**Introduction**

For this assignment I have decided to outline the details of the test plan that a group of testers brought in by myself will test the different features of the game Pixel Wizard to see if the features work perfectly or if the are errors or bugs in the game that need to be fixed.In each testing department there will be different roles carried out for each tester in each department as some people specialise in other fields better then others as they would have more experience in certain areas.The concept of this game is that the player has to overcome it’s enemies and move on to the next level.

**Objectives And Tasks**

**Objectives**

* Check that whether the game Pixel Wizard **functionality** is working as expected without any error or bugs.
* Check that the external interface of the website such as **UI** is working as expected and meet the users need.
* Verify the **usability** of the website. Are those functionalities convenient for user or not?

**Tasks**

Plan Test

|  |
| --- |
| * Identify Requirements for Test |
| * Assess Risk |
| * Develop Test Strategy |
| * Identify Test Resources |
| * Create Schedule |
| * Generate Test Plan |

Design Test

|  |
| --- |
| * Workload Analysis |
| * Develop Test Suite |
| * Identify and Describe Test Cases |
| * Identify and Structure Test Scripts |
| * Review and Access Test Coverage |

Implement Test

|  |
| --- |
| * Setup Test Environment |
| * Record or Program Test Scripts |
| * Develop Test Stubs and Drivers |
| * Identify Test-Specific functionality in the design and implementation model |
| * Establish External Data sets |

Execute Test

|  |
| --- |
| * Execute Test Scripts |
| * Evaluate Execution of Test |
| * Recover from Halted Test |
| * Verify the results |
| * Investigate Unexpected Results |
| * Log Defects |

Evaluate Test

|  |
| --- |
| * Evaluate Test-Case Coverage |
| * Evaluate Code Coverage |
| * Analyze Defects |
| * Determine if Test Completion Criteria and Success Criteria have been achieved |
| * Create Test Evaluation Report |

**Scope**

**General**

The interfaces of the game will be tested to see if it’s working perfectly and to make sure it will catch any errors or bugs in the game.I will make sure that the testers will catch it as early as possible so that it’s easier to fix the problem.

The testers will make sure that it meets the users requirements and that it will have everything correctly set up.

Make sure that the functionality of the game is easy to use and is understandable to the player playing the game

**Tactics**

The tactics for the test plan is that we will have each tester at their own department and will work on different types of testing.

Where they are based in depends on where is there field of expertise in regards to testing.

Then they will test different parts of the game to see if it’s all working or not

**Testing Strategy**

These are all of the testing strategies that I have placed for the testers to test for this game

**Unit Testing**

Definition: A unit test is a way of testing a unit - the smallest piece of code that can be logically isolated in a system. In most programming languages, that is a function, a subroutine, a method or property.

Participants:Devlopers will participate in Unit Testing

Methodology: Tests will run periodically, often after every change to the source code. The more often the better, because the sooner you will catch problems.

**System and Integration Testing**

Definition:System Integration Testing is defined as a type of software testing carried out in an integrated hardware and software environment to verify the behaviour of the complete system.

Participants: Developers and testers will participate in this.

Methodology: Confirmation tests at this level will identify environment-specific problems, such as errors in memory allocation and de-allocation. The practicality of conducting software integration in the host environment will depend on how much target specific functionality is there. For some embedded systems the coupling with the target environment will be very strong, making it impractical to conduct software integration in the host environment.

**Performance and Stress Testing**

Definition:Performance testing is carried out to check the system's performance under varying loads. Stress testing is carried out to check the behaviour of the system under the sudden increased load. It contains load and stress testing as components.

Participants: Developers will participate in this testing.

Methodology:  The goal is to find out **how the application behaves in such stress situations**to**determine its robustness.**

**User Acceptance Testing**

Definition: The purpose of acceptance test is to confirm that the system is ready for operational use. During acceptance test, end-users (customers) of the system compare the system to its initial requirements.

Participants: Leo Varadkar,

Michael D Higgins,

Simon Coveney

Methodology: The main purpose of this testing is to validate the software against the business requirements. This validation is carried out by the end-users who are familiar with the business requirements.

**Batch Testing**

Definition:Batch testing is performed by running the entire test set. All automated test scripts are executed one at a time by keeping the other scripts in waiting mode.

Participants: Testers will participate in this.

Methodology: Every test Batch consists of multiple dependent test cases. The result of one Script failure or pass, fails or passes the whole batch test.

**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.

Participants: Testers will participate in this.

Methodology: Develop test cases and prioritize testing activities. Execute all the test case and report defects, define severity and priority for each defect.

**Beta Testing Participants**

Methodology: Beta testing is an opportunity for real users to use a product in a production environment, with the goal of uncovering any bugs or issues so they can be addressed before a general release.

**Test Schedule**

* Test Plan to be done within 4 weeks
* Test Cases to be done within 3 months after the test plan is finished first
* Test Incident Reports to be done a month after the test cases are finished
* Test Summary Reports to be done a month after the test incidents reports are finished

**Control Procedures**

**Problem Reporting**

This will be to report all the problems the tester/developer will encounter while performing tests on the game .It will identify all the bugs and errors that was in the system that eventually had to be fixed.

**Change Requests**

In order to request changes in terms of changing tests or changing something in the system, these requests will have to go through the head developer or the head tester in order for it to be granted their wish.

**Features To Be Tested**

* Functionality
* Gameplay
* Load Game
* Delete Game
* Exit Game

**Features Not To Be Tested**

* Instructions Page

**Resources/Roles & Responsibilities**

* Head Developer: John Smith
* Head Tester: Jason Green
* Developers: Mary Walsh, Peter Folan, Patrick Joyce
* Testers: Edward Conroy, Raymond Dooley, Niall Rooney, Peadar Monaghan

**Schedules**

**Test Plan:**1st February – 29th February 2020

**Test Cases:**1st March – 1st June 2020

**Test Incident Reports:**2nd June – 2nd July 2020

**Test Summary Reports:**3rd July – 3rd August 2020

**Risks/Assumptions**

The main risk in this process is either not finding any bugs/errors early or not completing the objectives in time.

**Tools**

* Bug Tracking Tool
* UFT One
* TestComplete
* **SoapUI**
* IBM Rational Functional Tester (RFT)
* Tricentis Tosca
* Ranorex.