**Test Plan Template: The Pixel Wizard**

**Prepared by Daniel Gallagher – 8/5/2020**

**TABLE OF CONTENTS**

1.0 - INTRODUCTION

2.0 - OBJECTIVES AND TASKS

2.1 - Objectives

2.2 - Tasks

3.0 - SCOPE

4.0 - Testing Strategy

4.1 - Unit Testing

4.2 - System and Integration Testing

4.3 - Performance and Stress Testing

4.4 - User Acceptance Testing

4.5 - Batch Testing

4.6 - Automated Regression Testing

4.7 - Beta Testing

5.0 - Test Schedule

6.0 - Control Procedures

7.0 - Features to Be Tested

8.0 - Features Not to Be Tested

9.0 - Resources/Roles & Responsibilities

10.0 - Schedules

11.0 - Risks/Assumptions

12.0 – Tools

**1.0 INTRODUCTION**

The product being tested is a new game development called “The Pixel Wizard”. It is a new game which is a 2D side-scrolling platformer game. The player of the game controls a wizard character which progresses through levels which contain several enemies which they must defeat to progress. At the end of each level there is a boss character that must be defeated before the player can progress to the next level. Each level also contains pickable items such as health etc. which the player can pick up along the journey

Main game movements – The game is both playable on pc and phone which has two different control methods. To move forwards and backwards in the game it is simple with the mobile platform having arrows on the screen which you press and hold and on pc you simply use (D,A) keys for forwards and backwards or the arrow key in that direction. To crouch it is simple too with an arrow on mobile and on PC you simply hold (C) key. To attack as your character on phone you use the dedicated attack button. On PC the user simple left clicks or clicks (R). Inside the game there is also the available function of pausing the game this can be done on mobile using the button in the top right of the screen or done on PC using the spacebar.

Shooting – One key aspect of the game is shooting. The player shoots using the dedicated button on mobile or the R key or left click on pc. When the player shoots it is of vital importance that the player shoots a bullet and on colliding with the enemy the bullet kills the enemy and they die. This is a key function of our game.

Menus – Once the game starts and the main menu loads there is a lot of functionality on the home screen which are “Play Game”, “Settings”, “Load Game”, “Delete Game”, and “Exit Game”. These buttons all must work correctly. Play game loads the player into a new game if no game is played before hand to load or loads them in to where they left off. Settings should bring the user to a menu where they can fix settings such as sound and music level. Load game allows the user to load a previous game that they saved using the saving system. Delete game allows the user to delete a stored game as they wish. Exit game exits the user from the game allowing them to close the game down.

In-Game menus – The in-game menu is show when the user presses the space bar on pc or the top right-hand button when using mobile. This pauses the game but displays the pause menu. This menu contains and few options such as “Save Game”, “Settings” and “Exit Game”. Save game allows the user to save their progress up to this point where they can resume their game once they are back. Settings allows the user to change the setting such as music volume and sound volume. Finally, the exit game button closes the game for the user.

In-Game Items – In the game there are in-game items which drop. They are health packs which replenish the players health which may have been lowered fighting enemies. It is vital that these are functioning correctly otherwise they are useless to the player. When navigating the map and the player picks up one of these items their health should be fully replenished allowing them to continue in the game with more health points. If these are not functioning correctly it may cause the game to be harder for the player to complete and they may become stuck or frustrated.

**2.0 OBJECTIVES AND TASKS**

**2.1 Objectives**

In this section we will cover the testing objectives, what we need tested and who is responsible for testing what parts. The objectives for our testing are to ensure every aspect of the game works as designed by the developers for our users before publishing the finished product.

The tasks which need testing are broken down and divided among the three teams below. Each team is responsible for certain testing of aspects of the game and this is a team effort to ensure the game works as we planned. Each team contains an appointed team leader who oversees tasking his or her team with tasks and they are responsible for informing the project manager of any issues that may arise during their testing periods in order to get the fixed. A group has been created on Microsoft Teams and each person on each team has been emailed an invite link to join the group. This will be used for communications during this project and any questions are welcomed to be asked.

A weekly Microsoft teams meeting will be scheduled between the project manager and all teams. This will take place on a Monday at 2pm GMT. Teams are required to state what they have tested the week prior to the meeting and what they will test during the upcoming week to stay on top of testing. Any time issues can be sorted in these meetings and any issues that may arise during a week are welcome to be told to the team is extra help is needed.

**2.2 Tasks**

Team 1 – You are allocated to testing the opening of the game, by this I mean you need to cover the loading of the game ensuring it loads correctly when downloaded , the main menu displays including testing the functionality of the buttons contained on the splash screen when entering the game. This includes “Play Game”, “Settings”, “Load Game”, “Delete Game”, and “Exit Game” buttons contained on the homepage are working and functioning correctly for the user. Any issues encountered in the testing of these areas should be forwarded to Daniel Gallagher (Project manager) immediately in order to fix issues or bugs encountered. Team lead for team 1 is assigned to Mike O’Neill.

Team 2 – You are allocated to gameplay. This is mainly checking the game functionality is working correctly when a new game is created, and an old game is loaded. Main areas to concentrate on are player movement, enemy’s movement, health of enemies and player, health pickups, level changes including boss battles, and generally ensuring gameplay works as described on the design document including loading a previous game plays correctly when loaded. This is a key aspect of our testing as any issues found here will highly impact the user experience on our game and if we publish a game with game breaking errors it will be costly to correct. Team lead for team 2 is assigned to John Jackson and any issues encountered should be immediately passed on to Daniel Gallagher. (Project manager)

Team 3 – You are allocated to pause menus of the game and ensuring these are all working correctly along with making sure saving a game is working using these menus. This may seem like a small job but if saving our game from the pause menu isn’t working it might make the user stop playing the game if they lose all the progress they made while playing so it is essential this is working correctly as it might cause the game to become unlinked and it will cost significant time and money to regain users. Team lead 3 is assigned to Saoirse Fitzsimons and any issues encountered should be immediately passed on to Daniel Gallagher. (Project manager)

**3.0 SCOPE**

**General**

In this test plan, we will be testing most aspects of the game which I have outlined above in the breakdown of teams and testing they will be carrying out. In the introduction to this document the main functionality is listed out and we are going to test all these points to ensure this game is ready to be released to the users online for downloading.

**Tactics**

To accomplish the high level of testing for this game we have taken on we will be using three teams to separately test different parts of the game independently and they will be reporting back to their team leader who is assigned and the team leaders will voice any concerns to the project manager.

**4.0 TESTING STRATEGY**

The overall approach to testing is going to be a continuous group effort whereby groups responsible for different features will ensure they are working for this game in development. For example, team 2 assigned to the gameplay aspects are responsible for checking movement works correctly and damage all the way down to level changes and boss fights. Each team testing objectives are found in part 2.1 where each team is handed out job, they are responsible for.

**4.1 Unit Testing**

**Definition:** is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

**Participants:** All teams will be using aspects of unit testing as they will be individually testing different small parts of the game to ensure each part is working correctly

**Methodology:** All teams will be using aspects of unit testing as they will be individually testing different small parts of the game to ensure each part is working correctly. For example, team 1 who are testing the main menu are testing that small aspect of menus rather than the game. Each team has been broken down jobs to do and each team is responsible for their own testing. Therefore, all teams are using unit testing.

**4.2 - 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 behavior of the complete system. It also verifies a software system's coexistence with others and tests the interface between modules of the software application

**Participants:** All teams will be taking part in these tests.

**Methodology:** We are carrying out these tests to ensure the game is behaving as we expected and that all independently tested aspects are now working together and working as they should be. If this is not the case, then the game needs to get fixed fast before we move on.

**4.3 - Performance and Stress Testing**

**Definitions:**

**Stress Testing:** Stress testing refers to the testing of software or hardware to determine whether its performance is satisfactory under any extreme and unfavorable conditions, which may occur as a result of heavy network traffic, process loading, underclocking, overclocking and maximum requests for resource utilization.

**Performance Testing:** Performance testing is the process of determining the speed, responsiveness and stability of a computer, network, software program or device under a workload. Performance testing can involve quantitative tests done in a lab or occur in the production environment in limited scenarios.

**Participants:** All teams are involved in the performance testing of this game.

**Methodology:** If the game is too slow or parts of the game are not performing as expected they will contact the team lead who will contact the project manager. As we are not using online aspects of this game there is less testing in this category as there is no online scoring system or severs involved meaning it is solely down to a single player point of view. If all parts are performing correctly on our end the same tests should and will be carried out on a regular average mobile phone as this will give a better more accurate reading as to how well our game performs as the PC’s and phones we may be using in these tests could be higher end than most people’s devices.

**4.4 - User Acceptance Testing**

**Definition:** User Acceptance Testing (UAT), also known as beta or end-user testing, is defined as testing the software by the user or client to determine whether it can be accepted or not. This is the final testing performed once the functional, system and regression testing are completed.

**Participants:** Teams must be on standby also there will be “real people” using the game and reporting their experience and issues with the game

**Methodology:** As soon as all teams have completed their testing objectives any errors will be corrected to mend the game and once this is done and tested proving the game is ready for users to play it all teams will combine the parts to make the full game which works. Instead of an instant release we will release it to a certain number of users who have signed up to our beta testing forum online. Players will be allowed to interact with the new game and will report any bugs or problems they encounter to our support team.

**4.5 - 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:** All teams are to be involved as we run the entire test set on the game as any issues found will have to be fixed by the responsible team.

**Methodology:** The entire test set is run when all parts are tested fully by the different teams. Any issues found during this time then must be fixed by the team responsible for that areas before the release of the game.

**4.6 - Automated Regression Testing**

**Definition:** REGRESSION TESTING is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

**Participants:** The teams are working through each area of their testing plan but if any issues are found they need to first report it and secondly then fix it. These regression tests ensure we have no further broke the game.

**Methodology:** As teams work through the parts they are testing if any issues are found and fixed it should be subject to regression testing to ensure the code and game are still working as we planned. This is of high importance as we could solve one issue with the game but indivertibly create another more costly or time-consuming error which could impact our project in a highly negative way.

**4.7 - Beta Testing**

**Definition:** Beta Testing of a product is performed by "real users" of the software application in a "real environment" and can be considered as a form of external User Acceptance Testing. Beta version of the software is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation. It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to tests the product in customer's environment.

**Participants:** The teams and “real world” people are participants of this, and it is the last stage before game release to the general public.

**Methodology:** This is the last step before large release of this game to the public and is the last place for errors to be fixed before release. All teams must be on standby to try and find any errors reported regarding the areas they have tested (i.e. If a player reports a bug with the main menu team 1 is responsible for fixing this and reporting it to project manager along with getting it ready and patched for final release of the game.

**5.0 – TEST SCHEDULE**

For our testing plan we have our teams already defined which is going to help with time and doing this faster. Below is going to be a breakdown of the weeks and where we hope to be and at what stage. If any issues may arise, we can tweak this plan slightly to allow extra time. Each section includes a calculated amount of time to allow for unforeseen circumstances or other circumstances that may impact our plan and timings.

Week 1 – Starts on Monday with a group meeting at 2pm GMT. Teams meet for the first time and the team leader separates work into sections which are assigned to individuals to work on. These people then begin on testing the product under supervision of leaders and test the area they are assigned.

Week 2 – Monday meeting again at 2pm GMT where teams talk about how they are progressing with their work and present any issues they have found. Testing should finish during this week and allow the fixes to begin.

Week 3 - Monday meeting again at 2pm GMT where all issues found during testing are presented and then assigned to be fixed. If there are lots of issues 2 weeks will be assigned to fixes if not 1 week will be assigned.

Week 4 - Monday meeting again at 2pm GMT where issues and fixes are discussed. If fixes have been completed and all teams are happy beta testing can commence on the Wednesday of this week.

Week 5 - Monday meeting again at 2pm GMT where current ongoing beta testing and any issues found are discussed among the teams. Beta testing continues for this week allowing over a week of “Real People” playing the game.

Week 6 - Monday meeting again at 2pm GMT where beta testing is closed, and all issues found are addressed. 2 weeks are allocated from this point to have all issues and code working together before general release.

Week 7 - Monday meeting again at 2pm GMT where all teams continue working on issues that were reported in beta testing. If teams have completed their fixes and code is running correctly on their sections, they can start to implement the changes.

Week 8 - Monday meeting again at 2pm GMT where the whole game is remade an any issues are tested by the teams for the following few days. On Wednesday of this week teams should be done testing and either have a working game or more issues to fix. More than likely teams will be complete, and our game will be working ready for release to general public.

Week 9 - Monday meeting again at 2pm GMT where the now finished game is published on the store.

**6.0 - Control Procedures**

**Problem Reporting:** This section is going to detail the correct wat to report a problem if encountered when you are testing the game in your teams. If you happen to encounter an error, please request the team leader of your team to come to your assistance. Please write down in a word document where the error is in the game along with a brief description of what is going wrong and what way it should be functioning and screenshots of the error. This document should then be sent to the team leader who can show the project manager who will decide the course of action. Each team leader needs to keep all problem reports contained in a document to be presented to the project manager called “Problem Reporting [team number]”.

**Change Requests:** If modifications are being made to the software the team leader must be informed before anything is changed. Changes you are making need to be stated on a word document along with why they are necessary and how you are going to change it. If these changes impact other parts of the game these need to be included as if another team is working on other parts you are impacting, they need to be notified. Once you have this document completed the team lead should contact the project manager who will come and sign off these changes along with a short meeting about them. Each team leader needs to keep all changes contained in a document to be presented to the project manager called “Changes Made [team number].”

**7.0 FEATURES TO BE TESTED**

As discussed before all features of this game need to be tested before the game is released to the public. Every aspect of the game is going to be undergoing testing. In my introduction I outlined the main functionality that needs the most testing which I will outline briefly below.

For a more in-depth view of things that need to be tested please go to the top of this document and in 1.0 INTRODUCTION you will find all aspects in depth of what is being tested.

* Main game movements
* Shooting
* Menus
* In-Game Menus
* In-Game Items

**8.0 FEATURES NOT TO BE TESTED**

There are no functionality aspects of this game that we are not going to test. This game is due to be released to the public soon so every aspect is going to be tested under this plan and we will be leaving no stone unturned in our testing plan.

**9.0 – RESOURCES / ROLES & RESPONSIBILITIES**

The roles for this project are down to the team managers to assign. The teams 1-3 have been assigned by management according to people’s skills and spreading out people with expertise in every area to be in each group. As outlined above your team is responsible for certain aspects of the game which can be found in 2.2 TASKS.

The team leaders as follows and they are responsible for assigning jobs to people, time management of the testing cases, changes and overall being a point of confidence and advice for their team along with reporting issues to higher ups.

Team 1 Leader – Mike O’Neill

Team 2 Leader – John Jackson

Team 3 Leader – Saoirse Fitzsimons

**10.0 SCHEDULES**

There are a few documents that are deliverable both by team member and by team leaders. Test plan and test cases need to be submitted to management by each team at the start of this testing process. There are two key documents leaders need to submit which are “Problem Reporting [team number]” for reporting problems so they can be tracked and kept on top of and secondly “Changes Made [team number].” Which will be used to track the changes made to the game so that if anything goes wrong there is a traceable path back to where errors are made and by who and what team.

**11.0 RISKS/ASSUMPTIONS**

There are a few risks associated with this project. Firstly, if there are any delays it could lead to a late release which would not look good on the company. The schedule has been made with spare time included to try and best accommodate this and minimize the impact of time loss on the project. If any issues may arise leading to delays, please inform management immediately.

**12.0 TOOLS**

Bugzilla is a bug tracking tool which we can use for this project. It is a highly useful tool and is very easy to use in situations like these top keep on top of bugs and tracking them.

Selenium is an automation tool used for software testing which is very handy to use. It is an integrated extension on most browsers which can be downloaded for free and used. It is very easy to use and can keep test cases organized and available as you go through them.

**SCREENCAST**

Link to screencast - <https://www.youtube.com/watch?v=_IXRS0Dk0jc&t=4s>