Appendix 3

Test plan document

Test Plan Template: Pixel Wizard

Prepared by: Grace Keane

05/05/2020

Table of Contents

[1.0 INTRODUCTION 3](#_Toc39603296)

[2.1 Objectives 4](#_Toc39603297)

[2.2 Tasks 4](#_Toc39603298)

[3.1 General 5](#_Toc39603299)

[3.2 Tactics 6](#_Toc39603300)

# 1.0 INTRODUCTION

**This test plan describes the testing approach and overall framework that will drive the testing of the “Pixel Wizard” 2D application.**

The customer wants a perfect game, which passes the full cycle of manual testing. Given the specificity of the application it is very important to have the same quality as portrayed in the appendix 2 documentation.

The test plan has been created to facilitate communication within the team members. It describes approaches and methodology that will apply to the unit, integration and system testing of the game design document. It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones and approach. This document has clearly identified what the test deliverables will be and what is deemed in and out of scope.

*Brief overview of the application itself*

This goal is focused on creating a test plan document for a 2-Dimensional shooter game called Pixel Wizard. The main theme is set in a forest. The background is used repeatably to make the level seem longer. On the top left and right of game play there is health representations for both player and enemy. These are displayed by red and blue diamonds. Players have an opportunity to increase their health using a health pickup which is located on a platform in game play. There are three characters to this game, they are player, enemy and boss. These characters are programmed to attack. The main theme of this game is "kill or be killed". If the player is successful, they will progress to other levels and soon win the game overall.  
  
The game has various requirements which will allow this application to be of high quality and fit for purchase. The game has a start-up main menu which contains buttons for settings, load game, delete game, exit game and a play game button which takes the player from the title screen to the point that gameplay begins.  
  
An in-game pause menu is also included. Once the game has been paused using the space bar, the player can access settings or exit the game, a save game option will also be presented, allowing the player to save the game state.  
  
The way in which the player controls the game entities is relatively simple which will cause no confusion to the player. Control is different for both laptop and mobile phone.

2.0 OBJECTIVES AND TASKS

### 2.1 Objectives

1. Ensuring that the software under test is bug free before release.
2. Gaining confidence in and providing information about the level of quality.
3. To make sure that the result meets the business and user requirements.
4. To ensure that the application satisfies the client.
5. To gain the confidence of the customers by providing them a quality product.
6. Find as many software defects as possible.
7. Result should be a production-ready software.

**Document to be used**  
Appendix 2 – Game Design Document

### 2.2 Tasks

* Ideally one test must be carried out at a time to avoid tracking errors or bugs arising.
* Small, clear and non-complex test cases should be written.
* Exit criteria and test closure must be fully thought out.
* Testing must be carried out at an early stage to leave time for possible critical errors.
* Different independent test cases must be carried out to avoid pesticide paradox.
* Work must be evenly divided amongst testers.
* If problems arise, they must be recorded and solved quickly.
* Main components of the game must be tested individually and then tested again when combined with the rest of the application.
* Post testing must be fully done when application is complete.

3.0 SCOPE

### 3.1 General

*General idea of what will be tested*

* Exploratory test opening main menu functionality and check if all buttons work as expected.
* Play game - should take the player into the game and the player should begin at Level 1.
* Settings - should navigate to another page which will allow the player to edit game settings. Game settings like sound and music volume will be tested also.
* Load game - this should allow the player to choose which level to load. Test can the user select a game and note does to game navigate to the correct one selected.
* Delete game - should allow players to delete a games history.
* Exit game – should quit the application.
* Exploratory test the in-game menu to check if it is working correctly with no bugs.
  + Once in game and user clicks the required key (Laptop – spacebar, Mobile – button on top right of screen). Program should navigate to pause screen.
  + Save game – should save position and status of current game being played.
  + Settings – should navigate to the settings page. Also allows for change of sound and music volume.
  + Exit game – allows users to exit the game being played.
  + Resume game –The player can resume the game by selecting the appropriate option or simply pressing the assigned button for pausing/ resuming the game. When key clicked it should enable the game to be played again at current position and status.
  + Restart level – should reset the entire level.
* Fully test in-game functionality and design using functionality testing, compatibility testing, regression testing.
  + Test player navigation with correct key specified in document.
  + Test crouch defence mechanism.
  + Test jump mechanism.
  + Test if player can shoot using key specified.
  + Note if enemy can shoot at player.
  + If player gets hit by an enemy’s bullet check if health score decreases.
  + If enemy gets hit by the players bullet check if health score decreases.
  + hit Note can a health increase be accessed by player touching the “+” symbol.
  + Test whether player can get hit and eventually die.
  + Test whether enemy can get hit and eventually die.
  + Test whether player can move onto next level when level one has been completed successfully.

### 3.2 Tactics