**Gregg Moscosa** **Raul Flores**

Interactions and Scripted Enemies

Iterative Improvement Worksheet

## Modifications to the Given Enemy:

1. Write one or more player stories to describe the experience of fighting your improved enemy. Describe what happens action-by-action, paying special attention to what choices or skill challenges the player has that impact winning or losing.

* **Starting the game off, we can approach the patrolling enemy, and it will chase us with an updated speed, and attack cooldown timer. Enemy will keep attacking until we die.**

1. Detail the new actions involved (if any). Give enough detail to capture what is mechanically different about each action (e.g. targeting, windup, hit mechanics, telegraph, cooldown).

* **Enemy has a longer cooldown period to perform attacks, to make the game more balanced.**

1. Detail the new systems (if any) needed to implement those actions (e.g. building up combo points and then using them, putting a trap on the ground that blows up later).

* **Updated local variables to modify enemy actions and behaviors. Changed the speed and location of patrol waypoints to alter behavior.**

1. Detail the new behaviors involved, and how they work together. Be sure to be specific about the conditions that trigger transitions between behaviors.

* **Chase behavior has been updated to only follow Player. To achieve this we added a “Player” tag to the player, so the enemy can differentiate between the other enemy and the player to follow.**

1. Implement and test (you can iterate here as much as needed).

* **Iteration 1: After updating the local variables, speed and cooldown of attack were updated, but the enemy would get confused and follow the other enemy instead of the player.**
* **Iteration 2: After adding the “Player” tag to the player object, we were able to make the enemy only follow the player, even when the other enemy was close.**

1. Reflect and report briefly on how your implementation did or did not achieve the player experience you were going for.

* **The modifications made on the local variables to alter movement speed, and attack cooldown worked as expected. The main obstacle was making the enemy follow the player only and not the enemy. After being able to fix that issue, we were able to achieve the desired behavior change for enemy #1.**

## New Second Enemy:

1. Write one or more player stories to describe the experience of fighting your improved enemy. Describe what happens action-by-action, paying special attention to what choices or skill challenges the player has that impact winning or losing.

* **Starting the game off, we will notice that Enemy #2 behaves similarly to Enemy #1, as the variable updates also changed the speed and cooldown of this enemy. But once the enemy gets close to the player, it will initiate a new action script that will make the enemy dash towards the player every 6 seconds. Enemy will continue to follow, and dash relentlessly towards the player until the games ends.**

1. Detail the new actions involved. Give enough detail to capture what is mechanically different about each action (e.g. targeting, windup, hit mechanics, telegraph, cooldown).

**-With this enemy, we implemented a new Dash action by creating a new script, and attaching the component to the enemy so it will use the new dash ability. This action will make the enemy dash forward towards the player, giving it strategic advantage in case the player is trying to get away. To make this action balanced towards the player, the Dash script will only activate every 6 seconds to give the player a fair chance to escape.**

1. Detail the new systems needed to implement those actions (e.g. building up combo points and then using them, putting a trap on the ground that blows up later).

-**To implement our Dash action, we had to create a new script to be able to give this enemy a new ability. Using local variables to determine the dash speed, and the timer for the dash cooldown, we were able to implement conditional statements to perform the dash at the desired times. We then utilized the tag system to get the player location, and use our dash ability towards that point where the player is.**

1. Detail the new behaviors involved, and how they work together. Be sure to be specific about the conditions that trigger transitions between behaviors.

-**Aside from the speed and attack cooldown, the new behavior implemented on this enemy was giving it a new set of waypoints to patrol. We created new empty variables that hold the new waypoint locations, and then applied it to this enemies Patrol script instance. This will make Enemy #2 patrol a different section than Enemy #1.**

1. Implement and test (you can iterate here as much as needed).

**-iteration 1: Starting off with trying to apply the Dash action, we ran into some issues with making the enemy actually perform the dash, and also the direction the dash was going.**

**-iteration 2: We were able to get the Enemy to perform the dash, as well as being able to dash towards the player object.**

**-iteration 3: We changed the behavior of this enemy, by giving it a new set of waypoints to it patrols differently than Enemy #1.**

**-iteration4: To give the game some aesthetic improvements, we added some game sprites to the player and enemy objects to make it look more appealing and fun.**

1. Reflect and report briefly on how your implementation did or did not achieve the player experience you were going for.

* **The desired implementation for Enemy #2 was relatively close to what we intended for the end outcome. The only issue we encountered when applying the new Dash action, was that the enemy did not dash further when we changed the dash variables, and that the enemy will dash through the player, and not collide with it after the dash. Further improvements towards player/enemy collisions would be ideal.**

**Iteration #2:**

## Modifications for Third and final Enemy:

1. Write one or more player stories to describe the experience of fighting your improved enemy. Describe what happens action-by-action, paying special attention to what choices or skill challenges the player has that impact winning or losing.

-Enemy starts to flee as soon as the player is in range of this Enemy. It will always continue to **flee away from player, and if Player gets close enough to this enemy, It will heal the player health.**

1. Detail the new actions involved (if any). Give enough detail to capture what is mechanically different about each action (e.g. targeting, windup, hit mechanics, telegraph, cooldown).

* **The new action given to this Enemy has been the ability to heal the player, but it will not give it so easy as it will always be fleeing away from the player.**

1. Detail the new systems (if any) needed to implement those actions (e.g. building up combo points and then using them, putting a trap on the ground that blows up later).

- **Attempted to implement another enemy script called Enemy2, to manipulate this enemy only. However, it was not necessary to use it as we were able to use the original enemy script, and turned the attack action into a healing action instead.**

1. Detail the new behaviors involved, and how they work together. Be sure to be specific about the conditions that trigger transitions between behaviors.

* **The new behavior for this enemy is fleeing away from the player, as opposed to the other enemies who chase the player to fight him. Implementing the Flee script, we will make this enemy keep fleeing so it will not use the heal action all the time on the player.**

1. Implement and test (you can iterate here as much as needed).

**-Iteration1: Enemy was not fleeing from player when implementing the flee script. Attempted to use a separate Enemy script (Enemy2) to manipulate the behavior of this enemy only, however it did not work and we ended up using the original Enemy script to make Flee work.**

**-Iteration2: Now that we have the Flee script working, we changed the way the attack works for this enemy, turning it into a healing ability instead of a damaging one. This will give the player some health back if it is able to catch up to the enemy.**

**-Iteration3: For the final iteration of this enemy, we were able to implement the healing action successfully. Healing the player to full health once it catches up to this special enemy.**

1. Reflect and report briefly on how your implementation did or did not achieve the player experience you were going for.

* **So far the Flee behavior is working as expected, the only obstacle we have encountered is making the healing action always heal the player. As it will only heal the player with the first shot. We managed to overcome this obstacle by clearing the attack list before continuing with our “healing” attacks to ensure that every consecutive attack done by this enemy will heal the player to full health.**