# Test Plan Template

Test Plan Template:

The Pixel Wizard

Prepared by:

Simon O’Sullivan

(Date)

TABLE OF CONTENTS

1. INTRODUCTION

A brief summary of the product being tested. Outline all the functions at a high level.

//////////////

2.0 OBJECTIVES AND TASKS

2.1 Objectives   
Describe the objectives supported by the Master Test Plan, eg., defining tasks and responsibilities, vehicle for communication, document to be used as a service level agreement, etc.

The objective of the test is to verify that the functionality of The Pixel Wizard works according to the specifications.

The tests will help identify, fix and retest all issues with the code to ensure that this game meets the game requirements and design document. The tests will also help increase the control flow, logic flow, data flow coverage as well as to stimulate user actual flow. We will be using the Design document as the service level agreement, as it explains exactly how the game is meant to be.

If we encounter a defect, we will log it, take a screenshot and explain the defect before adding it to the manual where all the defects will be stored. This will be given to the software development team.

2.2 Tasks   
 List all tasks identified by this Test Plan, i.e., testing, post-testing, problem reporting, etc.

* Test data will be provided by dev team before functional testing begins.
* In each testing phase, a cycle will be initiated if the defect rate is high in the previous cycle.
* With the test data, we will test the game and examine the response to see if it satisfies the game requirements
* All the defects would come along with a snapshot PNG format, steps to reproduce the defect, name of person who found it and urgency of the defect.
* Dev team will provide defect fix plans based on the Defect meetings during each cycle to plan.
* Any defect fixes planned will be shared with the Test Team prior to applying the fixes on the Test environment

3.0 SCOPE   
General

This section describes what is being tested, such as all the functions of a specific product, its existing interfaces, integration of all functions.

In Scope :

* Player
* Enemy
* Player/Enemy projectiles
* Health and health pickup
* The platforms
* Main menu
* The in-Game Menu.

Tactics

List here how you will accomplish the items that you have listed in the "Scope" section. For example, if you have mentioned that you will be testing the existing interfaces, what would be the procedures you would follow to notify the key people to represent their respective areas, as well as allotting time in their schedule for assisting you in accomplishing your activity?

4.0 Testing Strategy   
Describe the overall approach to testing. For each major group of features or feature combinations, specify the approach which will ensure that these feature groups are adequately tested. Specify the major activities, techniques, and tools which are used to test the designated groups of features.

The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation of the time required to do each one.

4.1 Unit Testing   
Definition:

Unit Testing is a level of software testing where individual units/components of a software are tested. A **unit** is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

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:

List the names of individuals/departments who would be responsible for Unit Testing.

Responsible to do the Unit Testing will be the development team.

Methodology:

Describe how unit 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 the sequence of events of Unit Testing and how will the testing activity take place?

Unit Tests will be preformed on every new method added to the project to ensure it is working as intended. The test will consist of an expected answer to be compared to actual answer and Boolean will let us know if it passes or fails the test. The Unit Testing will be done by the development team, specifically the developer who created the new method that was added to the project. The method should only be added to the version control, along with the test, if it passes its Unit Test.

4.2 System and Integration Testing   
Definition:

System and Integration Testing is testing the project when all the units and modules are integrated together to see if they work without any issues

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

Participants:

Who will be conducting System and Integration Testing on your project? List the individuals that will be responsible for this activity.

System and Integration testing will be done by both the development team and testing team. The development team will do it first before the project is sent over to the testing team to test it again. This is to ensure less risk of bugs getting through the testing phase and also to send something worth testing to the testing team.

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?

System & Integration testing will be done every time a new module is added to the project. Testing will be performed on the 3 layers:

* Data state within the integration layer
* Cross check the web service request using XSD and WSDL.
* Run some unit tests and validate the data mappings and requests.
* Data state within the database layer
* Check if all the data from the integration layer has reached successfully at the database layer and has been committed.
* Validate the constraints and data validation rules applied in the database as per business specifications.
* Data state within the application layer
* Check if all the required fields are visible in the UI.
* Execute some positive and negative test cases and validate the data properties.

4.3 Performance and Stress Testing   
Definition:

**Performance testing** is carried out to check the system's **performance** under varying loads. **Stress testing** is carried out to check the behaviour of the system under the sudden increased load.

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 out 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:

User Acceptance Testing - the software is tested in the "real world" by the intended audience to ensure the software can both handle real-world tasks and perform up to development specifications.

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.

User Acceptance Testing will be the performed by end-users, as they are the target audience for the game.

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?

The project will be sent through User Acceptance testing. First UAT testers will analyse the requirements, so they know what to look out for. Next UAT plan will be developed for UAT testers to follow. Preparation of Test data is next. Should use live data, it will all be secured. Run test cases and report bugs if any. Once UAT testing is complete, UAT testers need to send an email to confirm if the game is ready for production.

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. Automation is used for regression testing because the modules should still all work the same as before. Automated rerunning to see if everything still works as expected.

Participants:

Software quality assurance team will take care of the Automated Regression Testing

Methodology:

Regression testing will be conducted by the SQA Team through automation, as it is tedious and repetitive task. Tests will retest all functionalities of the game to ensure nothing has broken due to an improvement to the code base.

4.7 Beta Testing   
 Methodology:

5.0 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.

6.0 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.

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   
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 the test 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
* Test Incident Reports
* Test Summary Reports

11.0 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).

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