CMPT 742 – Assignment 3
Fall 2019

Group:
Luciano Oliveira #301393900
Chenhao (Zoe) Wang #301409375

Report

Parts of the assignment:

a) Creating the basic enemy character

We've created 4 enemies with their red uniform and red helmet, and an enemy controller script. We also have 8 different targets around the game, so each enemy is assigned different targets to walk to (and also a different number of targets). In the enemy controller script, whenever they reach a distance of 1 meter or less to a target (calculated using Vector3.Distance), they switch to the next target (we use the mod operator with the number of targets, so after the last target it goes back to the first one).

b) Detecting the player

The enemy is able to detect the player based on their field of view (with an angle of 45 degrees from his “forward vector”). We converted 45 degrees to radians, then calculated the cosine of it, and compared it to the Vector3.Dot product of his “forward vector” and the normalized vector from player to enemy. Moreover, in order for the enemy to be able to successfully detect the player, the player should be at least within 20 meters from him. If they detect the player, we use the transform.LookAt function to make it turn to player.
We also have a “Rifle Run” animator, so whenever an enemy detects the player, they will run towards the player until reaching a distance of 10 meters. We also have a transition from the “fire” animator to the “Rifle Run” animator, so if the player runs away, the enemy will run again to chase him in order to maintain a distance of 10 meters.

c) Shooting to the player

When player is detected and enemy reaches a distance of 10 meters from the player, it will shoot (with sound) the player because of its “fire” animator that has a transition coming from “Any State”. We have a variable to count the amount of time that has passed (based on Time.deltaTime), and whenever it's higher than 0.2 seconds, the enemy will fire the gun (and we reset the counter by subtracting 0.2 from it). This assures that the enemy will fire 5 times per second. We make the enemy aim for the player using Physics Raycast, but before that we randomize 2 variables (with values from -0.1 to 0.1) that will change the X and Y coordinates of the “end” position of the gun. When player dies, it runs the “Dying Backwards” animator, and camera is moved to one of the two different fixed camera positions that we have (one camera for each room).

d) Player shooting and health

Player can shoot, reload, and walk (it can also walk with keys A, S, D, W from keyboard, and fire with mouse click). When the player shoots a wall, it creates some holes (which disappear after 10 seconds), or if it shoots an enemy, it hits for 35 health points. When player is shot, it loses 20 health points per hit, and its health can be seen on screen via some UI Canvas. Enemies' weapons have a Rigidbody and a Box Collider that is inactive. Whenever an enemy dies, its weapon's Box Collider gets activated and it becomes a separate object from its body. Whenever an enemy is shot, that specific enemy will automatically detect the player.

e) Creating the environment

The game has 4 enemies and 2 rooms (2 enemies per room), and also one fixed camera per room. Every enemy has its own set of path targets (game has 8 targets at total; 4 per room). The environment also has a corridor between the two rooms. At the end of the final room, there's a door object (downloaded from free3D.com), which has a trigger collider. This trigger has a function that disables the player's scripts and Character Controller. It will also destroy the door (for making an illusion that the door was opened). After 10 seconds, it will restart the scene and the game will start again. Game is also restarted after 10 seconds if they player dies. Additionally, it will also show a different message based on whether the player has won the game or died.

f) Ammo supply

We've downloaded another object in the shape of a bullet. It's been placed in the middle of the corridor between the rooms. Whenever the player gets within 1 meter of it, the bullet object will disappear (as well as being inactivated), and the player ammunition will be refilled by 30 extra bullets.

g) Enemies getting cover

Not implemented.

h) Detecting body part hits

Enemies have different Box Colliders for head, arms, and legs. So every collider has a different tag (head, arm, or legs), and a new script attached to them called BodyParts.cs. So whenever a player hits an enemy, it will detect the tag and reduce their health accordingly. Hitting the head, chest, arms, and legs reduce their health by 100, 30, 10, and 20 respectively.
There's also a new Box Collider for the player head with the tag called “head”. So if a player is hit on the head, it dies instantly.

I) Swapping guns

Not implemented.