Software Testing S6 project 2020

Game Development International Ltd

Test Plan

BSc in Computing in Software Development

Grace Keane

G00359990

Appendix 3

Test plan document

Test Plan Template - Pixel Wizard

Prepared by - Grace Keane, John Walsh, Sam Kelly, Olivia Simpson, Jane Murphy.

05 / 05 / 2020

Table of Contents

1.0 INTRODUCTION

This test plan describes the testing approach and overall framework that will drive the testing of the "Pixel Wizard" 2D application.

The customer wants a perfect game, which passes the full cycle of manual testing. Given the specificity of the application it is very important to have the same level of quality as portrayed in the appendix 2 documentation.

The test plan has been created to facilitate communication within the team members. It describes approaches and methodology that will apply to the unit, integration, system testing and other software testing methodologies of the game design document. It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones and approach. This document has clearly identified what the test deliverables will be and what is deemed in and out of scope. The end products should be a fully functioning, high quality Pixel Wizard application.

Brief overview of the application itself

Pixel Wizard is a 2-Dimensional horizontal scrolling shooter game called. The main theme is set in a forest. The background is used repeatably to make the level seem longer. On the top left and right of game play there is health representations for both player and enemy. These are displayed by red and blue diamonds. There are three characters to this game, they are player, enemy and boss. These characters are programmed to attack. The main theme of this game is "kill or be killed". If the player is successful, they will progress to other levels and soon win the game overall. Unfortunately is the player is killed it will either be game over or they will have an opportunity to increase their health using a health pickup which is located on a platform in game play.

The game has various requirements which will allow this application user friendly, efficient and professional. The game has a start-up main menu which contains buttons for settings, load game, delete game, exit game and a play game button which takes the player from the title screen to the point that gameplay begins.

An in-game pause menu is also included. Once the game has been paused using the space bar, the player can access settings or exit the game, a save game option will also be presented, allowing the player to save the game state.

The way in which the player controls the game entities is relatively easy which will cause no confusion to the player. Control is different for both laptop and mobile phone.

2.0 OBJECTIVES & TASKS

2.1 Objectives

1) Ensuring that the software under test is bug free before release.
2) Testing is to be started at an early stage to leave time to deal with critical and major bugs.
3) Gaining confidence in and providing information about the level of quality.
4) To make sure that the result meets the business and user requirements.
5) To ensure that the application satisfies the client.
6) To gain the confidence of the customers by providing them a high-quality product.
7) Find as many software defects as possible.
8) Clearly define tasks and responsibilities.
9) Tasks will be evenly distributed between testing members.
10) Result should be a production-ready software.

Document to be used

Appendix 2 – Game Design Document

2.2 General Tasks

- Ideally one test must be carried out at a time to avoid tracking errors or bugs arising.
- Documentation should be real and fully understood by each member of the test team.
- Small, clear and non-complex test cases should be written.
- Exit criteria and test closure must be fully thought out.
- Testing must be carried out at an early stage to leave time for possible critical errors.
- Different independent test cases must be carried out to avoid pesticide paradox.
- Work must be evenly divided amongst testers.
- If problems arise, they must be recorded and solved quickly.
- Main components of the game must be tested individually and then tested again using system testing.
- Post testing must be fully done when application is complete.
- All in scope material must be fully tested by the test team.
- Regression testing must be carried out with every addition of code.

3.0 SCOPE

3.1 General

The concept of test plan is to outline what needs to be tested and what tests should be used for each individual and combined component of Pixel Wizard. Software testing is an important procedure to evaluate the functionality of a software application with an intent to find whether the developed software meets the desired recommendations and quality standards. Testing also should identify defects in the program as time progresses, these errors will be documented and fixed to ensure that the finished product is defect-free, of relatively high standard and corresponds to deliverables.

All components listed in the game design document should be thoroughly tested using numerous testing approaches such as unit testing, integration testing, system testing, performance & stress testing, user acceptance testing, batch testing, automated regression testing and beta testing.

Testing jobs and test script writing are evenly shared between Grace Keane, Jane Murphy, Olivia Simpson, John Walsh, and Sam Kelly.

3.2 Tactics

- Exploratory testing would be carried out once the build is ready for testing.
- All defects seen should come along with a snapshot JPEG format and they should be fully recorded.
- The test team assumes all necessary inputs required during test design and execution will be supported by development/ business analysts appropriately.
- Test case design activities will be performed by the QA group.
- Any defect fixes planned will be shared with test team prior to applying fixes to application.
- Project manager will review and sign-off on test deliverables.
- Project team has the required experience and knowledge.
- Test team will manage test planning, test design and test execution support.
- Be familiar with game rules and test the gameplay against these rules.

4.0 TESTING STRATEGY

4.1 Unit testing

Description

Unit testing is a level of software testing where individual units/ components of a software is tested. The purpose is to validate how each single unit of the software performs. A unit is the smallest testable part of any software material. It usually has one or a few inputs but usually only one output.

Participants

Grace Keane

John Walsh

Sam Kelly

Popular unit testing framework

One of the most popular C# unit testing frameworks is NUnit. It is an open-source unit testing framework for the .NET framework and Mono. It serves the exact same purpose as Junit does in the java world.

NUnit features

- Tests can be run from a console runner, within a visual studio app through a test adapter, or 3rd party runners.
- Tests can be run in parallel.
- NUnit has a strong support for data driven tests.
- It supports multiple platforms including .NET Core, Xamarin Mobile, Silverlight or Compact framework.
- Test cases can be added to one of more categories, this allows selective running.

Methodology

How unit test will be carried out

Typically, unit testing is utilized to ensure that the underlying logic does not change with refactoring and other changes such as new featured. Throughout development of this Pixel Wizard all added functionality must be tested in small units to avoid bugs from appearing. Testing can be done using NUnit and Visual Studio Code (VSC). Unit testing is going to test the quality of the code. It should test the main code functions defined in scope such as page navigation, button functions, game controls, attack mechanism, movement mechanism, Health decrease, collision, score, volume control, pause functionality etc. Unit testing should be carried out until 100% decision coverage is met.

Who will write test scripts?

Script writer – Grace Keane

Testers – John Walsh, Sam Kelly

4.2 System & integration testing

System testing

System testing is testing conducted on a complete fully fledged system to evaluate the system's compliance with its specified requirements. System testing is one of the last tests to be carried out on a software project/ application.

Integration testing

Integration testing tests interfaces between components and interactions to different parts of a software systems. It occurs after unit testing and before validation testing.

Participants:

Jane Murphy

Olivia Simpson

Grace Keane

Methodology

System testing of software is testing conducted on a complete and integrated system using the game platform to evaluate the system's compliance with its specified requirements. It is the one of the last tests to be carried out on this Pixel wizard application. System testing includes functional & non-functional testing and is performed by the testers.

System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product. System testing can be performed on the game platform when the whole application has been put together. Functional testing considers the specific behaviour of the software application. Non-functional testing tests the quality characteristics such as how well or how fast something is done or how well the system works.

Integration testing should be carried out after unit testing. Once all the individual units are created and tested, them components should then be combined to performing the integrated testing. The main goal here is to test the interfaces between the units/modules.

For integration testing the team must use Bottom-up testing. This is an approach to integrate testing where the lowest level components are tested first, then higher-level components. The process is repeated until the components at the top of the hierarchy are tested.

Examples of low-level components to test

- Page navigation
- Button functionalities
- Exit game
- Volume control settings
- Game movement, crouch, jump and shooting
- Play game button

Examples of High-level components to test

- Load, exit, resume, restart, save delete game
- Collision detection
- Health mechanism
- Die mechanism
- Score system

How can Integration testing be carried out?

1) Prepare the test integration plan
2) Decide on the type of integration testing approach
3) Design test cases, test scenarios and test scripts accordingly
4) Deploy the chosen modules together and get the integration tests running
5) Track the defects and record the test results of tests
6) Repeat the above steps until the complete system is tested

4.3 Performance and stress testing

Performance testing

Performance testing is the process of determining the speed, responsiveness, and stability of an application under a workload.

Stress testing

Stress testing is the software testing approach for exercising a software system beyond its maximum design load. Stress testing causes defects to come to light faster. Stress testing can be done by continuously over running the Pixel Wizard through the game platform.

Participants

Sam Kelly

Olivia Simpson

Methodology

For this Pixel Wizard application, performance testing can be used to test whether the game can be played at a certain speed, page refreshes are relatively fast and shoot speeds are as expected throughout the game. These can be tested by manually playing the game and recording speed data.

Steps when stress testing

Stress Testing process can be done in 5 major steps:

1. Planning the Stress Test. Here you gather the system data, analyse the system, define the stress test goals.
2. Create Automation Scripts: The Stress testing automation scripts should be designed and implemented as well as generating the test data for the stress scenarios.
3. Script Execution: In this stage, the Stress testing automation scripts should be run.
4. Results Analysis: In this stage, you analyse the stress test results and identify bottlenecks.
5. Tweaking and Optimization: Lastly the system should be fine-tuned, configurations changed, optimize the code with goals that meet the desired benchmark.

Stress testing can be done on separate units of the application or on the completed finished software.

4.4 User acceptance testing

Definition

User acceptance testing focus on functionality validating fitness for purpose. It is Performed by users and application managers. User acceptance testing Links tightly to system test and may overlap in time E.g. performance - load time may be an issue, usability of product.

Participants

John Walsh

Jane Murphy

Methodology

UAT is done by the intended users of the system or software. This type of Software Testing usually takes an Alpha or Beta testing approach.

For this application Pixel Wizard is going to take an alpha testing approach. This procedure is conducted at the developer's site in a controlled environment. The developer monitors the user to observe and note problems.

User acceptance testing is one of the last tests to be done on a software product. It is carried out when the full application is complete. It fully tests the main components of the software product.

4.5 Batch testing

A Group of tests executing sequentially one by one is called Batch Testing. All automated test scripts are executed one at a time by keeping the other scripts in waiting mode. The end state of one test is base normally the start state to another test. Batch testing should be constantly monitored and testing throughout the development of Pixel Wizard.

Participants

Sam Kelly

Jane Murphy

4.6 Automated regression testing

Definition

Regression testing is the selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still works as specified in the requirements. Regression testing is useful for detecting bugs at the beginning stages of product development.

Participants

Grace Keane

Jane Murphy

Olivia Simpson

Methodology

regression tests must be incorporated in the daily test cycle. After every change to code or the system, regression testing should take place. Testers should use a bug tracking tool to monitor errors occurring from the addition of new data. This ensures that the addition of extra features has not corrupted the functionality of the original game.

4.7 Beta testing

Definition

A beta test is a type of testing that is conducted prior to any sort of official release. It is considered the last stage of software testing. Beta testing involves releasing the application to sites or individual users outside Game Development International Ltd.

Methodology

Beta testing will be conducted using cross-platform users. The users send records of errors to the development organization where defects will be analyzed and repaired. Developers will not be present when beta testing has begun therefore it is not held in a controlled environment.

What beta testing should highlight

1. Application does what it is intended to do
2. Reaction speed during game play and in menus should be relatively fast
3. Navigation buttons should work as expected
4. Attach mechanisms should be fast and non-glitchy
5. Movement should be quick and smooth
6. Health and score increase and decrease efficiently
7. There should be no delay anywhere in the application
8. Level progression should be smooth and efficient

Participants

John Walsh, Sam Kelly, Grace Keane.

5.0 TESTING SCHEDULE

Test procedure	Test Rank	Test duration
Unit testing	1	Continuous until main components of application are complete.
Automated regression testing	2	Continuous until main components of application are complete.
Integration testing	3	5 days
Batch testing	4	5 days
System testing	5	5 days
Performance & stress testing	6	1 week
User acceptance testing	7	1 week
Beta testing	8	2 weeks

6.0 CONTROL PROCEDURES

Problem reporting

A problem report documents problem that occur when testing a software application. When testing this “Pixel Wizard” application errors must be fully documented immediately when found by using the sample bug report below. After documented, problems encountered must then be analyzed, fully fixed and additional regression tests must be carried out to ensure a high-quality product is constructed.

Sample bug report

Bug Name: Name of bug
Bug ID: Bug ID
Area Path: Path
Bug description: Brief description of bug encountered.
Build Number: Number
Severity: (High/Medium/Low)
Priority: HIGH (High/Medium/Low)
Assigned to: Developer
Reported By: Your Name
Reported On: Report date
Reason: Defect, bug, missing data/ functions
Status: New/Open/Active
Environment: Windows, SQL etc..

Change requests

A change request is an important document which is part of the change management process, as it states the data and reasons for the change in the Pixel Wizard application. Below are some key components to focus on and document when a change might be made. Once changes are documented they must be Explained to the development team using and inspection review process.

Project name: Pixel Wizard
Request number: Request number
Requestor name: Who is requesting this change?
Description of change: Brief description of change.
Reason: Reason for change?
Impact: What impact will this have on Pixel Wizard
Actions: What actions should be taken to deal with this change
Business priority: What is the business priority of this change?
Change status: Complete, Started, Ended, Needs more work etc.

7.0 FEATURES TO BE TESTED

Fully test opening main menu components and check if all buttons work as expected.

- Play game - should take the player into the game and the player should begin at Level 1.
- Settings - should navigate to another page which will allow the player to edit game settings. Game settings like sound and music volume will be tested also.
- Load game - this should allow the player to choose which level to load. Test can the user select a game and note does to game navigate to the correct one selected.
- Delete game - should allow players to delete a games history.
- Exit game – should quit the application.
- Take note of efficiency and reaction speed when carrying out tests.

Fully test the in-game menu to check if it is working correctly with no bugs.

- Once in game and user clicks the required key (Laptop – spacebar, Mobile – button on top right of screen). Program should navigate to pause screen.

- Save game – should save position and status of current game being played.
- Settings – should navigate to the settings page. Also allows for change of sound and music volume.
- Exit game – allows users to exit the game being played.
- Resume game –The player can resume the game by selecting the appropriate option or simply pressing the assigned button for pausing/ resuming the game. When key clicked it should enable the game to be played again at current position and status.
- Restart level – should reset the entire level.
- Take note of efficiency and reaction speed when carrying out tests.

Fully test in-game components, functionality, and design.

- Test player navigation with correct key specified in document.
- Test crouch defence mechanism.
- Test jump mechanism.
- Test if player can shoot using key specified.
- Note if enemy can shoot at player.
- If player gets hit by an enemy's bullet check if health score decreases.
- If enemy gets hit by the players bullet check if health score decreases.
- hit Note can a health increase be accessed by player touching the "+" symbol.
- Test whether player can get hit and eventually die.
- Test whether enemy can get hit and eventually die.
- Test whether player can move onto next level when level one has been completed successfully.
- Test for clipping (two or more objects overlapping or cancelling each other out).
- Test for incorrect and inappropriate collision.
- Move the character through all the available objects and levels and closely examine the effect.

- Take note of efficiency and reaction speed when carrying out tests.

8.0 RESOURCES, ROLES & RESPONSIBILITIES

Roles & Responsibilities

Grace Keane

1. Script writer for unit testing
2. Test scripts implementation & design for system & integration testing
3. Regression tester
4. Implementation of Test scripts for beta testing

Jane Murphy

1. System tester
2. User acceptance tester
3. Regression tester
4. Script writer for batch testing

Olivia Simpson

1. Integration tester
2. Write test script for performance & stress testing
3. Test script implementation for regression testing

John Walsh

1. Unit tester
2. Write test scripts for user acceptance testing
3. Beta tester

Sam Kelly

1. Unit testing - tester
2. Performance & stress testing - Stress testing
3. Beta testing
4. Batch tester

9.0 SCHEDULES

Test Deliverable	Date	Time
Test Cases	01/05/2020	1:00 pm
Test scripts	02/05/2020	12:00 pm
Test Plan	10/05/2020	12:00 pm
Test Change Reports	15/05/2020	3:00 pm
Test Incident Reports	29/05/2020	12:00 pm
Test Summary Reports	05/06/2020	12:00 pm

10.0 RISK ASSUMPTIONS

If by valid reasons the team does not have enough time to test Pixel Wizard, more people might have to be added to the testing team or move hours must be put in to ensure all tests have been carried out. Main thing is to fully test Pixel Wizard and ensure the product is of high quality and fit for purchase.

11.0 TOOLS

1. Game platform
2. NUnit
3. Visual Studio Code (VSC)
4. Test adaptor
5. Bug tracking system