The Pixel Wizard Test Plan

Prepared by:

Sagheer Ahmad
G00357770

09/05/2020

Contents

1.0 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.

2.0 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
- User 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.

Team 1: A team of testers who also have a knowledge in developing and programming.
Team 2: A team of testers whose main concern is to find and report problems in a program.
Team 3: Developers of the game
Team 4: Test Engineers
Team 5: End users

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.

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

Entry Criteria for Unit Testing:

- Technical Design document is finished, approved and released
- System design is documented

Exit Criteria for Unit Testing:

- Units tests pass
- Code is complete
- The tests have been documented
- Redundant code is removed

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.

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

System Testing will be conducted by:

- Creating a test plan
- Creating system cases and scripts
- Preparing the test data that will be used
- Execute the test cases and scripts
- Document and report the bugs

Entry Criteria for Integration Testing:

- Unit Testing must be completed
- Test plan document must be ready and signed off
- Test cases are prepared
- Test environment is set up

Exit Criteria for Integration Testing:

- Integration tests pass
- The tests have been documented
- Report is prepared

4.3 Performance and Stress Testing

Definition:

Performance Testing is a process in which the speed, response time, scalability and reliability of a software program is tested. This is all done under the expected workload. The purpose of performance testing is to performance bottlenecks in the software, in this case The Pixel Wizard.

Stress Testing is a type of performance testing. It involves testing a product under extreme workload to see how it will handle high traffic. Stress testing will push The Pixel Wizard beyond its limits.

Participants:

Since the speed and reliability of the game is being tested, Team 2 will be responsible for performance testing. They will be doing this independent to the developers.

Methodology:

Performance and Stress testing will be carried out by:

- Identify the testing environment
- Design performance tests/scenarios
- Identify the target metrics
- Prepare the environment
- Develop the tests
- Execute the tests
- Analyse the test
- Report and document the problems
- Retest

Entry Criteria for Stress Testing:

- Integration Testing must be completed
- Test plan document must be ready and signed off
- Test cases are prepared
- Test environment is set up
- Test data is prepared for testing

Exit Criteria for Stress Testing:

- tests pass
- The tests have been documented
- Report is prepared

4.4 User Acceptance Testing

Definition:

User Acceptance testing is the last phase of software testing. During this testing actual software users test the application to make sure it can handle the required tasks and the in-world scenarios which are specified. The user can either be a consumer or the client themselves. It is normally performed just before a product is released.

Participants:

The participants involved in this testing will be the client who asked for the testing.

Methodology:

User Acceptance Testing will be conducted by:

- Analysing the business requirements
- Create a test plan
- Identify the test scenarios and the test cases
- Prepare the tests
- Execute the tests
- Record the results in a document
- Compare the results to the business requirements

Entry Criteria for User Acceptance Testing:

- Performance Testing must be completed
- Analyse business requirements
- Test plan document must be ready and signed off
- Test cases are prepared
- Test environment is set up

Exit Criteria for User Acceptance Testing:

- Integration tests pass
- The tests have been documented
- Report is prepared
- Compare report to business requirements

4.5 Batch Testing

Definition:

Batch testing is a group of tests that are executed in a sequential order. Each test batch consists of multiple dependant test cases. In batch testing an end state of one test is base state to another test. If one test case fails or passes, it decided if the rest of the tests will fail or pass.

Participants:

The participants involved in this testing will be Team 4 which is a team of test engineers.

Methodology:

To conduct batch testing, the engineers will create a group of scripts that will be run automatically with an automation tool.

Entry Criteria for Batch Testing:

- User Acceptance Testing must be completed
- Test plan document must be ready and signed off
- Test cases are prepared

Exit Criteria for Batch Testing:

- Batch tests pass – run in sequential order
- The tests have been documented
- Report is prepared

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:

Team 1 and developers will be responsible for regression testing.

Methodology:

Regression testing will be carried out by:

- Understanding what changes have been made
- Determine which parts have been or will be affected
- Determine if you will do a full, partial or a unit regression test
- Schedule the time
- Preparing test cases and test scripts
- Automating these tests
- Execute the tests

Entry Criteria for Regression Testing:

- Determine which parts will be affected
- Test plan document must be ready and signed off
- Test cases are prepared
- Test environment is set up

Exit Criteria for Regression Testing:

- Regression tests pass
- Tests are automated
- The tests have been documented
- Report is prepared

4.7 Beta Testing

Definition:

Beta Testing is a level of testing in which we evaluate the level of customer satisfaction with the application. The end users validate the application and give their feedback. The end users are the people who will use the application.

Participants:

For Beta Testing we will be getting end users to test the Pixel Wizard.

Methodology:

To carry out Beta Testing we will:

- Define the goals
- Recruit participants to take part in the testing
- Distribute the game to the participants
- Share the instructions
- Collect and evaluate the feedback
- If there are no errors and everything works – game is ready for launch

Entry Criteria for Beta Testing:

- Define the goals
- Recruits end users
- Test plan document must be ready and signed off
- Test cases are prepared
- Test environment is set up

Exit Criteria for Beta Testing:

- Regression tests pass
- Feedback is collected
- The tests have been documented
- Report is prepared
- Game is ready for launch

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 TEST SCHEDULE

The testing activities and milestones are dependent on the development phases. The testing is split up in 6 phases. Each phase consists of a full test cycle with test planning, designing, developing, executing and evaluation.

The 6 phases will be assigned a start and end date and will be documented separately. The test schedule will be reviewed by the project manager.

An Example of a milestone task is:

Milestone Task	Start Date	End Date
Phase 1 Test Planning Test Design Test Development Test Execution Test Evaluation	1st May	7Th May
Phase 2 Test Planning Test Design Test Development Test Execution Test Evaluation	7th May	14th May

7.0 CONTROL PROCEDURES

Problem Reporting

ID	Unique identifier given to the defect. (Usually, automated)
Project	Project name.
Product	Product name.
Release Version	Release version of the product. (e.g. 1.2.3)
Module	Specific module of the product where the defect was detected.
Detected Build Version	Build version of the product where the defect was detected (e.g. 1.2.3.5)
Summary	Summary of the defect. Keep this clear and concise.
Description	Detailed description of the defect. Describe as much as possible but without repeating anything or using complex words. Keep it simple but comprehensive
Steps to Replicate	Step by step description of the way to reproduce the defect. Number the steps.
Actual Result	The actual result you received when you followed the steps.
Expected Results	The expected results.
Attachments	Attach any additional information like screenshots and logs.
Remarks	Any additional comments on the defect.

Defect Severity	Severity of the Defect. (See Defect Severity)
Defect Priority	Priority of the Defect. (See Defect Priority)
Reported By	The name of the person who reported the defect.
Assigned To	The name of the person that is assigned to analyze/fix the defect.
Status	The status of the defect. (See Defect Life Cycle)
Fixed Build Version	Build version of the product where the defect was fixed (e.g. 1.2.3.9)

This is a template of what the report document should look like. This template was taken from http://softwaretestingfundamentals.com/defect-report/. The errors will be documented in this template format and will be shown to the developers who can use it to fix the defects and make the game better. The errors will be reviewed by the project manager so that it will save time and effort. The report will be very specific and objective.

Change Requests

Change Requests are used to document and track defects, enhancement requests and any other type of request for a change to the game. They provide a record of decision and ensure that change impacts are understood across the project. The document will show if that the changes will affect any of the other parts of the game. The change requests will be signed off by the project manager.

8.0 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

9.0 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

10.0 RESOURCES/ROLES & RESPONSIBILITIES

The table below shows the staff that was involved in the test project.

Name	Responsibilities
Mary Brown (Task Manager)	Provide management oversight Provide technical experience Manage the teams Review all documents before signing off on them
Ray Allen (Test Designer)	Generate test plans Generate test suite Evaluate if the tests were effective
Thomas Baker (System Tester)	Execute tests

	Document the results Document the defects Report errors to task manager
Adam Gray (Test Engineer)	Prepare batch tests Execute tests in sequential order Document results
Participants (Testers)	Test the final product Give feedback on the game
Jack Smith (Designer)	managing, designing, preparing, executing, and resolving the test activities

11.0 SCHEDULES

Major Deliverables

Deliverables	For	Date
Test Plan	Project Manager QA Manager Test Team	7th May
Test Cases	Project Manager Test Team	14th May
Test Incident Reports	Project Manager Stakeholders	21st May
Test Summary Reports	Project Manager Stakeholders	24th May

12.0 RISKS/ASSUMPTIONS

A test plan is very prone to risks, some of these risks can be:

- Tight deadlines
- Undefined project scope
- Not enough resources
- Wrong resources
- Requirements may change

To avoid deadlines, staff will need to work overtime and make sure the test plan is on course. The project manager must make sure the resources arrive on time and are the correct resources required. Another way to avoid these risks may be to make a list of what could go wrong and prepare for them

13.0 TOOLS

The following tools were used to test The Pixel Wizard:

- Unity Test Framework
- AltUnity Tester
- Unity Test Runner
- Trello Bug Detector
- RayGun
- SmartLook