NAME OF THE PRODUCT

Prepared By:

Fabricio Gentile

01/05/2020

TABLE OF CONTENTS

1. Introduction

The product is a pixeled 2D side scrolling platform game. At first divided into three levels with a different range of enemies and bosses. In order to kill the enemies, the player can attack shooting a ball of flame from using the left mouse click. In the game scene, the player’s health bar and boss’s health bar will be represented by diamonds (red for the player and blue for the boss), the background will be used repeatedly, the player will be presented with some health pickups to increase his health

The player will be presented with a main menu which he can start the game, change the

The game will have three options on start-up: ‘Play’, ‘Settings’, and ‘Exit Game’.

Selecting ‘Play’ will take the player into the game and the player will begin at Level 1. If a save system is able to be implemented, the player will begin at their last saved point. ‘Settings’ will allow the player to edit game settings, such as sound level and music level. ‘Exit Game’ will quit the application’. OPTIONAL: Include a ‘Load Save’, ‘Save Game’, and ‘Delete Save’ option. This will allow the player to create multiple save files and be able to choose which one to load.

2.0

2.1. Test Objectives

The objective of testing is to assure that the system meets the full requirements, fit metrics for each quality requirement and satisfies the use case scenarios and maintain the quality of the product. The software test will ensure that our software is free from high and medium severity defects and all the game logic will function as intended. For example:

After killing the boss, the player should go to the next level.

Picking up the “health pickup”, the health of the player should increase

3.0 SCOPE

* 3.1 Functions to be tested. he menu and the menu functions
* art (character model, platform, texture, objects, enemies, etc.)
* animation (when the player/ enemy is attacking)
* music
* audio and the sound effect
* title screens
* game logic and flow
* world/scene/level
* increasing levels of difficulty
* the scoring
* the game options (game start or menu selection, game pause, pause menu options)
* health bar and health pickup

3.2

The

4.0

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

4.1 Test Automation

Automated unit tests are part of the development process, and UI smoke-tests from CHL01 must be also automated during which performance data must be captured

The testing for the Reassigned Sales project will consist of Unit, System/Integration (combined) and Acceptance test levels. It is hoped that there will be at least one full time independent test person for system/integration testing. However, with the budget constraints and time line established; most testing will be done by the test manager with the development teams participation.

UNIT Testing will be done by the developer and will be approved by the development team leader. Proof of unit testing (test case list, sample output, data printouts, defect information) must be provided by the programmer to the team leader before unit testing will be accepted and passed on to the test person. All unit test information will also be provided to the test person.

SYSTEM/INTEGRATION Testing will be performed by the test manager and development team leader with assistance from the individual developers as required. No specific test tools are available for this project. Programs will enter into System/Integration test after all critical defects have been corrected. A program may have up to two Major defects as long as they do not impede testing of the program (I.E. there is a work around for the error).

ACCEPTANCE Testing will be performed by the actual end users with the assistance of the test manager and development team leader. The acceptance test will be done in parallel with the existing manual ZIP/FAX process for a period of one month after completion of the System/Integration test process.

Programs will enter into Acceptance test after all critical and major defects have been corrected. A program may have one major defect as long as it does not impede testing of the program (I.E. there is a work around for the error). Prior to final completion of acceptance testing all open critical and major defects MUST be corrected and verified by the Customer test representative.

A limited number of distributors will participate in the initial acceptance test process. Once acceptance test is complete, distributors will be added as their ability to generate the required EDI data is verified and checked against their FAX/ZIP data. As such, some distributors will be in actual production and some in parallel testing at the same time. This will require careful coordination of the control tables for the production system to avoid posting test data into the system.

4.1 Unit Testing

**Definition:**

Unit Test is basically the test of the smallest testable part of the code. If one part of the game is buggy/broken, it might result in other areas of the game to fail.

**Participants:**

John; Terry; Bobby.

**Methodology:**

It will be done by the developer and will be approved by the development team leader. Proof of unit testing (test case list, sample output) must be provided by the programmer to the team leader before unit testing will be accepted and passed on to the test person.

4.2 System and Integration Testing

**Definition:**  
System and Integration testing aim to find integration failures between the units, it consists of carrying out various types of tests that aim to determine whether the components of a computer system (involving other software and / or hardware components) integrate well and perform the functionalities that have been specified to them.

**Participants:**

Monica; Tim.

**Methodology:**

It will be performed by the development team leader with assistance from the individual developers as required. Programs will enter into System/Integration test after all critical defects have been corrected.

4.3 Performance and Stress Testing

Definition:

*Performance and Stress testing is a form of deliberately intense or thorough testing used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results.*

*Participants: Patrick; Jordan.*

*Methodology:*

4.4 User Acceptance Testing

Definition:

User Acceptance testing is to conform that system is developed according to the specified user requirements and is ready for operational use.

Participants:

Ruby; Tessa.

Methodology:

Testing will be performed by the actual end users with the assistance of the test manager and development team leader. The acceptance test will be done after completion of the System and Integration test process.

4.5

4.6 Automated Regression Testing

Definition:

Regression testing is a software testing technique that consists of applying the most recent versions of the software, to ensure that no new defects have appeared in components already analysed.

Participants:

John; Bobby.

Methodology:

*4.7*

Beta Testing

Beta testing is a product in its beta version is for users to use it for the purpose of reporting bugs and feedback to the developers and the company that develops it.

Participants:

Trevor

Methodology:

The beta test is implemented by users, usually with little or no management by the development organization (or another non-end user). Beta testing is the most subjective of all acceptance testing strategies.

5.0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task Name | Start | Finish | Team | Comments |
| Test Planning | 05/05 | 07/05 |  |  |
| Review Requirements documents | 05/05 | 06/05 |  |  |
| Create test basis |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

6.0

See Appendix 1 and Appendix 2

7.0

**Features To Be Tested**

|  |  |
| --- | --- |
| **Music/**  **Sounds** | ON/OFF sound & background music |
| Verify if sound effects are in sync with action |
| **User Interface** | Check for animation, movement of character, graphics etc |
| Test whether one object overlaps with another |
| Verify if loading indicator is displayed wherever required |
| Character should not move out of the screen/specified area |
| Check for screen title |
| Font displayed (color, size etc) |
| Check other objects (platform, pickup items, tree on background, etc) |
| **Performance** | Check the loading time of a game (starting the game and changing levels) |
| **Score** | score calculation |
| Check for level completion syncs with the score |
| **Multitasking** | Switch between different apps and play game, check for sound, score, UI |
| **Pause** | Check if game is paused when multitasking or spacebar is pressed |
| **Functionality** | Check game area, game logic |
| play till last level |
| Check for bonus score |
| Check the score hike when level gets increased |
| Menu options |

8.0

**Features Not To Be Tested**

|  |  |
| --- | --- |
| Check Share options | Post score via mail/FB/Twitter |

Not to be included in this release of the Software.

9.0

|  |  |  |
| --- | --- | --- |
| Role | Staff Member | Responsabilities |
| Project Manager | Joe | In charge of recruiting, staff supervision, and staff training.  Responsible for test budgeting and test planning and the cohesive integration of test and development activities.  Coordinate meetings and keep track of the progress of the testing as well as ensuring that test-product documentation is complete. |
| Lead Designer / Team | Smith | Develop all testing scenarios and procedures.  In charge of training new testers, the procedures for bug and status reporting.  Be able to identify the best ways to leverage a test tool on the project and to review test reports. |
| Designers | Patrick  Jordan  Carol  Pheobe | Execute automated test cases using test scripts designed by the programmers as well as manual tests.  Prepare test reports which will be reviewed by the Lead Designer. |
| Lead Programmer | Tim | In charge of the technical aspect of leadership for the testing.  Be able to verify the quality of the requirements, including testability, requirement definition, test design, test-script and test-data development, test automation, test-environment configuration; test-script configuration management, and test execution.  The Lead Programmer will help train new testers to use existing test tools. |
| Programmers | John  Terry  Monica  Bobby | Maintaining test environment and creating automated scripts.  Responsible for executing security, load and performance stress test.  Preparing test reports which will be reviewed by the Lead Programmer. |
| Public Testers | Ruby  Tessa  Trevor | Public Testers will be introduced to the project to provide external feedback from the consumers and usability testing. |

10.0

|  |  |  |
| --- | --- | --- |
| Deliverable Documents | Start | Finish |
| Test Plan | 05/05 | 07/05 |
| Test Cases | 08/05 | 20/05 |
| Test Incident Reports | 20/05 | 21/05 |
| Test Summary Reports | 20/05 | 21/05 |
|  |  |  |
|  |  |  |

11.0

*Risk and Mitigation*

|  |  |  |
| --- | --- | --- |
| Task Delay | High | - Pinpoint problem origin and solve it immediately  - Require tester to sign task completion report form |
| Unexpected absence of staff | Medium | - Enlist a temporary substitute employee through specialist or freelance.  - If unable, director are to temporarily fill in the work.  - All staff are to weekly document their work in preparation for their substitutes. |
| Insufficient tester | Low | - Project leads are to temporarily fill in the work  - Hire 3rd party tester or freelance |
| Insufficient skilled tester | Low | - Make sure testers go through tester training briefing |

Appendix 1

PROBLEM REPORT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Name** | **Issue** | **Responsible** | **Date** | **Action Taken** | **Status** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Appendix 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Module** | **Request** | **How is now** | **How it will be** | **Reasons for change** | **Positive Impact** | **Negative Impact** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |