**Test Plan Template:**

Pixal Wizard

**Prepared by:**

Noel Melia

2/5/20

**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**

This test plan describes the strategy that will be carried out to verify that the product/game is working properly with minor errors from the design specification and other requirements. The document will assist the tester and staff to the testing that has been carried out

This is a Horizontal Shooter game that is one player. It has many levels with each level getting harder as the player moves through each level through the enemy health increasing and more enemies to kill in each 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 this test plan is to find out as many game defects as possible and get the game finished and ready to be created. We will begin by stating the main game components and we will expand on them in tasks. The Main Components are as follows:

1. Front End
2. In Game Menus
3. Control Mechanisms
4. The Game

These are only but a few of the main components. In the following Tests the team will testing these in detail and check to see if there are many defects or errors in the game.

**2.2 Tasks**

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

Main Tasks of this game are:

1. Menu at beginning is working correctly
2. Each button is connected to the correct level or menu
3. Quit the game
4. Load the game
5. Save the game
6. Pause the Game
7. Delete the game Saved
8. Controls are connected to what they are said to do
9. Player Moves Easily
10. Settings can be adjusted
11. Enemies die when hit with bullets
12. Player dies when hit with projectiles
13. Menu Logo Pauses the game Easy to follow through levels
14. Pickup items can be picked up

These are but a few tasks to be tested and they will be more as I go along.

**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.

Main Functions to be tested are as follows:

1. Buttons
2. Controls
3. Levels Linked
4. Player Dies
5. Enemy Dies

**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?

1. Buttons on Main Menu
   1. For this to be tested, I will have to Start the game and check that the buttons are linked correctly to the corresponding fields, such as Play Button will bring the player to Level 1 of the game and Settings Button will be brought to adjust the game settings like the volume and music levels.
2. Controls of the Player
   1. The game begins and the player will be presented with a pop up menu showing the player the controls in how to control and move the player. The player if using the PC/Mobile Device, will then press the corresponding keys to see if the game will respond to the directory.
3. Levels Linked to Each Other
   1. The player will have to play the game and finish the first level by defeating all the enemies and getting the through the levels with a small bit of health. Once completed the player will exit and be brought to the next level.
4. Player Dies and What Happens
   1. Player is hit by projectiles and killed,
5. Enemy Dies and What Happens
   1. Player kills the enemy and what exactly happens to the enemy. Doe she explode or what exactly happens to him.

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:**

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.

The most 'micro' scale of testing; to test functions or code modules. It requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code; may require developing test driver modules or test harnesses.

**Participants:**

Programmers

**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?

4.2 System and Integration Testing

**Definition:**

Testing of combined parts of an application to determine if they function together correctly. The parts can be code modules, individual applications, client, and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

**Participants:**

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

**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:**

Used to describe such tests as system functional testing while under unusually heavy loads, heavy repetition of certain actions or inputs, input of large numerical values, large complex queries to a database system.

**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:**

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**, 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**

**Definition:** Group of tests executing sequentially one by one is called Batch Testing. Every test Batch consists of mutiple dependent test cases. In those batches every end state is base state to next case.

**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:** testing when development and testing are essentially completed, and final bugs and problems need to be found before final release. Typically done by end-users or others, not by programmers or testers

**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.