Test Plan

**Name of Product:**

The Pixel Wizard

**Prepared by:**

Jack McNamee

28/04/2020

Contents

[Introduction 3](#_Toc39059683)

[Objectives and Tasks 3](#_Toc39059684)

[Objectives 3](#_Toc39059685)

[Tasks 3](#_Toc39059686)

[Scope 3](#_Toc39059687)

[Testing Strategy 4](#_Toc39059688)

[Unit Testing 4](#_Toc39059689)

[System and Integration Testing 4](#_Toc39059690)

[Performance and Stress Testing 4](#_Toc39059691)

[User Acceptance Testing 4](#_Toc39059692)

[Batch Testing 4](#_Toc39059693)

[Automated Regression Testing 4](#_Toc39059694)

[Beta Testing Participants 5](#_Toc39059695)

[Test Schedule 5](#_Toc39059696)

[1. Analyse the product 5](#_Toc39059697)

[2. Define the test strategies 5](#_Toc39059698)

[3. Define the test objectives 5](#_Toc39059699)

[4. Define the test criteria 5](#_Toc39059700)

[5. Plan the test environments 5](#_Toc39059701)

[6. Determine the test deliverables 5](#_Toc39059702)

[Control Procedures 6](#_Toc39059703)

[Problem Reporting 6](#_Toc39059704)

[Change Requests 6](#_Toc39059705)

# Introduction

This is a 2D side-scroller game where the player, a wizard, must progress through levels containing several enemies, each with their own boss, by defeating them with magic. The levels also have powerups for the player, including ones that can replenish the player’s health.

This test plan will document the strategies used to ensure that this game meets its requirements and contains guidelines on how the product is to be tested.

# Objectives and Tasks

## Objectives

* To find defects in the game
* To make sure that the end result meets the requirements
* To prevent defects to help in providing an error free game
* To verify all requirements have been fulfilled
* To evaluate work products such as requirements, user stories, design, and code

## Tasks

* Test the game until the desired result is achieved
* Identify bugs and errors, and notify the team
* Test the completed game on different devices

# Scope

The components of the game to be tested include:

* The levels (enemies, powerups, heath, background, sound effects)
* The controls
* Game assets
* Saving, loading & deleting games
* Main menu, in-game menu, settings
* Performance

I will accomplish the items in the scope by using the testing strategies outlined in the Testing Strategy section below.

Upon completing each item in the scope, I will notify the team of my progress and any bugs or errors I have come across.

# Testing Strategy

## Unit Testing

This is a testing method used to test individual units of the game to determine if they are up to standard. This helps to find errors early in the development cycle.

Test cases will be written for each part of the game, e.g. level one, in order from the main menu that greets the player upon starting, to when the user finishes the game.

## System and Integration Testing

The objective of this testing strategy is to ensure the game functions correctly on the environments it is being created for.

To carry this testing out, individual units of the game will be tested on mobile phones, Apple and Android, to ensure that it works correctly.

## Performance and Stress Testing

This determines how the game will perform under a particular workload, in terms of responsiveness, durability, stability and reliability.

I will test this game out on multiple mobile devices and make sure that the UI, graphics, layout, controls, etc. work correctly.

## User Acceptance Testing

This is usually done by end-users of the product to ensure that it is ready for operational use, by comparing it to its original design requirements.

This will be done when the game is finished, on multiple mobile devices, by different end-users. Their response to the game will be recorded, regarding UI, controls, entertainment, etc.

## Batch Testing

This testing strategy consists of multiple test cases that are dependant of each other. For example, the end of the first test case is the start of the second test case.

To carry out batch testing on this game, I will divide it into multiple parts starting from the main menu until the game over screen.

## Automated Regression Testing

This involves testing units of the game repeatedly to ensure that modifications are not negatively impacting the system.

I will use progressive regression testing, which is testing when certain changes have been implemented, so I can ensure that no features of the game have been compromised.

## Beta Testing Participants

This testing is performed by a limited number of end-users of the application in a real environment to obtain feedback of the product.

This will be the final test before the game is shipped out.

# Test Schedule

## Analyse the product

* Learn more about the game, how it’s designed, how it works, who is it aimed at, what software/hardware it uses

## Define the test strategies

* Define the scope of testing
* Identify the testing strategies
* Document the risks and issues
* Determine who will test and when

## Define the test objectives

* Determine the overall goals of testing the game
* Such as finding defects, ensuring the game is working

## Define the test criteria

* Determine the standards for which a test case can be based upon
* Such as exit criteria

## Plan the test environments

* Determine the end-users to test the game
* Determine the software/hardware requirements

## Determine the test deliverables

* Decide what they are before, during and after testing

# Control Procedures

## Problem Reporting

To report a problem, an email will be sent to the testing team informing them of what was carried out, the errors that occurred and any other notable information.

## Change Requests

Only after each feature has passed testing, changes to the current product can occur to further improve the game.

# Features to be tested

* Game Play
  + Movement of character
  + Controls
  + Levels
  + Enemies
  + Health
  + Background
  + Sounds
* Menu’s
  + Main menu
  + Game over screen
  + Settings
  + In Game menu
* Saving
  + Check if games are saved
* Performance
  + Loading time of game
  + Game flow is fast

# Schedules

Deliverable Documents:

* Test Plan
* Test Strategies
* Test Cases
* Bug Reports
* Test Summary Report
* Installation Guide
* User Guide
* Test Status Report

# Tools

Bug Tracking: backlog

Automation: Unity Test Tools