Software Testing

Test Planning Project

By William Vida, G0035677

Contents

[Introduction 3](#_Toc40375052)

[Objectives and Tasks 3](#_Toc40375053)

[Objectives 3](#_Toc40375054)

[Tasks 3](#_Toc40375055)

[Scope 3](#_Toc40375056)

[General 3](#_Toc40375057)

[Tactics 4](#_Toc40375058)

[Testing Strategy 5](#_Toc40375059)

[Unit Testing 5](#_Toc40375060)

[System and Integration Testing 5](#_Toc40375061)

[Performance and Stress Testing 5](#_Toc40375062)

[User Acceptance Testing 6](#_Toc40375063)

[Automated Regression Testing 6](#_Toc40375064)

[Test Schedule 6](#_Toc40375065)

[Control Procedures 6](#_Toc40375066)

[Problem Reporting 6](#_Toc40375067)

[Change Requests 7](#_Toc40375068)

[Features to be Tested 7](#_Toc40375069)

[Resources/Roles & Responsibilities 7](#_Toc40375070)

[Schedules 7](#_Toc40375071)

[Risks/Assumptions 7](#_Toc40375072)

[Tools 7](#_Toc40375073)

# Introduction

The purpose of this Test Plan is to assess all the features and functions of the 2D platformer game “The Pixel Wizard” which will see the player “navigate progressively difficult levels with a wizard type character that uses magic a lá ‘Skyrim’. Each level will have several enemies that the player must defeat to progress. Each level will also have a boss that the player must defeat to progress to the next level. Each level will contain pickups for the player, such as health pickups to replenish the player’s health”.

The game will “Allow the player to control a specific character, that has an important fictional/narrative role”, “Have game statistics and/or relational attributes with other game objects, enemies, and/or the player character”, “Allow the player to take on and navigate the levels using an easy-to-use user interface” and “Have obstacles that the player must overcome, such as enemies and bosses”.

# Objectives and Tasks

## Objectives

To ensure that the game is free from any errors, bugs, or glitches and all the features described in the “Game Requirements Document” work as intended. This test plan will describe the steps for the testing of the front-end, main menu, in-game menu, control mechanisms and the game itself.

## Tasks

The tasks which must be completed are:

* Unit Testing
* System and Integration Testing
* Performance and Stress Testing
* User Acceptance Testing
* Batch Testing
* Automated Regression Testing

The testing team consists of the following:

* Adam
* Bob
* Charles
* David
* Eoin

# Scope

## General

**Front End**

* Selecting the “Play Game” option will bring the player into the game.
* Selecting the “Settings” option will bring the player into the settings menu.
* Selecting the “Load Game” option will allow the player to use his last saved game.
* Selecting the “Save Game” option will allow the player to save his current game.
* Selecting the “Delete Game” option will allow the player to a saved game.

**In-Game Menu**

* Pausing the game will activate the in-game pause menu, the game stops, and the options appear.
* Selecting the “Resume Game” option will bring the player back into the game.
* Pressing the button to pause the game while in the in-game menu will bring the player back into the game.
* Selecting the “Save Game” option will allow the player to save the current game.
* Selecting the “Restart Level” option will allow the player to restart the current level, this includes “resetting the player’s position to where they were when they first started, resetting the enemies’ position, and resetting any and all pickups”.
* Selecting the “Settings” option will allow the player to change game settings such as the sound level or the music level.

**Control Mechanisms**

The controls should do as outlined in the design document which are as follows:

|  |  |  |
| --- | --- | --- |
| ***Action*** | ***PC*** | ***Mobile*** |
| **Move Forward** | Right arrow key/D | Arrow on screen |
| **Move Backwards** | Left arrow key/A | Arrow on screen |
| **Jump Up** | Up arrow key/W | Arrow on screen |
| **Crouch** | C | Arrow on screen (hold) |
| **Attack** | Left mouse click/R | Dedicated button |
| **Pause/Resume** | Spacebar | Button in top right of screen |

**The Game**

When the player selects the “Play Game” option from the main menu the game will start immediately with a text informing the player of the control screen. The player can move as stated previously, the player’s health is represented by red diamonds and is displayed in the upper left-hand side of the screen and the player can pick up powerups such as a health increase. In the upper right-hand side of the screen, the enemy boss’s health is shown, represented by blue diamonds if the boss exists. Enemies, just like the player can shoot projectiles. After the player completes the level, the next level will load. The levels get increasingly more challenging, for example, the enemies will get an increase in their health. Once the player finishes the last level, an option will be presented allowing the player to either restart from the first level or go to the main menu.

# Tactics

All testing is planned to be complete in five weeks, with each test taking one week. All features of the game will be tested to make sure no errors exist.

# Testing Strategy

## Unit Testing

**Definition**

[Unit Testing is](http://softwaretestingfundamentals.com/unit-testing/) “a level of software testing where individual units/components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc”.

**Participants**

Adam.

**Methodology**

Unit Testing is done before Integration Testing. Adam, a software developer, will test the individual features, units, and components of the game to make sure they all do as intend. It is best to use a framework testing tool for the game’s language.

## System and Integration Testing

**Definition**

[System and Integration testing is](https://www.softwaretestinghelp.com/system-integration-testing/) “the overall testing of the whole system which is composed of many sub-systems. The main objective of SIT is to ensure that all software module dependencies are functioning properly and the data integrity is preserved between distinct modules of the whole system”.

**Participants**

Bob.

**Methodology**

[In System and Integration Testing](https://www.guru99.com/system-integration-testing.html), all modules are integrated in advance, and the entire program is tested. But during this process, a set of errors is likely to be encounter. System and Integration Testing uses Black Box Testing, either bottom-up or top-down integration can be used.

## Performance and Stress Testing

**Definition**

[Performance Testing is](https://www.geeksforgeeks.org/difference-between-performance-and-stress-testing/) “a type of software testing that is carried out to determine system performance in terms of sensitivity, reactivity and stability under a particular workload”.

[Stress Testing is](https://www.geeksforgeeks.org/difference-between-performance-and-stress-testing/) “a type of software testing that verifies the stability and reliability of the system. This test particularly determines the system’s robustness and error handling under extremely heavy load conditions”.

**Participants**

Charles.

**Methodology**

Performance Testing can identify problems such as long loading times, high response time and bottlenecking. Some common performance bottlenecks are CPU utilisation, memory utilisation and operating system limitations. Stress Testing like Performance Testing can identify problems at high workloads. Charles will overload the game greatly.

## User Acceptance Testing

**Definition**

[User Acceptance Testing is](https://www.guru99.com/user-acceptance-testing.html) “a type of testing performed by the end-user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done”.

**Participants**

David.

**Methodology**

User Acceptance Testing is usually the last test to be done before the game is released. As it is the last test to be done the game must be tested very intensively and to make sure that a possible bug or glitch is found and is fixed.

## Automated Regression Testing

**Definition**

[Automated Regression Testing is](https://www.guru99.com/regression-testing.html) “a type of software testing to confirm that a recent program or code change has not adversely affected existing features”.

**Participants**

Eoin.

**Methodology**

Automate Regression Testing is best done using automated tests. Eoin will make changes to the code and then make test cases to find out how much the game has changed.

# Test Schedule

* Unit Testing will be done in Week 1.
* System and Integration Testing will be done in Week 2.
* Performance and Stress Testing will be done in Week 3.
* User Acceptance Testing will be done in Week 4.
* Automated Regression Testing will be done in Week 5.

# Control Procedures

## Problem Reporting

If a problem occurs during testing, then the tester must document the problem and send it to their respective superiors.

## Change Requests

If the testers would like to make any changes for the game, they must contact their superiors or the project manager and after a discussion, a decision will be made.

# Features to be Tested

In the front-end, all the options on the main menu will be tested, as well as all the options in the in-game pause menu. The control mechanisms will be tested. The gameplay will also be tested.

# Resources/Roles & Responsibilities

* Adam was the Unit Tester for the game. Adam tested the individual units and components of the game.
* Bob did the System and Integration Testing. Bob will make sure that software module dependencies are functioning properly.
* Charles did Performance and Stress Testing. Charles put a lot of load on the game to verify its stability.
* David did the User Acceptance Testing. David tested the intensively to make sure that there were no errors in the game.
* Eoin did the Automated Regression Testing. Eoin made changes to the game and tested the game after to see if certain features were not negatively affected.

# Schedules

Testing for the game must be done within five weeks, with the commencement date of 1 June 2020.

# Risks/Assumptions

Some risks include:

* A tight schedule might mean some errors could slip through and be present in the final product. To counteract this an extension may be granted.
* The game and its related files might become corrupted after a certain change or test. A backup of the game and files could be created.
* If new defects occur in the game that were not located before, it may delay the commencement of certain tests and they will have to fix before the testers can continue.

# Tools

“Bugzilla” is the bug tracking that will be used. For the automated and manual tests, “Xray” will be used.