# Wizard Wars

Prepared by:

12/05/2020

Dylan Creaven

John Smith

Jane Doe

Michael Jones

David James

Teodore Johansson

Table of Contents

[Wizard Wars 1](#_Toc40219021)

[1.0 INTRODUCTION 2](#_Toc40219022)

[1.1. Purpose 2](#_Toc40219023)

[1.2. Project Overview 3](#_Toc40219024)

[1.3. Audience 3](#_Toc40219025)

[2.0 OBJECTIVES AND TASKS 4](#_Toc40219026)

[2.1 Objectives 4](#_Toc40219027)

[2.2. Test Assumptions 4](#_Toc40219028)

[2.3. Test Principles 5](#_Toc40219029)

[3.0 SCOPE 6](#_Toc40219030)

[3.1 Exploratory Testing 6](#_Toc40219031)

[3.2 Functional Testing 6](#_Toc40219032)

[3.3. User Acceptance Test (UAT) 8](#_Toc40219033)

[3.4 Unit Testing 8](#_Toc40219034)

[3.5 System and Integration Testing 9](#_Toc40219035)

[3.6 Performance and Stress Testing 9](#_Toc40219036)

[4.0 TESTING STRATEGY 10](#_Toc40219037)

[4.1. Entry and Exit Criteria 10](#_Toc40219038)

[4.2. Test Cycles 10](#_Toc40219039)

[4.3. Test Metrics 10](#_Toc40219040)

[5.0 TEST SCHEDULE 11](#_Toc40219041)

[6.0 Validation and Defect Management 11](#_Toc40219042)

[6.1 Problem Reporting 11](#_Toc40219043)

[6.2 Defect tracking & Reporting 12](#_Toc40219044)

[7.0 RESOURCES/ROLES & RESPONSIBILITIES 13](#_Toc40219045)

[8.0 SCHEDULES 13](#_Toc40219046)

[9.0 RISKS/ASSUMPTIONS 14](#_Toc40219047)

[10.0 TOOLS 14](#_Toc40219048)

## 1.0 INTRODUCTION

### 1.1. Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the game, Wizard Wars.

The document introduces:

• Test Strategy: rules the test will be based on, including the givens of the project (e.g. start / end dates, objectives, assumptions); description of the process to set up a valid test case (e.g. entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).

• Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes to the defects/bugs found

• Test Management: process to handle the logistics of the test and all the events that come up during execution

### 1.2. Project Overview

Wizard Wars is a 2-D side-scrolling platformer video game. The gameplay consists of the player, a wizard fighting and attempting to defeat many low-level enemies as well as a Boss at the end of each level. It is done in 16-bit art style and is modelled after games in the same genre such as Shovel Knight and The Elder Scrolls V: Skyrim.

### 1.3. Audience

• Project team members perform tasks specified in this document and will provide their personal input and recommendations based on this document.

• Project Manager Plans for the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results.

• Technical Team ensures that the test plan and deliverables are in line with the design, provides the environment for testing and follows the procedures related to the fixes of defects.

## 2.0 OBJECTIVES AND TASKS

### 2.1 Objectives

The objective of the test is to verify that the game Wizard Wars, performs as expected and as detailed in the specifications.

The test will execute and verify the game’s scripts, assets, sprites, and controls, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects.

The final product of the test is twofold:

• A game ready for shipment to stores

• A set of stable game assets to be re-used to future games within the studio.

### 2.2. Test Assumptions

#### Key Assumptions

• Production like data required and be available in the system prior to start of Functional Testing

• In each testing phase, Cycle 3 will be initiated if the defect rate is high in Cycle 2.

#### General

• Exploratory Testing would be carried out once the build is ready for testing.

• Performance testing is not considered for this estimation.

• All the defects would come along with snapshot JPEG format which clearly displays the error alongside context and instructions for how to recreate the defect.

• The Test Team will be provided with access to a private Test Server (TS) via a VPN.

• The Test Team assumes all necessary inputs required during Test design and execution will be supported by Development appropriately.

• Test case design activities will be performed by QA Testing Group.

• Test environment and preparation activities will be owned by Development Team.

• Dev team will provide Defect fix plans based on the Defect meetings during each cycle to plan. The same will be informed to Test team prior to start of Defect fix cycles

• Business Analyst will review and sign-off all Test cases prepared by Test Team prior to start of Test execution

• Project Manager/Business Analyst will review and sign-off all test deliverables

• The project will provide test planning, test design and test execution support

• Test team will manage the testing effort with close coordination with Project PM/Business Analyst

• Project team has the knowledge and experience necessary, or has received adequate training in the system, the project and the testing processes.

• There may be downtime when testing defect/bug fixes and solutions

• The system will be treated as a black box; if the information is correct within the reports and online if will be assumed that the database is also functioning

• Cycle 3 will be initiated if there are more defects in Cycle 2.

### 2.3. Test Principles

• Testing will be focused on meeting the business objectives, cost efficiency, and quality.

• There will be common, consistent procedures for all teams supporting testing activities.

• Testing processes will be well defined, yet flexible, with the ability to change as needed and as the game director’s vision/plan/requirements change.

• Testing activities will build upon previous stages to avoid redundancy or duplication of effort.

• Testing environment and emulate a production environment i.e. in-game as much as possible.

• Testing will be a repeatable, quantifiable, and measurable activity.

• Testing will be divided into distinct phases, each with clearly defined objectives and goals.

• There will be entrance and exit criteria.

## 3.0 SCOPE

### 3.1 Exploratory Testing

**PURPOSE**: These tests are to make sure certain high priority defects are fixed before proceeding onto the next level.

**SCOPE**: Tests menu navigation, player movement, usability and control’s learnability.

**TESTERS**: Testing team.

**METHOD**: this exploratory testing is carried out in the application without any test scripts and documentation

**TIMING**: Done once each new cycle begins.

### 3.2 Functional Testing

**PURPOSE**: Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

**Scope**: Scope of functional testing to test certain functions of the application e.g. health bars depleting as player/boss gets hit, player pick-ups affecting the gameplay.

**TESTERS**: Testing Team.

**METHOD**: The test will be performed according to Functional scripts.

**TIMING**: Tests will be performed once Exploratory testing has concluded.

TEST ACCEPTANCE CRITERIA

1. Test cases approved and signed-off prior to start of Test execution.
2. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects.
3. Test Server with application installed, in a stable and ready to use state.

**TEST DELIVERABLES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Deliverable Name** | **Author** | **Reviewer** | |
| 1. | Test Plan | Test Lead | Project Manager (David James) / Business Analyst (Jane Doe) |  |
|  |  |  |  | |
| 2. | Functional Test Cases | Test Team | Business Analyst – Jane Doe |  |
|  |  |  |  |  |
| 3. | Logging Defects | Test Team | Programming Lead – John Smith |  |
|  |  |  |  | |
|  |  |  |  |  |
| 4. | Daily/weekly status report | Test Team/ Test Lead | Project Manager – David James |  |
|  |  |  |  |  |
| 5. | Test Closure report | Test Lead | Project Manager – David James | |

### 3.3. User Acceptance Test (UAT)

**PURPOSE**: This test allows the end users (players) to complete one final review of the system prior to deployment.

**TESTERS**: The UATs is performed by the end users i.e. players/gamers

**METHOD**: Since players/gamers/consumers are the users that are going to be using this application and will provide the most feedback about the product, it may happen that the users do some validation not contained in the scripts. Test team write the UAT test cases based on the inputs from End and Business Analyst’s.

**TIMING**: After all other levels of testing (Exploratory and Functional) are done. Product will be released to production after UAT testing complete

### 3.4 Unit Testing

#### Definition:

Specify the minimum degree of comprehensiveness desired. Identify the techniques which will be used to judge the comprehensiveness of the testing effort (for example, determining which statements have been executed at least once). Specify any additional completion criteria (for example, error frequency). The techniques to be used to trace requirements should be specified.

#### Participants:

List the names of individuals/departments who would be responsible for Unit Testing.

#### Methodology:

Describe how unit testing will be conducted, including a description of tests to be carried out. Who will write the test scripts for the unit testing, what would be the sequence of events of Unit Testing and how will the testing activity take place?

### 3.5 System and Integration Testing

#### Definition:

List what is your understanding of System and Integration Testing for your project.

#### Participants:

Who will be conducting System and Integration Testing on your project? List the individuals that will be responsible for this activity.

#### Methodology:

Describe how System & Integration testing will be conducted, including a description of tests to be carried out Who will write the test scripts for the unit testing, what would be sequence of events of System & Integration Testing, and how will the testing activity take place?

### 3.6 Performance and Stress Testing

#### Definition:

List what is your understanding of Stress Testing for your project.

#### Participants:

Who will be conducting Stress Testing on your project? List the individuals that will be responsible for this activity.

#### Methodology:

Describe how Performance & Stress testing will be conducted, including a description of tests to be carried out Who will write the test scripts for the testing, what would be sequence of events of Performance & Stress Testing, and how will the testing activity take place?

## 4.0 TESTING STRATEGY

### 4.1. Entry and Exit Criteria

* The entry criteria refer to the desirable conditions in order to start test execution; there must be a stable build of the game and all game-breaking defects removed.
* The exit criteria are the desirable conditions that need to be met for the testing to be completed and to proceed with the implementation of the code.
* Entry and exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk and provide a recommendation.
* Entry criteria to start the execution phase of the test: the activities listed in the Test Planning section of the schedule are 100% completed.
* Entry criteria to start each cycle: the activities listed in the Test Execution section of the schedule are 100% completed at each cycle.

### 4.2. Test Cycles

1. There will be two cycles for functional testing. Each cycle will execute all the scripts.
2. The objective of the first cycle is to identify any blocking, critical defects, and most of the

high defects. It is expected to use some work-around in order to get to all the scripts.

* 1. The objective of the second cycle is to identify remaining high and medium defects, remove the work-around from the first cycle, correct gaps in the scripts and obtain performance results.
* User Acceptance Test (UAT) will consist of one cycle.

### 

### 4.3. Test Metrics

A metric is something used to measure in data, and Test metrics are used to measure the progress and level of success of the test developed and is submitted to the project manager for their approval. Some of the metrics for the tests are:

|  |  |  |
| --- | --- | --- |
| **Report** | **Description** | **Frequency** |
| Test Execution Status & Test Preparation | %Pass & %Fail reports on % of tests completed | Weekly/ Daily |
| Daily Status | Pass/Fail/Game-breaking defects | Daily |
| Weekly Status Report | Project reports | Usually Weekly |

## 5.0 TEST SCHEDULE

Include test milestones identified in the Software Project Schedule as well as all item transmittal events.

Define any additional test milestones needed. Estimate the time required to do each testing task. Specify the schedule for each testing task and test milestone. For each testing resource (that is, facilities, tools, and staff), specify its periods of use.

### 6.0 Validation and Defect Management

### 6.1 Problem Reporting

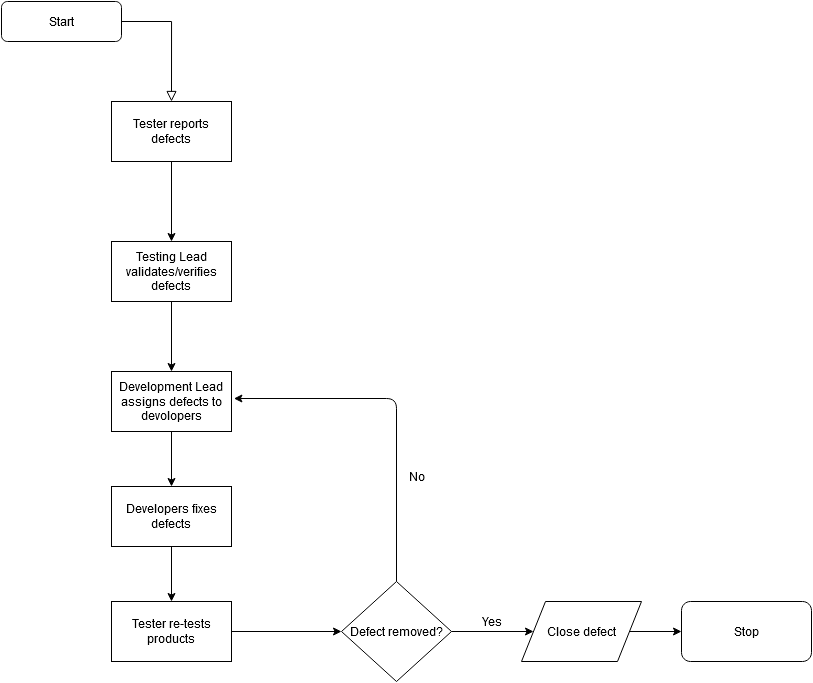
* It is expected that the testers execute all the scripts in each of the cycles described above. However, it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. This is especially relevant in the second cycle, when the Business analyst’s join the Testing Centre of Excellence (TCOE) in the execution of the test, since the Business Analysts have a deeper knowledge of the marketing and business behind developing a video game for commercial use.
* The defects will be tracked through GitHub Issues Tracker only. The technical team will gather information daily from GitHub and request additional details from the Defect Coordinator, Charles Foxe. The technical team will work on fixes.
* It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Defect Manager to review the severity of the defects and facilitate with the technical team the fix and its implementation, communicate with testers when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the technical team to review HP ALM on a daily basis, ask for details if necessary, fix the defect, communicate to the Defect Manager the fix is done, implement the solution per the Defect Manager request.

During testing, defects/bugs/glitches will be logged according to their impact on the overall game, as detailed below:

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| Critical | This bug is critical enough to crash the system, cause file corruption, or cause potential data loss.  It causes an abnormal return to the operating system (crash or a system failure message appears).  It causes the application to hang and requires re-booting the system. |
| High | It causes a lack of vital program functionality with workaround. |
| Medium | This Bug will degrade the quality of the System i.e. the game’s frame rate will drop below an acceptable limit (will decrease player usability and enjoyment).  This bug prevents other areas of the product from being tested.  However other areas may be tested independently. |
| Low | There is an insufficient or unclear error message, which has minimum impact on product use |
| Cosmetic | There is an insufficient or unclear error message that has no impact on product use. |

### 6.2 Defect tracking & Reporting

The Defect Tracking Process:



## 7.0 RESOURCES/ROLES & RESPONSIBILITIES

Specify the staff members who are involved in the test project and what their roles are going to be (for example, Mary Brown (User) compile Test Cases for Acceptance Testing). Identify groups responsible for managing, designing, preparing, executing, and resolving the test activities as well as related issues. Also identify groups responsible for providing the test environment. These groups may include developers, testers, operations staff, testing services, etc.

## 8.0 SCHEDULES

Identify the deliverable documents. You can list the following documents:

- Test Plan

- Test Cases

- Test Incident Reports

- Test Summary Reports

## 9.0 RISKS/ASSUMPTIONS

Identify the high-risk assumptions of the test plan. Specify contingency plans for each (for example, delay in delivery of test items might require increased night shift scheduling to meet the delivery date).

## 10.0 TOOLS

List the Automation tools you are going to use. List also the Bug tracking tool here.