**[FPS Assessment] Test Plan Document**

Author: [Ethan McGimpsey]

**Introduction**

[It is a brief summary of the product that is being tested. Outline the functions being tested at a high level.] The product I'll be testing is my FPS project which is a First Person Shooter game in which the player can walk around, jump and shoot enemies. The functions that are going to be tested will be checking whether the enemy is taking damage by the player's weapon and check if the weapon is firing when the player is holding down the mouse button.

Objectives  
[Describe the objectives of your Test Plan, **i.e.**, defining tasks and responsibilities, types of documents used, overview of what is being tested, etc.] The Objetives of this Test Plan Document are to ensure that certain mechanices in my project are functioning without any errors via Unity's Unit Test. Unit Test will be testing a single units of code, such as if the player can able to fire or wether the enemy is taking damage by the player's weapon.

Tasks  
[List all the tasks identified by this Test Plan, **i.e.**, testing, documenting, post-testing, problem reporting, etc.]

* Testing
* Documenting
* Problem Reporting

Scope

**Inclusion:** [High level overview about what is being tested] The players ability to damage the enemy with the weapon. The test will be checking to ensure that when the player fires the weapon at the enemy, the enemies health will start to drop and then eventually die. **Exclusion:** [High level overview about what is NOT being tested] Other game scripts is not going to be tested **Tactics:** [How will you accomplish the items in the scope] By using a inbuilt Unity program called Unit Testing system.

**Testing Strategy**

Unit Testing

**Definition:** [What are you testing? What is the minimum amount of detail being tested?] Testing to see if the player is able to fire with the weapon and the minimum detail being tested will be making sure that the enemy is able to damage the player when firing.

**Methodology:**[Describe how unit testing will be conducted.] Unit Testing will be led by utilizing the Test Runner which is the unit testing feature gave by Unity by using the NUnit system guaranteeing everything acts as it should. The steps on how to use the Unit Testing can be broken down to few simple points:

* Prepare
* Review
* Rework
* Baseline

Tools  
[List the Automation tools that you are going to use.]

* Unity
* Unity's inbuilt Unit Testing software

**Unit Test Plan**

Test Area 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#ID** | **Test Case Description** | | **Comments** | | **Pass/Fail** |
| 1 | Shooting (Player) | | Testing to see if the left mouse button makes player shoot | | Pass |
| **#ID** | **Execution Step** | **Expected Results** | | **Obtained Results** | |
| 1 | [Test being run] | The player able to open fire when holding down the left mouse button | | Player was able to shoot | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#ID** | **Test Case Description** | | **Comments** | | **Pass/Fail** |
| 1 | Enemy Dead | | Testing if the enemy ragdoll itself when his health < 0 | | Pass |
| **#ID** | **Execution Step** | **Expected Results** | | **Obtained Results** | |
| 1 | [Test being run] | The enemy triggers itself to ragdoll | | The enemy died and ragdolled itself when his health was < 0 | |