Prepared by: (Conor O’ Donoghue)

(11/05/2020)

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

**Introduction**

The following test plan is for Pixel Wizard which is a 2D PC and mobile device game aimed at all age groups. The aim of this test plan is to insure there are no glitches/bugs in the game which could negatively impact our players. In order to insure the testing follows testing standards there is an independent development team carrying out the testing process. This improves the quality and accuracy of our test plan.

**Objectives/Tasks**

Objectives

* Find any possible defects which may have been created during the development process.
* Prevent further defects.
* Insure the game meets user standards.
* Gain confidence in the finished product.
* Insure the game meets the Business requirement specification and System requirement Specifications.

Tasks

Main Menu Test

1. On load the game must present 3 options, ‘Play’, ‘Settings’, and ‘Exit Game’.
2. Clicking Play from the main menu will start the game at level 1

2.1. When Game starts on level one controller screen will display.

3. Clicking Settings on the main menu will display sound and music level options.

3.1 Sound Option can be adjusted.

3.2 Music Option can be adjusted.

4. Clicking Exit Game will close the game completely.

In Game Test

1. At the start of level one the controller options description screen must pop up.
   1. Move forward = Right Arrow key/D
   2. Move Backwards = Left Arrow Key/A
   3. Jump = Up arrow key/W
   4. Crouch = C
   5. Attack = Left Mouse click/R
   6. Pause/Resume = Spacebar
2. At the end of level 1 the player should advance to level 2.
3. Level 2 should have increased difficulty.
4. Level 3 should also have higher difficulty.

**Scope**

The tests being completed are aimed at insuring the overall satisfaction of any players on the game, this means checking that controls are configured correctly, and menus lead to the correct places and there are no graphical or major bugs.   
 In order to complete the testing to the highest testing standards we will be following all major testing guidelines and also completing a range of tests, this includes unit testing, System and integration testing, Performance and stress testing, User acceptance testing, there is then a range of beta test which must be complete, this range from hardware requirements to control procedures and testing schedules.

**Testing Strategy**

A Testing strategy is vital for a test plan to work. The main objective of a testing strategy is to outline what approach will be taken towards the testing. We also need to adhere to a strict timeline and insure we achieve the desired goals. This is to insure any and all glitches or defects are removed from the game before being exploited or discovered.

4.1 Unit Testing

Participants: Conor, Rob

Explained:

Unit testing is when individual pieces of the software are tested. These components are often functions and methods, ranging from character movement, damage, abilities, menu items, etc. Unit testing should have one or more inputs and one output. By testing components like this we achieve an extremely specific and accurate result.

Participants:

Unit testing can be done manually however automating the testing process can speed things up further. Unit tests must be completed before any other form of testing may be applied. Rob and Bob took the lead for testing the in game character controls and level difficulty checks.

Methodology

4.2 System and Integration Testing

Explained:

System and integration testing is defined as a type of software testing which is carried out in an integrated hardware and software environment to verify the behaviour of the entire system. Also used to evaluate the systems compliance with its specified requirements.

Applied: