The Pixel Wizard – Test Plan

**Prepared by:**

Connor Brookfield (G00361129)

12/05/2020

Table of Contents

[**1.0 INTRODUCTION** 2](#_Toc40378833)

[**2.0 OBJECTIVES AND TASKS** 2](#_Toc40378834)

[2.1 Objectives 2](#_Toc40378835)

[2.2 Tasks 3](#_Toc40378836)

[**3.0 SCOPE** 3](#_Toc40378837)

[**4.0 TESTING STRATEGY** 3](#_Toc40378838)

[4.1 Unit Testing 3](#_Toc40378839)

[4.2 System and Integration Testing 4](#_Toc40378840)

[4.3 Performance and Stress Testing 4](#_Toc40378841)

[4.4 User Acceptance Testing 5](#_Toc40378842)

[4.5 Automated Regression Testing 5](#_Toc40378843)

[4.6 Beta Testing 6](#_Toc40378844)

[**5.0 TEST SCHEDULE** 6](#_Toc40378845)

[**6.0 CONTROL PROCEDURES** 7](#_Toc40378846)

[**7.0 FEATURES TO BE TESTED** 7](#_Toc40378847)

[**8.0 FEATURES NOT TO BE TESTED** 8](#_Toc40378848)

[**9.0 RESOURCES/ROLES & RESPONSIBILITIES** 8](#_Toc40378849)

[**10.0 SCHEDULES** 9](#_Toc40378850)

[**11.0 RISKS/ASSUMPTIONS** 9](#_Toc40378851)

[**12.0 TOOLS** 9](#_Toc40378852)

# **1.0 INTRODUCTION**

The Pixel Wizard is a platformer game in which the player-controlled character must jump and climb between suspended platforms while avoiding obstacles and killing enemies and bosses to gain.

Once the player has opened the game, they will be presented with three options: ‘Play’, ‘Settings’, and ‘Exit Game’. Selecting ‘Play’ will take the player into the game and load the first level. The game will start immediately. From here, the player can progress through the level and once completed and once completed, the next level will load. At the start of the first level, text will appear on-screen informing the player of the control screen. The game will feature at least three levels, with each increasing in difficulty. This could range from having more and more enemies in the progressive levels, enemies having more health, the player character having less enemies, etc. Once the player has completed all the levels, the player will be presented with the option to either start again from the first level or quit the game. If the player chooses ‘Settings’ instead, they can instead adjust the sound level or music level. ‘Exit Game’ will close the game.

# **2.0 OBJECTIVES AND TASKS**

## 2.1 Objectives

The objective of this test play is to make sure that all areas fit the design specifications. Including:

* Menu Items
* Options menu
* Jumping mechanics
* Movement mechanics
* Damage mechanics
* Shooting mechanics
* Progression mechanics
* Health mechanics
* Enemy mechanics

## 2.2 Tasks

To using multiple testing strategies on the individual functions listed in the objectives, prioritising the player mechanics before prioritising enemy mechanics, then damage, shooting mechanics, health, progression, menu items, options menu and exit game.

# **3.0 SCOPE**

**General:**

Testing the functions included in the objectives, tasks, and staying within the design specifications.

**Tactics**:

Prioritise the objectives as outlined in the tasks, in order.

# **4.0 TESTING STRATEGY**

To ensure that all defined functionality in the design specifications work, using multiple testing methodologies, as defined below.

## 4.1 Unit Testing

**Definition:**

Testing individual components and parts of the game individually, prioritising the items in the order below:

1. Jumping mechanics
2. Movement mechanics
3. Shooting mechanics
4. Damage mechanics
5. Health mechanics
6. Enemy mechanics
7. Progression mechanics
8. Menu Items
9. Options menu

**Participants:**

* Jason Statham, Senior Unit Tester
* John Claud, Junior Unit Tester

**Methodology:**

The testing will be done onsite, with the test case to be written up by the Senior Unit Tester and tested by the Junior Unit Tester.

If the testing of a component fails, the tester should stop the test, write a bug report, inform the project manager.

## 4.2 System and Integration Testing

**Definition:**

Testing the game on the platforms outlined in the project specifications, to ensure compatibility.

**Participants:**

* Niall Rogers – Senior Systems and Integration Tester
* Sean Paul – Junior Systems and Integration Tester

**Methodology:**

Test the interfaces between the different components and its interactions with the different operating systems, hardware and file systems outlined in the projects design specifications.

## 4.3 Performance and Stress Testing

**Definition:**

To ensure that the game will still run with desired performance/stability under heavy loads and doesn’t fail catastrophically, and to maximise the running efficiency of the game.

**Participants:**

* Jeff Bezos – Senior Performance and Stress Tester
* Tim Henson – Junior Performance and Stress Tester

**Methodology:**

The Senior Performance and Stress tester will write the scripts for testing, and the Junior Performance and Stress tester will carry out the tests. The tests to be carried out include

* Spawning as many enemies as possible, and recording the max stable amount
* Use different machine specifications, to determine the minimum and recommended system specifications for the game.

## 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 game compare the game to its initial requirements. Users, timeframe, how their feedback is collated and analysed.

**Participants:**

Participants include a selected group of people/customers who fit the target market.

**Methodology:**

Game will be released to a closed group of people/customers (off site) who will play the game and report feedback regarding bugs, new features, performance, and enjoyability.

## 4.5 Automated Regression Testing

**Definition:**

Testing the game again after previous modification, to ensure that defects/bugs have not been introduced, as a result of the previous revisions of code.

**Participants:**

* Martin Clune – Senior Regression Tester
* Billy Ocean – Junior Regression Tester

**Methodology:**

Testing of previously tested components will be tested again by the Senior and Junior Regression testers to ensure that not bugs have been uncovered. If bugs are found, the bug will be documented and escalated to the Project manager.

## 4.6 Beta Testing

**Definition:**

Customer/user testing where the developer is not present to guide the user/customer through the application.

**Participants:**

Customers and users.

**Methodology:**

Customers/user will play the game and send records of incidents/results to the development team on a regular basis.

# **5.0 TEST SCHEDULE**

Finishing each testing phase/type in order presented in this document, within the following timeframes:

1. *Unit Testing – 4 days*
2. *System and Integration Testing – 1 week*
3. *Performance and Stress Testing – 2 days*
4. *User Acceptance Testing – 1 week*
5. *Regression Testing – 1 week*
6. *Beta Testing – 1 month*

# **6.0 CONTROL PROCEDURES**

**Problem Reporting:**

When an incident or bug is encountered, the escalation process is as follows:

1. The tester documents the incident/bug
2. The report is delivered to the Project Manager
3. The Project Manager reviews the report, and determines the priority of the issue
4. Project Manager then informs the development team, who fix the issue
5. The module goes back to the testers, where this process is repeated if another issue/bug has been uncovered.

**Change Requests:**

Any testers can raise a change request which may include changing features, functionality, and structure. The Project Manager signs off on the request if approved based on assessing the feedback, benefits, cost and timeframe of the change request.

# **7.0 FEATURES TO BE TESTED**

Features to be tested include:

1. Jumping mechanics
2. Movement mechanics
3. Shooting mechanics
4. Damage mechanics
5. Health mechanics
6. Enemy mechanics
7. Progression mechanics
8. Menu Items
9. Options menu

# **8.0 FEATURES NOT TO BE TESTED**

The following features do not have to be tested against each other due to their lack of relationship.

* Jumping, movement, shooting and damage mechanics do not have to be tested against progression, menu items, or the options menu.
* Enemy mechanics do not have to be tested against menu items or the options menu.
* Progression mechanics do not need to be tested against menu items and the options menu.

# **9.0 RESOURCES/ROLES & RESPONSIBILITIES**

The staff members and their roles for the following project are defined below, as selected by the Project Manager – Joe Dolce.

* Unit Testing
  + *Jason Statham, Senior Unit Tester*
  + *John Claud, Junior Unit Tester*
* System and Integration Testing
  + *Niall Rogers – Senior Systems and Integration Tester*
  + *Sean Paul – Junior Systems and Integration Tester*
* Performance and Stress Testing
  + *Jeff Buckley – Senior Performance and Stress Tester*
  + *Tim Henson – Junior Performance and Stress Tester*
* User Acceptance Testing
  + *Game tester 1 – John Joe*
  + *Game tester 2 – Joe Duffy*
  + *Game tester 3 – Sean Connery*
* Regression Testing
  + *Martin Clune – Senior Regression Tester*
  + *Billy Ocean – Junior Regression Tester*
* Beta Testing
  + *Google Play store participants*
  + *Apple App store participants*
  + *Steam store participants*

# **10.0 SCHEDULES**

The following documents should be maintained and created at every stage of the testing process:

* Test Plan
* Test Cases
* Project risks and assumptions
* Test Incident Reports
* Test Summary Reports

# **11.0 RISKS/ASSUMPTIONS**

Some high-risk assumptions of the test plan include:

* Not meeting deadlines for each test scenario as outlined by the Project Manager.
* The beta testers will find important bugs within the timeframe.
* Unit testing does not identify key runtime issues.
* Leaving issues unaddressed.

# **12.0 TOOLS**

Some tools to be used for testing this project include:

* “T Plan” - An automation tool for desktop and mobile game GUI testing.
* Zoho BugTracker – Simple, fast and scalable bug tracker to improve bug management efficiency.