

This Is It

Test Plan

Table of Contents

1. Overview..	3
1.1. Purpose.	3
1.2. Scope.	3
2. Testing Summary.	4
2.1. Scope of Testing.	4
3. Analysis of Scope and Test Focus Areas.	5
3.1. Platform Testing.	5
3.2 Error Logging	5
3.3 Controls	5
4. Functional Tests.	6
5. Other Testing.	8
6. Test Environment Plan.	8
6.1. Test Environment Details.	8
7. Assumptions and Dependencies.	9
7.1. Assumptions.	9
7.2. Dependencies.	9

1. Overview

1.1. Purpose

The purpose of this document is to define:

- The test scope, focus areas and objectives
- The test responsibilities
- The test dependencies

1.2. Scope

This document details the testing that will be performed by the project team for the “This Is It” capstone project. It defines the overall testing requirements and provides an integrated view of the project test activities. Its purpose is to document:

- What will be tested;
- How testing will be performed;
- What resources are needed

2. Testing Summary

2.1. Scope of Testing

2.1.1. In scope

The scope will strictly include user acceptance testing of the following criteria:

- Overall Gameplay Experience
- Controls
- Audio
- AI Response
- Level Editor
- Environment Collision
- Building Buffs
- Health / Respawn
- Game Loop
- HUD elements

2.1.2. Out of scope

- Multi-platforms
- Graphics

3. Analysis of Scope and Test Focus Areas

3.1. Platform Testing

All testing will be done on Windows capable PC. A gamepad will be required.

3.2. Error Logging

Unfortunately, GitHub has a cap on file size that prevents us from making the build available there so the Test Build is available for download at:

<https://drive.google.com/file/d/0B6E0MTDwT0PMaG1BYk5tX0VabEk/view?usp=sharing>

All bug logging is to be done through GitHub at:

https://github.com/jdfinnie/ThisIsIt_TestBuild

Testers can log any comments by clicking the “Issues” button on the GitHub page. This will require the tester to log into GitHub.

3.3. Controls

This game uses an Xbox controller. Button mapping is as follows:

In Game:

Left Joystick - Movement (forward, backward, strafe left, strafe right)

Left Joystick Click/Hold Click - Sprint

Right Joystick - Movement (turn, look around)

Left Trigger - Aim

Right Trigger - Fire Weapon

Right Bumper - Use Ability

A Button - Jump

B Button - Change Weapon

X Button - Reload

Y Button - Choose to enter City Editor (Only active at the end of waves from wave 3 onward)

Start Button - Opens pause menu

In Editor:

Left Joystick - Move

Right Trigger(hold) - Reduce movement step size

Right/Left Bumper - Navigate inventory

A Button - Select/place object

B Button - Cancel selection

X Button - Store selected object

Y Button - open inventory

Start Button - pause menu

Select Button - Return to Third-Person map

Menus:

Left Joystick - Change selection

A Button - Select

B Button - Back

4. Functional Tests

This section outlines all the areas that should be considered by the users. The sections below are meant as guidelines. Any bugs or issues not listed below should also be reported.

Progressive levels of difficulty

- How many respawns are used?
- How many waves are completed?
- Does the user feel overwhelmed or too strong?

Animation (the like and feel of the movement, realism, frame rate)

- Are they realistic?
- Timing

Art (character model, texture, terrain or world, crowd, objects, etc.)

- Realism

Camera (follow, zoom in and out (Aiming / Editor)

- Does the camera follow the character?

- Does the camera respond well to zooming? (i.e is the aim negatively affected)
- Does the zoom work / is it necessary in the editor?

Collisions

- Can the player clip through objects/characters?
- Do projectiles impact correctly on surfaces/characters?
- Are buildings able to be selected/placed in the Editor without issue?

Game flow and logic

- Do rounds end as expected?
- Are Editor buildings successfully stored?
- Do changes in the Editor reflect in the third person map?
- Does the game create levels correctly on loading a new level?

Descriptive text

- Does the user understand the command?
- Is the text an appropriate size?
- Is it displayed for long enough?

Music

- Does it provide an appropriate atmosphere for the genre?
- Does it loop correctly?
- Does it play reliably or cut out?

Sfx (weapons)

- Are they in-sync with the action?
- Are they realistic?

Statistics (in-game like health, waves, and ammo)

- Does the user have a good sense of their health limit
- Does the user have a good sense of their current wave / number of enemies remaining
- Does the user have a good sense of their ammo amounts?

The AI logic (for both defensive play and offensive play; player movement and positioning)

- Is there a sense that the enemy is making decisions?
- Can enemies follow a new path when returning from the Editor?
- Do the allies detect enemies effectively?

The condition to advance to the next level (what are the rules?)

- Does the user understand how to reach the next wave?
- Does the user understand the objective and win/lose conditions

The ease of use of the button functions / controls

- Do they work?
- Are the controls intuitive?
- Are they responsive?

The event/object triggers

- Do players/characters spawn correctly?
- Do level transitions trigger?

The game options

- Does the game pause correctly?
- Are the options displayed and selectable?
- Do changes in options save to the game?

5. Other Testing

No other testing is planned at this time.

6. Test Environment Plan

6.1. Test Environment Details

6.1.1. Hardware and Firmware

The user will require a Windows PC (preferably a dedicated gaming computer) and an internet connection to download Unreal Engine game files.

The user will also require a PC Gamepad (such as a wired xbox controller).

6.1.2. Software

The user will require a current build of the capstone project and have the associated game support files from Unreal. These support files will download automatically if the user is connected to the internet.

7. Assumptions and Dependencies

7.1. Assumptions

Detail any assumptions made for testing.

- Developers will not have direct contact with testers.
- If the test lab is available, testers will be monitored during gameplay.

7.2. Dependencies

Detail testing dependencies

- Check that Unreal Engine game files and a current build of the game are installed on the test computer(s)