**Test Planning Project**

**Test Plan Template: (The Pixel Wizard)**

**Prepared by: (Abdalla Fadul)**

**(11/05/2020)**

**TABLE OF CONTENTS**

1. INTRODUCTION
2. TEST ENVIRONMENT

3.0 OBJECTIVES AND TASKS

3.1 Objectives

3.2 Tasks

4.0 SCOPE

5.0 Testing Strategy

5.1 Unit Testing

5.2 System and Integration Testing

5.3 Performance and Stress Testing

5.4 User Acceptance Testing

5.5 Batch Testing

5.6 Automated Regression Testing

5.7 Beta Testing

6.0 Test Schedule

7.0 Control Procedures

8.0 Features to Be Tested

9.0 Features Not to Be Tested

10.0 Resources/Roles & Responsibilities

11.0 Schedules

1. Risks/Assumptions

13.0 Tools

1. **INTRODUCTION**

In general, in game testing there are two different forms of testing that can be categorized as Black-Box and White-Box testing. These definitions are well-known in software testing world but process-wise aren’t different with game testing. Only their goals are slightly different in game testing.

Black-Box testing - focus on the functional and overall playability aspects of the game. In this type of testing, for example, testing the graphical user interface, user experience or any visual appearance is in the key roll. Menu, graphical elements, special effects, animations.

White-Box testing – focuses on the architectural, integration and systematic aspects of the game.

1. **Test environment**

The program will be compiled and tested on a pc with windows 10 Operating System as well as a Mac. The compiler used will be compiler included in Eclipse integrated development environment (IDE).

**3.0 OBJECTIVES AND TASKS**

3.1 The objective of this test plan is to ensure that the quality of this game specifications meets the client requirements, and also to ensure that Bugs and defects issues in different levels of the game are identified and fixed and to make sure that the game works flawlessly before releasing it.

3.2 Tasks

* Make list of the bugs and glitches of the game as I play it
* Perform the quality assurance by focusing on the game details to effectively document any glitches or bugs
* Navigate all menus to ensure that everything works correctly
* Test the game compliance with the requirements

1. **SCOOPE**

Main components:

1. I am looking for 100% decision coverage
2. I am looking for there to be no critical bugs

Software testers will do the testing:

* John Murphy
* Peter Parker
* Clark Kent
* Mary Higgins

Functions that has been tested

* The Start button which starts up the game
* The option of letting the player playing the game
* The option of letting the player accessing game setting
* The option of letting the player exiting the game
* The option of letting the player load the game
* The option of letting the player delete the game
* All levels working properly
* The player can move
* The player can jump
* The player can shoot
* If player dies what will happen
* Enemy movement
* Enemy can kill the player
* If enemy die what will happen
* The player can pause the game and resume it
* The scoring
* Animation (the like and the feel of movement, realism)
* Platform compatibility

**5.0 Testing Strategy**

Description of individual test Cases

Some tests cab be performed simultaneously if there are no obvious errors in one test and the current state of the game allows for further sequential testing of another test case

**Test 1**

Test objective: test if the player can start the game after selecting Play Game

Test Description: The Paly Game button will be clicked, and the tester will assess whether it is behaving (also as fast) as it should be

Expected result: Clicking the Play game button will take the user to the gameplay and should begin with level one

**Test 2**

Test objective: Test whether Setting button functions appropriately and that its contents function appropriately i.e. the option to run sound/ music on/off, the button that takes the user to the High Score screen and the button that takes the player to the achievements screen

Test Description: The Setting button will be clicked and the tester will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the setting screen appears and contains the proper buttons, i.e. High Scores, Sound/Music: On/Off, Achievements etc, additionally, these individual buttons/screens will be tested as well to ensure they function as desired

Expectation Result: This test will show the contents of the setting screen, including the buttons for the high scores screen, the Achievements screen, and the button for the option for turning on and off the music and sound effects. Additionally, the High Score button will take the user to the High Scores screen, and the Back button on this screen will go back to the setting menu. Also, on the Achievements screen the user is presented with the achievements they have already received as well as the Achievements they have yet to achieve, and there is a Back button on this screen to get back to the Setting screen. The sound/Music: On/Off button, when pressed will alternate between silencing the game and allowing the music and sound effects to be heard. There will be sound effect confirming when the music is turned on

**Test 3**

Test Objective: Test that the Exit button on the Main menu exit the game properly

Test Description: The tester will click the Exit button on the Main Menu and the game will exit without any errors

Expected Result: Command will return once more to the operating System and the game will have exited with no errors

**Test 4**

Test Objective: Test that the Load button will restart the game

Test Description: The tester will click the Load button on the Main Menu and the game will restart and the tester will assess whether it is behaving as fast as it should behave

Expected Result: after clicking the Load button game will restart on level one

**Test 5**

Test objective: Test that that the user can pause the game at any level and resume it again

Test Description: The tester will check if there is an option/button that pause the game at any level and the option to resume the game from the same level

Expected Result: after clicking the pause and resume buttons the game will start from where the user pauses the game without any errors

**5.1 Unit Testing**

Unit testing will be the first level of game testing, which is performed before other testing methods it will be isolated to ensure a unit does not rely on any external code or functions, and it will be done automated. This test comprises of three stages: Plan, cases and scripts and the unit itself. In the first step, the unit test is prepared and reviewed. The next step is for the test cases and the scripts to be made, then the code is tested

**Participants:**

1. John Murphy
2. Mary Higgins

**5.2 System and Integration Testing**

System testing is testing of the game as whole to check if the system is complaint with the user requirements. It is an end to end user perspective testing intended to find defects in the software system. It is a type of black box testing technique thus the knowledge of internal code is not required. It is a high-level testing which will be performed after integration testing.

The system testing is performed to check the following points:

* To check whether the software system is made according to the customer needs written in the game requirements specifications, it meets both functional and non-functional design requirements of the system
* When all modules are combined as a whole, many errors may arise which may not give the expected result, so system testing is performed to find the defects or bugs in all the interface as well the whole system

Integration testing tests the interface between modules of the software application. The different modules are first testing individually and then combined to make a system. Testing the interface between the small units or modules is integration testing. It will be conducted by software integration tester.

The different techniques for integration testing:

* Big Bang integration testing where the all the modules are combined first and then tested together
* Top Bottom integration testing which take place from top to bottom used Stubs which are substitutes of components. The top module is tested first
* Bottom to Top integration testing which take from bottom to top and uses drivers which are substitutes of components. The bottom module is tested first

Integration testing is performed to check the following points:

* To check whether the modules when combined are according to standards and gives the expected results
* When modules are combined, sometimes the data travelling between modules has many errors which may not give the expected results. So, integration testing is performed to find the defects or bugs in all the interfaces
* To check the integration between any third party is used

**Participants:**

System testing:

* Peter Parker
* Clark Kent

Integration testing:

* John Murphy
* Mary Higgins