Unity Assignment

Pre- Requisites

1. Reload Animation- We use events for a smooth animation. Event 1 causes the hand mag to appear while gun mag to disappear. Event 2 does the reverse.

2. Putting Holes in the wall- We use RayHit and instantiate a gameobject with the location of the rayHit collider. The gameobject is passed to Destroy function with destroys the objects in 1s.

3. Muzzle Flash- addEffects function is called whenever shot is fired. Using GetComponent<ParticleSystem>.()Play() the muzzle flash is played.

4. Shooting sounds- Similar to muzzle flash, an audio source is added which gets played after firing of the shot.

Assignment

1.1.1 ENEMY CHARACTER (Completed)

The movement of an enemy is defined by the target nodes. There are four nodes each having different positions. Both the enemies follow the 1-2-3-4 nodes successively, however, as the positions of the walkpath is different for both the enemies, each character has a different path from other.

To make the enemy come back and start to its original position, “modulo” function is used. The distance between the current target and the original position is found. If it gets smaller than 0.1, the current target is changed to a new one. Using the modulo function causes the target after the last to return to 0.

To change the appearance of the enemy, a new material is made. The texture provided by file Soldier_Body_diffuse_red.png is used.

1.1.2 DETECTING PLAYER(Completed)

To detect the player if it get’s closer to the enemy is done by finding the distance of the enemy and the player. If the distance lies between the threshold or if the player shoots the enemy, the “found_player” variable turns true.

A run parameter is used in the animator and is set to true if the player is found until the distance is less than 10.

1.1.3 ENEMY SHOOTING THE PLAYER(Completed)

To start shooting at the player and detecting the shots, RayCast is used. If the distance between the enemy and player lies in the range, the “fire” parameter is set to true. Time.Deltatime is used to calculate the time between successive frames. A temp variable is set to 0 before shooting starts and is added by time.deltatime to find elapsed time. If it gets higher than 0.25 (as 4 bullets are fired per sec), the function for shooting at the player gets called.

Using Random.Range on the up and right transform causes the shot to either “miss” the player or hit the player. Using Physics.RayCast and rayhit.transform.tag, we check whether the player is hit. If the player is hit, depending on the body part, the damage is subtracted from the health.

Once the player’s health decreases to zero, the death from front headshot animation is played. The character controller is disabled by setting the .enabled to false. Using SceneManager, the game is then restarted after 10secs. The isDead flag is set to false, which allows the camera to pan to a fixed position.

1.1.4 PLAYER SHOOTING THE ENEMY AND HEALTH(Completed)

Using the same function template described for the player, the player too can shoot at the enemies and also kill them. RayCast followed by calling the being_shot function makes the enemy get damage.

The player’s health is also added to the UI by copying the ammo canvas and using the copy for the health display.

Once the enemy gets killed, the dead parameter is set to true which causes the death animation to occur. Setting transform.parent to null causes the gun to be a separate entity from the enemy. BoxCollider and RigidBody component is added using AddComponent method and the enabled flag is also set to true. CharacterController is disabled and isKinematic is set to false so no physical interactions take place.

The enemy is able to detect the player when the player starts shooting at him or the distance between them lies within the range.

1.1.5 GAME ENVIRONMENT(Completed)

The game has two rooms and two enemies. Both of them a unique set of targets as the position

is different for both. An escape door is also added in the second room. A BoxCollider component is added to it. Checking the Trigger box, allows us to use, ”OnTriggerEnter”. If both the enemies are killed, the game is won and then restarted after 10 seconds.

BONUS POINTS

1.2.1 AMMUNITION SUPPLY (Works)

Ammo crates asset is added to the environment. If the player get's close to it, animation gets fired up, followed up by an increase of 30 in the ammo.

1.2.2 ENEMIES GETTING COVER (Works)

If the last enemy detects the player, it runs to the cover. Using two different states, the enemy is made to cover himself and using invoke causes him to fire after some seconds. This is repeated until the player dies or enemy is killed.

1.2.3 DETECTING BODY PART HITS (Works for hands,legs. Partly works for head)

Different body parts are detected using tags. Soldier_body is given a hands tag, head is given head tag while Hips is given the tag of legs. If any of the them is detected, the damage is changed to the given one.

1.2.4 SWAPPING GUNS (Works)

Pressing the spacebar , using Input.GetKeyDown, allows the player to swap a gun. The gun asset and two different animations for pull out and put away is downloaded. Once space is pressed, the animation for put away is ran. After exiting, pull out animation is triggered following by setting gun2 as active. Pressing space toggles the active nature of the guns.