­System and Unit Test Report

2Sticks4Shooting

Poverty Studios

3.12.2014

**User story from sprint 1:**

*As a developer, I want a basic enemy for the player to interact with so we can test collision detection, health, and damage systems.*

This task was successfully implemented from the first sprint within the estimated time. History of the story is recorded in *Sprint* *Report 1.docx*. The qualities implemented can be demonstrated in a scenario like the following:

1. the team developed C# script containing a custom *Enemy* class is usable when developing in the Unity game engine. This script will allow an enemy object to interact with the other team defined objects pertaining to our *Player* class.

Our *Enemy* script involves methods dealing with:

- Collision detection

- Path finding

- Destroying (i.e. getting shot by the player, leaving boundaries)

- Damaging the player

- General information exchanging (w/ the player object)

2. as a developer, you have the ability to apply these properties to compatible objects in Unity. For example, you can develop an enemy spawn controller of some sort, and reapply this script to create multiple *Enemy* type objects on screen.

**User story from sprint 2:**

*As a player, I want to be able to have a choice as to what weapon to use.*

This story came to complete fulfillment by the second sprint, as documented in *Sprint Report 2.docx*. This story reflected an aspect of personality we wanted the game to have. It was important that the player have multiple options of weapons, and an ability to switch between equipped weapons at will. Given a scenario like so, the player can select from three unique weapons in a scenario like the following:

1. by default, the player is spawned with a basic gun.

- Assigned to the player by default

- Implements a unique "basic gun" reticule

- fires a specific "basic" bullet

2. player picks up "Reverse Shotgun" equipment

- Implements a unique "shotgun" reticule

- fires many bullets

3. player picks up "Sin Gun" equipment

- Implements unique "sin wave" reticule

- fires bullets in a unique pattern

4. player initiates the "Weapons switch" button (Defined by specific user input type)

- Toggles between equipped weapons

- i.e., the player can switch between the "Sin Gun" and "Reverse Shotgun" if both are simultaneously picked up.

**User story from sprint 3:**

*As a player, I want to be able to play on a controller*

This story was completed by sprint 3, and as such its documentation is recorded in *Sprint 3 Report.docx*. Controller support referred to 3rd party means of input that the user may have wanted to use. In our game we implemented both WASD + mouse controls and a means of inputs utilizing a standard XBOX360 controller.

1. start up the game on a PC with no user device connected

- Controls will default to MOUSE + KEYBOARD

- Move Up: W Fire Weapon: Left mouse button

- Move Down: A Reload: R

- Move Left: S Switch Weapon: Q

- Move Right: D Move Reticule: Mouse

2. start up the game on a PC with standard user device connected

- Controls will default to XBOX360 layout style

- Move Up: Left stick +Y Fire Weapon: Right bumper

- Move Down: Left stick -Y Reload: A button

- Move Left: Left stick -X Switch Weapon: X Button

- Move Right: Left stick +X Move Reticule: Right stick movement