be part of the game.

# Tactics

The way we plan to test the game is to use the main four stages of testing that are required for testing a product before release. These four stages are:

* Unit Testing
* Integration Testing
* System Testing
* Acceptance Testing

We intend to use these testing approaches due to the game still being in the development stage. The testing team is going to be split up into different groups who will be testing each of the main functions e.g. Team one will be testing the main menu and Team two will test the game and the in game menus. The teams will also be required to document their progress and report the problems to the project manager.

For the back end of the game we can use Unit test to test the game and can be done in accordance with the developer. The front end will have a black box approach to it. If the game is shown correctly online and, in the reports, we can assume that the game works properly for the user. The front-end testing can also be automated to reduce the test times.

The dev team will provide defect fix plans based on the defect meetings during each cycle to plan. The same will be informed to the test team prior to start of defect fix cycles.

The project manager will review and sign off the tests prior to the test execution. This is done to save time that might be wasted on random tests.

# 4.0 TESTING STRATEGY

We will be using agile methodology to test The Pixel Wizard game. Agile Methodology is:

* Iterative approach to software development and testing
* Highly collaborative
* Continuous design improvement
* Guiding standards
  + Shared responsibility
  + Test automation
  + Test management
  + Data management
  + Accurate
  + High maintainability
  + Inter-operability

**Performance Testing**

Performance testing will be one of the main ways of testing the game because it covers a wide range of functional evaluations even when a detailed description is not given. It also tests most of the main features of the game. It also verifies and validates the performance requirements have been achieved.

# 4.1 Unit Testing

**Definition:**

Unit Testing is a level of software testing where individual components of a software is tested. The purpose of this type of testing is to validate each unit of the game is working properly and does what it is intended to do. A unit is the smallest component of any software. It usually has very few inputs and usually a single output. Because of this the testing can be more accurate. Unit testing frameworks, drivers, stubs, and mock objects are used to assist in unit testing.

**Participants:**

Team 1 will be responsible for all the unit testing. They can work along with the developers to test the back end of the game. Team 1 is a team full of testers but also have knowledge of developing.

**Methodology:**

Unit testing is performed by using the white box test method. It is normally performed by software testers or their peers but in this case, we are doing independent testing. We will be getting the developers to help us with the testing. The test scripts will be written by team 1.

The test team will prepare, review, rework and baseline the unit test plan and the unit test case. To carry out the unit testing the team must:

* Find the proper tool/framework for the coding language
* Isolate the development environment from the test environment
* Write test cases that independent to any other test case
* All paths must be covered especially loop conditions
* Make sure to use a version control system to keep track of the test scripts
* The test cases must be done continuously and frequently

# 4.2 System and Integration Testing

**Definition:**

Integration Testing is a level of software testing in which software modules are integrated logically and tested as a group. A normal software product has different modules that have been coded by multiple developers. The purpose of integration testing is to report problems in the interactions of these modules when they are integrated together. It focuses on checking data communication amongst these modules.

System testing is a type of testing that validates the complete and fully integrated software product. This type of testing is done to evaluate the system’s compliance with the specified requirements.

**Participants:**

Since this is more of a black box testing, team 2 will be responsible for conducting system and integration testing. Team 2 will be a team of testers who are good at testing the final product.

**Methodology:**

System and Integration testing will be done using the black box testing method. This testing is done after the unit testing and is usually done by independent testers. In this case team 2 will be doing the testing. Integration testing can be done by the developers or by team 2.

Integration Testing will be done by:

* Preparing a plan
* Designing different test scenarios, cases and scripts
* Executing the cases and reporting the problems/errors
* Track the defects and re-test until fixed

1. The very first step is to create a **Test** Plan.
2. Create **System Test** Cases and **test** scripts.
3. Prepare the **test** data required for this **testing**.
4. Execute the **system test** cases and script.
5. Report the bugs.

# 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. Who 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.

**Participants:**

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

**Methodology:**

Describe how the User Acceptance testing will be conducted. 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:**

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 HARDWARE REQUIREMENTS

* Computers
* Modems
* Stable Wi-Fi Connection
* Computer accessories – Keyboard and Mouse
* Microphone
* Web cam
* Sufficient desk space

Most of these requirements are important for meetings and documenting everything. The stable Wi-Fi connection is required for fast internet accessibility and testing.

**6.0 ENVIRONMENT REQUIREMENTS**

# 6.1 Main Frame

Specify both the necessary and desired properties of the test environment. The specification should contain the physical characteristics of the facilities, including the hardware, the communications and system software, the mode of usage (for example, stand-alone), and any other software or supplies needed to support the test. Also specify the level of security which must be provided for the test facility, system software, and proprietary components such as software, data, and hardware.

Identify special test tools needed. Identify any other testing needs (for example, publications or office space). Identify the source of all needs which are not currently available to your group.

**6.2 Workstation**

# 7.0 TEST SCHEDULE

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.

**8.0 CONTROL PROCEDURES**

# Problem Reporting

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

Document the process of modifications to the software. Identify who will sign off on the changes 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.

**9.0 FEATURES TO BE TESTED**

The features that will be tested will be:

* Pressing the play button
* Increasing and decreasing the volume for the game
* Opening a previously saved game
* Deleting a previously saved game
* Playing Each level
* Test the attacking
* Test the jumping and movement of the player
* Test the crouching
* In game menu pops up and resumes
* Access the volume setting while playing the game
* Restart the level
* Save the game while playing a level
* Exit the game

# 10.0 FEATURES NOT TO BE TESTED

These are some of the areas that will not be tested:

* Marketing
* Security

These features are optional, so they are both in features to be tested and in features not to be tested. They will remain in both until confirmed otherwise by the developers:

* Opening a previously saved game
* Deleting a previously saved game
* Save the game while playing a level

# 11.0 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 the test environment. These groups may include developers, testers, operations staff, testing services, etc.

**12.0 SCHEDULES**

# Major Deliverables

Identify the deliverable documents. You can list the following documents: - Test Plan

* Test Cases
* Test Incident Reports
* Test Summary Reports

6.2 Deliverables

Deliverable For Date / Milestone

Test Plan Project Manager; QA Director; Test Team

Traceability Matrix Project Manager; QA Director

Test Results Project Manager

Test Status report QA Manager, QA Director

**13.0 SIGNIFICANTLY IMPACTED DEPARTMENTS (SIDs)**

Department/Business Area Bus. Manager Tester(s)

# 14.0 DEPENDENCIES

Identify significant constraints on testing, such as test-item availability, testing-resource availability, and deadlines.

# 15.0 RISKS/ASSUMPTIONS

Identify the high-risk assumptions of the test plan. Specify contingency plans for each (for example, delay in delivery of test items might require increased night shift scheduling to meet the delivery date).

**16.0 TOOLS**

The following tools were used to test the

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