SOFTWARE TESTING

Aaron Moran - G00356519

Table of Contents

Test Plan: The Pixel Wizard

Prepared by: Aaron Moran

1.0 Introduction

For our Software Testing project, we are required to complete a document which will cover a wide range of testing and methodologies. The goal is to write about procedures that should be taken when testing the development and completion of a 2D game coded in Unity. The goal of the document is to give the reader a deep understanding of the design decisions and the choices that were made throughout the development of the game.

I will be discussing different types of testing and important features and functionality which require specific testing. The importance of testing software is crucial in Game Development especially if it is being pushed to stores such as Steam or the Play Store. If a sufficient amount of testing has been carried out this will reduce the number of bugs or glitches to occur when the user is playing the game. This will show that the game is developed at a high standard.

2.0 Objectives and Tasks

2.1 Objectives

The objective is to develop a 2D game using Unity. The game can be either a Shooter, Platformer or a Traditional Game (Board Game). You are permitted to use the “clone and tweak” method by taking a game that already exists and tweaking it into your own design. The end goal is to complete the design and development of a game that runs smooth without bugs for a user to play.

2.2 Tasks

For the design of the game you must make sure that you have included the following functionality to achieve the basics of Game Development. A Front End is required so the player can Start the Game, Pause the Game and End the Game. These UI components should all occur outside of the gameplay and carry out the correct function from any point within the game. In Game Menus should be accessible during gameplay so the player might have the option to change the volume of the game or to Resume or Exit the game. Control Mechanisms is essential within the game so the player can move and carry out the required functions to complete the game. It would be beneficial for the player to be able to access the Controls page through a Menu UI which would add to the design features of the game. The Game should have a progress and punishment aspect to the gameplay, without this it would make the game monotonous and the user would just play it from start to finish without any difficulty so making the game increasingly more difficult as you progress is essential.

3.0 Scope

3.1 General

The general scope of the functionality which we will be testing are the interfaces that will navigate the player around the game and ensure that all functionality within the interfaces are reliable and there is no point within the game where a functionality will fail to load. We will also be testing the player movement and controls to ensure their responsiveness and consistency. Ultimately, We will be testing the player pick-ups and enemy collision detection to ensure that these two important aspects of the game will adhere to their functionality while playing the game.

3.2 Tactics

To achieve our goal of testing the games aspects we must comply to a strict deadline schedule with tasks to be carried out by the designated participants. Each member of the team has a role in achieving results from testing functionality within the game and the tools and tasks will be provided for them. Each testing method will hold its own importance towards achieving success and progress within the following testing procedure. If we can stick to the deadlines and the schedule specified below the game should be release on its expected release date and no delays will be announced so the importance of catching any defects or bugs prior to release is essential.

4.0 Testing Strategy

The techniques we are going to use to test the functionality of the game developed will be thoroughly monitored. There will be no element left untested and no bugs will remain once all the tests have been completed. The strategy we will be using is the Methodical Strategy. Each member of the testing team will be given a checklist and a set of conditions to use to test specific aspects of the game. The Methodical Strategy will be beneficial also to testing the security of our game. The Reactive Strategy is a testing method that we will use for once the game is complete and ready for consumer use, we can test to see if it will be fully optimized and can withstand a high amount of traffic.

4.1 Unit Testing

Unit Testing must be carried out efficiently and accurately as it will save a lot of time and money when the product is completed. The unit testing, we will be carrying out is Manual Unit Testing.

Participants:

1. John Doe.
2. Jane Fahy.

Methodology:

John is tasked with writing the testing scripts and Jane is tasked with carrying out the required tests for Unit Testing. Jane will not be capable of carrying out tests until John has completed his task of writing the scripts. Once the scripts have been written Jane will gather test data to use for his task.

4.2 System and Integration Testing

System integration testing is very important for the project to assure that the interfaces are carrying out their required functions. Testing these features correctly will avoid any bugs on release and allows us to become aware of any errors or mistakes that were made before the game has been shipped. It is likely that we will find many of errors when carrying out this form of testing, but it is all beneficial towards creating a successful product.

Participants:

1) Mary Kelly.
2) Aaron Moran.

Methodology:

Tests will be written to discover any errors or defects within the code. The entire program will be tested, and no feature will be ignored. The Software Integrations will be divided into as number of levels.

4.3 Performance and Stress Testing

Performance and Stress Testing are extremely important for this project. In Game Development the user must be able to have an enjoyable experience playing the game and not have to deal with any performance issues. A games performance can make or break its success, so it is vitally important that we achieve a substantial level of performance.

Participants:

1) Mark Cleary.
2) Tom Hansbury.

Methodology:

The testing can be divided into Performance and Stress Testing. If Mark is to carry out the Stress Testing which will test to see how the software will perform under a high amount of pressure on the servers. Notes are essential to be taken in case of any faults or situations that appear that the software would fail. Tom can carry out Performance testing by running the game at different video settings. Also, a good strategy would be to test the game on different hardware components so a list of benchmark settings can be provided for the consumer.

4.4 User Acceptance Testing

This is a very important phase of testing. It ensures that the product will be put forward to the Production Environment.

Participants:

1) User.
2) Client.

Methodology:

The participants will receive a list of functionalities which will be tested. Each participant should have the design, plan and execution processes. Each participant then will carry out the given tests of each functionality and record the results that they receive.

4.5 Automated Regression Testing

This is a method of testing software that uses computing tools and procedures after it has been altered or updated.

Participants:

1) Jane Fahy.
2) Tom Hansbury.

4.6 Beta Testing Participants

For Beta Testing the strategy that we will use is by contacting Gaming media icons and Twitch streamers and provide them with Beta access to the game to test and advertise the game. This way we can grow an audience for the game and monitor any community feedback which we can implement before its full release date. By doing this we can get an idea of how gamers are enjoying the game and if there are any features that need tuning.

We can also provide a random number of people who have subscribed through our website to be notified when Beta release is available to be given the chance to test the game for themselves and experience a limited amount of content. This way we are broadening our community and acquiring feedback of any issues or bugs that may have been missed during our testing phases.

5.0 Test Schedule

Below is a list of the following tasks which must be completed each milestone has a start and finish date. Some tasks cannot be started until another task has been completed. The schedule specifies each person's role within the testing period and when they should start and finish. It is very important that we stick to the deadlines as any delays will be costly and slow down the rate of production.

1) Unit Testing.
2) System and Integration Testing.
3) Automated Regression Testing.
4) User Acceptance Testing.
5) Performance and Stress Testing.
6) Beta Testing.

TASK	EMPLOYEE	START DATE	FINISH DATE
Unit Testing	John, Jane	10th January 2020	29th January 2020
System & Integration Testing	Mary, Aaron	29th January 2020	4th February 2020
Automated Regression Testing	Jane, Mark	4th February 2020	8th February 2020
User Acceptance Testing	Tom, Aaron	4th February 2020	10th February 2020
Performance and Stress Testing	Mark, Tom	10th February 2020	BETA RELEASE DATE
Beta Testing	To Be Contacted.	-	-

6.0 Features to be Tested

1) Player Movement / Controls

Player Movement is a crucial feature to be tested. If there are any bugs within the players movement it will make the game unenjoyable to play and cause a lot of frustration.

2) Save Game / Load Game / Exit Game

Another crucial feature to be tested is the save and load game states. This will ensure that a player playing the game will be able to make significant progress and be able to return to the game at a different date and not have lost all that progress. Without a save and load game option the game will have to be started and finished within one sitting which is not an enjoyable gaming experience.

3) Pick Up’s

The Pick Up’s feature must be tested as they play a significant role in the Player regaining health. If the player cannot regain health, then the game becomes harder to survive and can increase the difficulty.

4) Attacking / Shooting

Attacking and shooting must be tested as it is one of the main functions of the game. If the Player cannot shoot, then the game cannot be completed or even played.

5) Health Bar

The Health Bar must be tested as it brings a sense of punishment to the game. When the player loses all their health, they must restart the level.

6) Collision Detection

This is very important and must be tested as the player must comply to the terrain of the game. The Player should not fall through any boxes or terrain items within the game map if Collision Detection has been implemented correctly. Additional collision detection should be tested for Player bullets colliding with an enemy and Enemies colliding with the Player.

7) Enemy to Player Pathfinder

It is important to test that the enemy sprites will find the shortest path to the player. This increases the difficulty of the game. Without the pathfinder the Enemies would be static which would make the game too easy to complete.

7.0 Features not to be Tested

1) Background

It is not necessary to test the background images as they will stay constant throughout a level design.

2) Sprites

Sprites also do not need to be tested as they have been assigned as prefabs and will be reused for each level.

8.0 Resources, Roles and Responsibilities

9.0 Schedules

Below is a list of deliverable documents which must be submitted and acquired after the completion of a set of tests.

- Test Plan.
- Test Cases.
- Test Incident Reports.
- Test Summary Reports.
- Community Feedback Report.
- Improvements Report.
- Defects Report.
- Release Notes.
- Benchmarks and Configuration Report.

10.0 Risks and Assumptions

- Many issues may occur in the delivery of results from tests that might cause deadlines to be pushed back further which then will delay the date of release. Issues like this can be prevented or worked around if we can create a fair and beneficial environment for the team members.

- If we face issues where we require more time to finish a specific task, we can assign more members of the team to a task to speed up its development and testing amounts.

- Nigh Shifts may be brought in if we are running close to deadlines for tasks that must be completed for another task to begin its course.

- If we face a lot of issues in the testing phases, we can hire extra employees to help us reach our end goal. This would mean an increase in costs so this method would be a last resort in means on financing and funding.

11.0 Tools

It is very important that powerful and reliable Software Testing tools are used. This way we can assure accuracy and tracking throughout the processes.

Bug Tracking System

- ASANA (https://asana.com/?noredirect)
- Easy tracking of bugs.
- Team Collaboration

Version Control

- GitHub (https://github.com/)
- Control Updates.
- Team Collaboration.