|  |
| --- |
|  |
| Test Planning Project  Software Testing |
| |  |  | | --- | --- | | G00332617@gmit.ie | Robert Ryan | |

***Contents***

[1.0 Introduction 1](#_Toc40370488)

[2.0 Objectives and tasks 2](#_Toc40370489)

[2.1 Objectives 2](#_Toc40370490)

[2.2 Tasks 2](#_Toc40370491)

[3.0 Scope 3](#_Toc40370492)

[4.0 Testing Strategy 4](#_Toc40370493)

[4.1 Unit testing 4](#_Toc40370494)

[4.2 System and Integration testing 5](#_Toc40370495)

[4.3 Performance and stress testing 5](#_Toc40370496)

[4.4 User Acceptance testing 6](#_Toc40370497)

[5.0 Test schedule 6](#_Toc40370498)

[6.0 Control procedures 6](#_Toc40370499)

[7.0 Features to be tested 6](#_Toc40370500)

[8.0 Resources/Roles & Responsibilities 7](#_Toc40370501)

[9.0 Schedules 7](#_Toc40370502)

[10.0 Risks/Assumptions 7](#_Toc40370503)

[12.0 Tools 7](#_Toc40370504)

Prepared by: Robert Ryan (G00332617)

Date: 08/05/20

# Introduction

The test plan document is to outline the procedure that will be implemented to demonstrate that the game meets the design specifications/requirements. This document consists of guidelines that will aid the people involved in testing this game.

# Objectives and tasks

## Objectives

The objective of this test plan is to ensure that the game is free from input errors as well as all game logic will preform as intended to do so.

* When game is loaded menu is displayed with play, settings and exit buttons displaying and fully functional.
* Once settings is selected users is able to control the volume of the music and game sound effects.
* In game pause menu has full functionality with “Resume game”, “Settings” and “Restart level” and “ Exit game” e.g. If user chooses “Resume” it will allow the player carry on from the exact position they paused the game at.
* Restarting a level in the pause menu will bring the player back to the beginning of the level with all the enemies being reset to their starting position also
* Controls for both PC and Mobile will work fully. “W/Up Arrow”, “A/Left Arrow”, “D/Right Arrow” will control movement for PC while movement on mobile will be controlled by onscreen buttons represented by arrows. “C” will control the crouch ability on PC while on an onscreen button positioned towards the top left of the screen will be used for mobile. “Spacebar” will be used for the pause functionality on PC, with mobile having a button positioned at the top right of the screen to control the pause menu. Left/Right mouse click will allow the user to attack.
* Interaction with enemies will show a deduction in enemies health if attacked by the player or the players health will decrease if the enemy attacks them.
* Health will be displayed for both play and enemy at the top of the screen for both PC and mobile. The health will be represented by diamonds for both player and enemy in separate colours.

## Tasks

* Unit Testing: Unit testing is where individual components of the game will be tested. This is to show that each component of the game performs as designed.
* Integration Testing: Integration testing is where the individual components are combined and tested together as a group. This is to expose the faults between integrated components.
* System Testing: System testing will be used to test the complete and integrated game. The purpose of this is to assess the games compliance with the previously stated requirements.
* Performance Testing: Performance testing is to test the game with focus on responsiveness and stability under a certain load.
* Stress Testing: Stress testing is used to test the game under a heavy load to ensure that the game will not crash.
* User Acceptance Testing (UAT): UAT is the one of the final tests we will carry out, actual game users will test the game to see if it can handle the required tasks in real-world scenarios.
* Automated Regression Testing: Automated regression testing is used to speed up the testing process, as testing everything manually is very time consuming and costly.

# Scope

To begin we will be testing the menu functionality. We need to make sure that every button does as required, Play game button brings the player to the first level. Settings button allows the users to control the audio both sound effects and music in game. Exit game button will close the game for the user when pressed. All these buttons will be tested on PC and mobile.

Next to be tested is how the character reacts to the movement inputs by the user “W, A, D as well as the arrow keys”. “C” for crouching will also be included in the testing of movement. Similarly the onscreen buttons for movement on mobile will also be tested.

The pause menu will be next to be tested as it is the last of the controls to be tested. The spacebar button is what triggers the game to pause and the pause menu to be displayed. The buttons accessible through the pause menu will be, “Resume”, “Restart Level”, “Settings” and “Exit”. The resume button show should allow the user to carry on where they left off before pausing the game. The restart button should reset the user to the beginning of the level and reset everything within the level the user is currently playing. The settings button should allow the user the same functionality as the settings button in the start menu, audio control. The exit game button should as the same as the exit button in the start menu by closing the game when it is pressed.

The next thing to be tested is how the users character interacts when faced with an enemy. The users health should deplete when being attacked by an enemy. When the health is empty the user should have a screen displayed to them that they has died/failed, with an option to restart.

How the user’s attacks affect the enemy, using the left/right mouse click will make the user perform an attack. How the enemy interacts with being attacked will be tested. This is important as all the enemies are not the same so a different amount of hits are needed to defeat different enemies.

Health pickups are scattered throughout the levels, they will be used to replenish the users health bar when acquired. The health pickups will need to be test on how they work when the users attempted to pick one up with full health and when they have been damaged.

Once the user has defeated all the enemies in the level, the users should be loaded into the following level. The game will consist of 3 levels, following the completion of the final level the user will be given the option to start again from the first level or exit the game.

# Testing Strategy

## Unit testing

* Start Menu

The first component to be tested is the start menu. The features of this component to be tested is the setting and exit. We will be testing the play game feature when the first level is integrated, this is due to the play feature being dependent on another component. The setting feature will be tested by changing the volume of the music, the default will be max volume and will be changed to confirm that the music volume will change. The exit button will be tested to show that the game will in fact close when pressed.

* Character Movement

When testing the movement we will ensure that the assigned buttons will act appropriately. W/Up arrow will control the jump movement. D/Right arrow will control the forward movement this will be used to progress through the level. A/Left arrow will control the backward movement and finally C will be used for the crouch movement this will be used when trying to avoid an enemy projectile or to navigate through a smaller area.

* Enemy Interaction

There are several features to be tested with enemy interaction. The first will be how the enemies attacks affect the users character, with there being several different enemy types they deal different amounts of damage. Each enemies attacks will be tested individually to make sure they are dealing the appropriate amount of damage. Following an attack from the enemy that users health bar should decrease depending on how many attacked are landed on the user.

The next feature to be tested is how the enemy reacts to being attacked by the user. Similar to the previous test the rolls are reversed and we are looking for the enemies heath bar to decrease when they are being attacked by the users. When the health bar is fully depleted the enemy should disappear from the users view are the are not dead.

* Level Completion

Upon defeating all the enemies of level 1 + 2 the user will be brought to the following level. Right after the final enemy has been defeated the user will witness a transition from their current scene/level to the following scene/level.

## System and Integration testing

The type of integration testing we will be using is the incremental approach.

* Play Feature (Start Menu)

The reason we wait to test this component in integration is due to there being no visible action to confirm that the play button works with the start menu and level integrated. Once the play button is pressed by the user, the user will be brought into the first level. When the first is loaded the instructions for the controls will be displayed on screen for a short period of time to get the user acclimated to the game.

* Pause Menu

To test the pause menu we have to wait until it is integrated with the levels. Without the pause menu integrated with the levels there is no way to test the component. There are several aspects of this component that will be tested, firstly when the pause button is pressed the whole level should be put on hold all enemies stop moving and attacking. The resume feature will be next to be tested when the user presses resume where ever the user left off will continue from there. Restart is the next feature that will be tested, once pressed the user and their health will be reset to the beginning of the level they were playing, included in this all enemies will be reset as well including their health as if they have just began the level. Settings is the next feature on the pause menu, this is expected to work the same as the settings in the start menu with controlling audio. Finally is the exit this like the setting feature will work the same as the exit that is in the start menu by closing the game when pressed.

* Completion Screen

On the final level once all the enemies are defeated the play will be displayed a win scene where they will have the option to exit the game or begin the game again from the first level. This component can only be tested once integrated with the final level as to invoke this component the user must complete the game.

Following the components that are dependent on other components being integrated we can then add the remaining components and test the game as a whole to make sure the game is fully functional together.

## Performance and stress testing

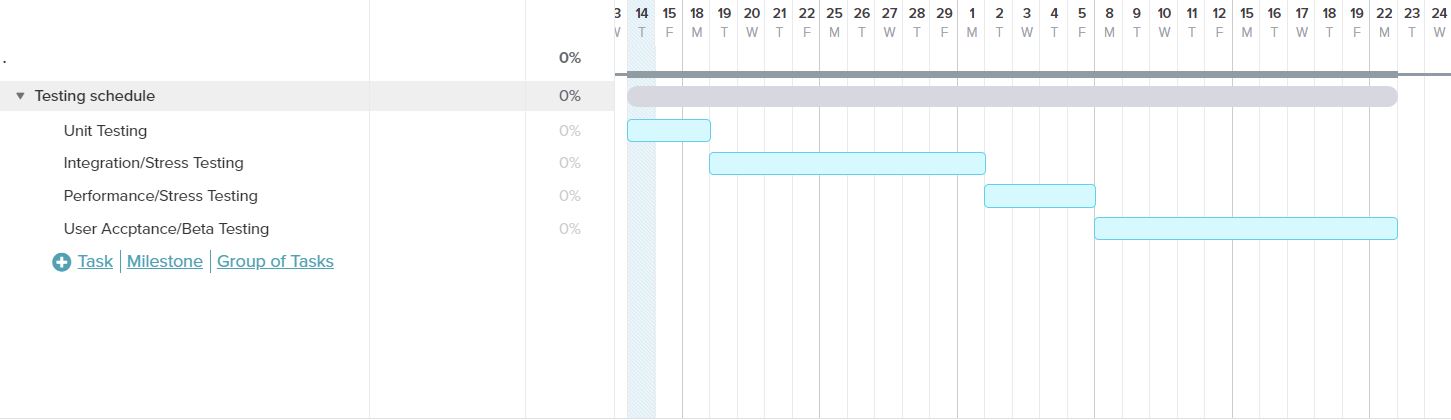
* Number of enemies spawned before game begins showing signs of potential failure or slowing down.

We have a goal of each level getting increasingly more difficult as the user progresses. We aim to accomplish this by increasing the number of enemies per level. We have find the right amount of enemies that can be added before the game does not run as smooth or potentially crashes.

## User Acceptance testing

We aim to carry out UAT in the form of a beta test. We aim to have a controlled test group of people that will test the beta. We are looking for feedback about how responsive the game is, how difficult the levels are. We want to have the game be noticeably harder as the users progress. There is a balance that is needed for how difficult the game can be as we want our users to be challenged but do not want it to be impossible to finish the game.

# Test schedule





# Control procedures

When an issue occurs it is reported to the project manager, following the reporting of the issue the project manager will then assign the appropriate team to address the issue to find a solution.

# Features to be tested

* Start menu
* Character movement
* Enemy movement
* Character attack
* Enemy attack
* Pause menu
* Settings
* Level transition
* Final level completion

# Resources/Roles & Responsibilities

* Jane Doe – Unit testing
* John Smith – Integration testing
* Ian O’Connor – System testing
* Eoghan Murphy – Performance
* James Whelan - Stress testing
* Jamie Moggan – User Testing Supervision
* Robert Ryan – Project Manager

# Schedules

|  |  |
| --- | --- |
| **Deliverables** | **Time Frame** |
| Test plans | 18/05/2020 |
| Test cases | 31 Days |
| Test Incident reports | following each test case completion |
| Test summary reports | Weekly to judge progress |

# Risks/Assumptions

In case of problems occurs that set off course we have contingency plans in place to stay on track to meet our deadline. We have made it available to do night shifts to meet our deadline as well as bringing on extra team members if needed.

# Tools

* Bug tracking tools

***Airbrake.io:*** Airbrake allows you to quickly locate the file, method, and line that caused the exception as well as identify affected users, browsers, URLs, and more. Dig deeper with backtraces, parameters, and other contextual info. Monitor code quality with deploy tracking, error trend graphs, and detailed dashboards.

* Hardware

PC (windows 10 OS)

Mobile (Oneplus 7T (Android), iPhone 10s Max)