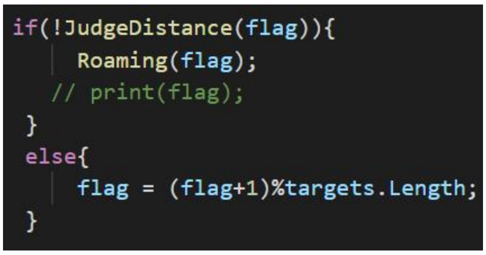
Overview

Add three extra scripts: Player, Enemy and Healthbar. The Enemy realizes the function of detecting player, maintaining the distance, shooting at the player at a random accuracy and separating gun with the enemy when dying. These functions work only when the player is not dead. The player realize the function of refiling the ammunition and escaping. The Health bar is to show the health value of the player in the UI Canvas.

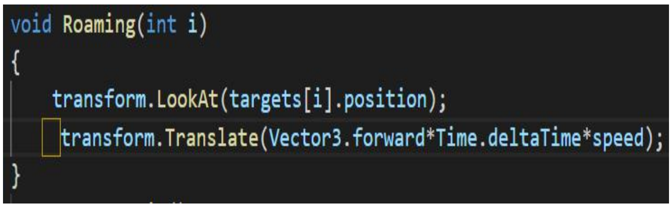
1.1.1 Enemy Character

Enemy walking in different path

walk towards the position if not reach, else, walk towards the next position

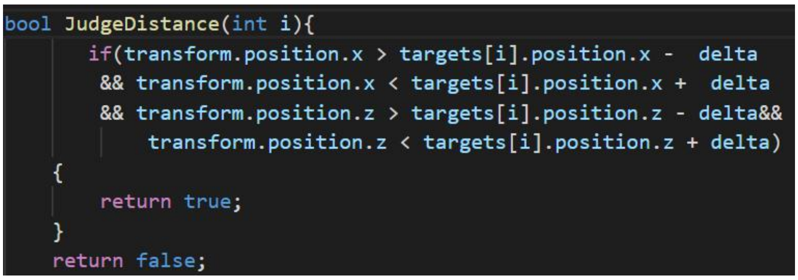


```
if(!JudgeDistance(flag)){
        Roaming(flag);
     // print(flag);
 }
 else{
        flag = (flag+1)%targets.Length;
 }
```



```
void Roaming(int i)
{
    transform.LookAt(targets[i].position);
    transform.Translate(Vector3.forward*Time.deltaTime*speed);
}
```

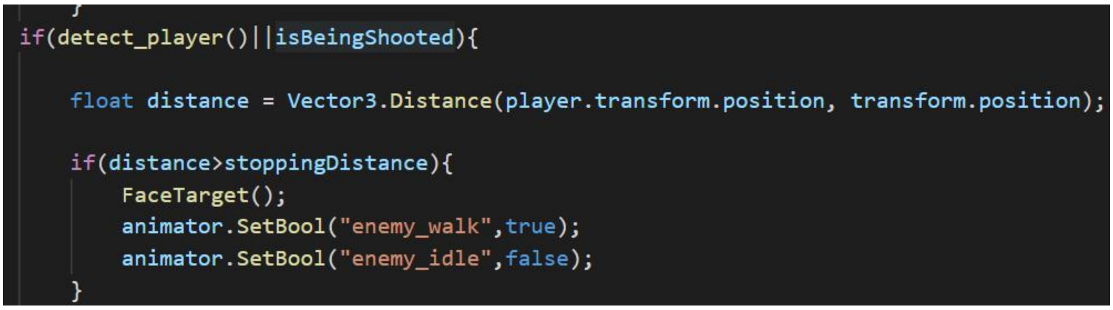
If the distance is less than delta, then we assume the enemy reaches the position



```
bool JudgeDistance(int i){
        if(transform.position.x > targets[i].position.x -  delta
         && transform.position.x < targets[i].position.x +  delta
         && transform.position.z > targets[i].position.z - delta&&
             transform.position.z < targets[i].position.z + delta)
    {
         return true;
    }
    return false;
```

1.1.2 Detecting Player

When the enemy detects the player and does not reach the shooting distance, the enemy runs runs towards the player. The function run the run animation and also this function can maintain the distance.



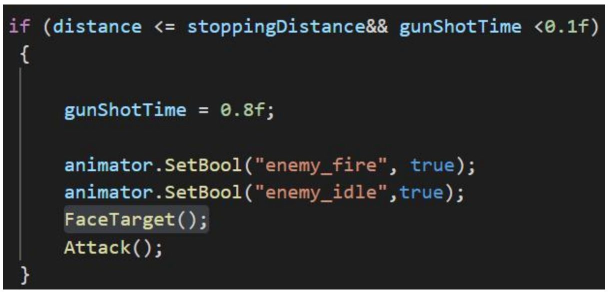
```
 if(detect_player()||isBeingShooted){

     float distance = Vector3.Distance(player.transform.position, transform.position);

     if(distance>stoppingDistance){
         FaceTarget();
         animator.SetBool("enemy_walk",true);
         animator.SetBool("enemy_idle",false);
     }
```

1.1.3 Shooting the player

When the enemy reaches the shooting distance, it stops and shoots at the player at a random accuracy. In the Player script, it runs the shooting animation to shoot and the idle animation to stop(added part,didn’t show in demo).

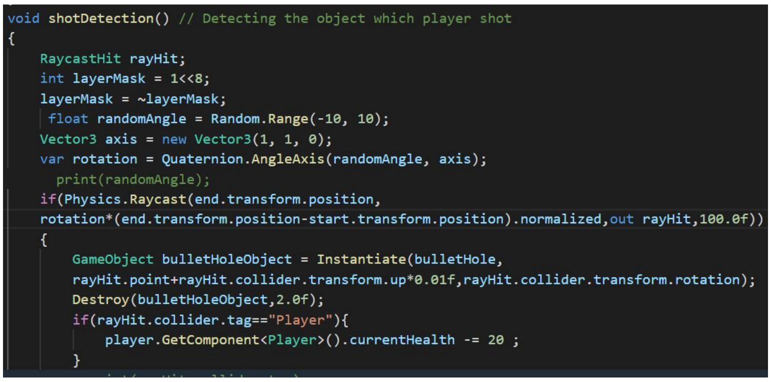


```
if (distance <= stoppingDistance&& gunShotTime <0.1f)
 {

     gunShotTime = 0.8f;

     animator.SetBool("enemy_fire", true);
     animator.SetBool("enemy_idle",true);
     FaceTarget();
     Attack();
 }
```

Use Random() to rotate the ray at a random direction, and reduce the player's health by 20 when the ray hits the player.

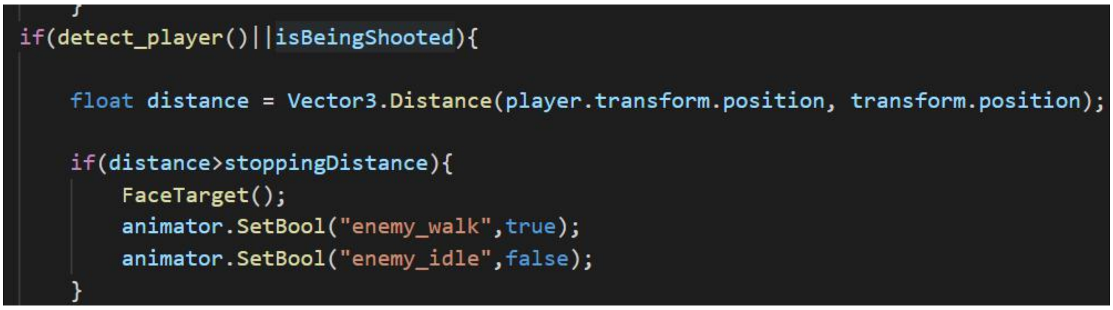


```
void shotDetection() // Detecting the object which player shot
{
    RaycastHit rayHit;
    int layerMask = 1<<8;
    layerMask = ~layerMask;
     float randomAngle = Random.Range(-10, 10);
    Vector3 axis = new Vector3(1, 1, 0);
    var rotation = Quaternion.AngleAxis(randomAngle, axis);
      print(randomAngle);
    if(Physics.Raycast(end.transform.position,
    rotation*(end.transform.position-start.transform.position).normalized,out rayHit,100.0f))
    {
        GameObject bulletHoleObject = Instantiate(bulletHole,
        rayHit.point+rayHit.collider.transform.up*0.01f,rayHit.collider.transform.rotation);
        Destroy(bulletHoleObject,2.0f);
        if(rayHit.collider.tag=="Player"){
            player.GetComponent<Player>().currentHealth -= 20 ;
        }
```

When the player dies, call the SceneManager.LoadScene() to restart the game.

1.1.4 Player shooting and health

The health value of the player will show in a health bar when the player's health decreases. The gun will be parented to the Enemy when the enemy dies so that it can be separated from the enemy. Add an variable isBeingShooted in the Enemy script, when the player shoots the enemy, it will be set to true, and the enemy will detect the player and run towards the player, as what the enemy does in detecting enemy part. This function is achieved by add an condition to if() in the Enemy Script.



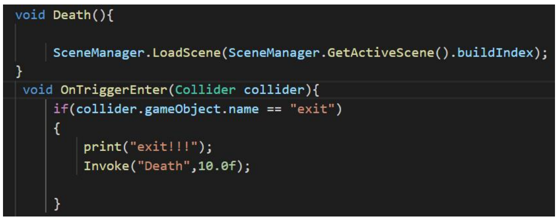
```
if(detect_player()||isBeingShooted){

    float distance = Vector3.Distance(player.transform.position, transform.position);

    if(distance>stoppingDistance){
        FaceTarget();
        animator.SetBool("enemy_walk",true);
        animator.SetBool("enemy_idle",false);
    }
```

1.1.5 Game environment

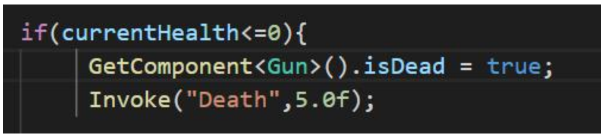
When the player reaches the escape door, the OnTriggerEnter function in the Player script is called to reload the scene. When the player gets killed , the health value decreased to zero, and the Death function is also called.



```
void Death(){

    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
}
 void OnTriggerEnter(Collider collider){
    if(collider.gameObject.name == "exit")
    {
        print("exit!!!");
        Invoke("Death",10.0f);

    }
```

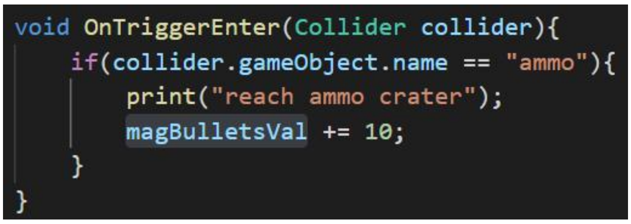


```
if(currentHealth<=0){
    GetComponent<Gun>().isDead = true;
    Invoke("Death",5.0f);
```

Bonus

1.2.1 Ammunition Supply

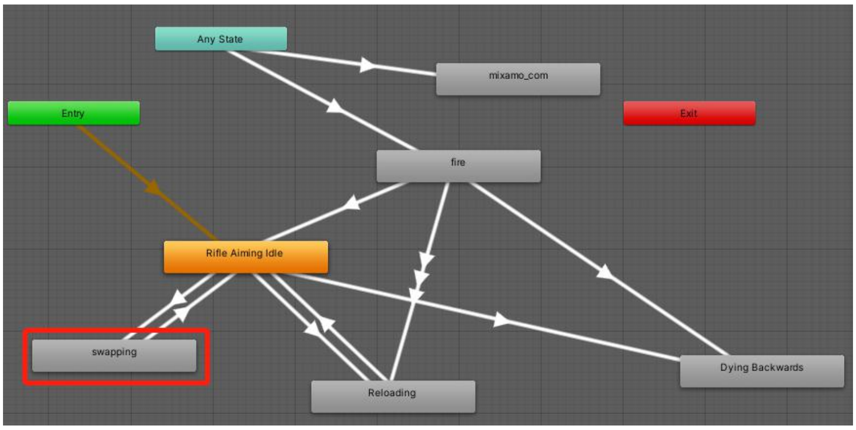
When the player reaches the ammo crater, the OnTriggerEnter function in the Player script is triggered to add the variable magBulletsVal.

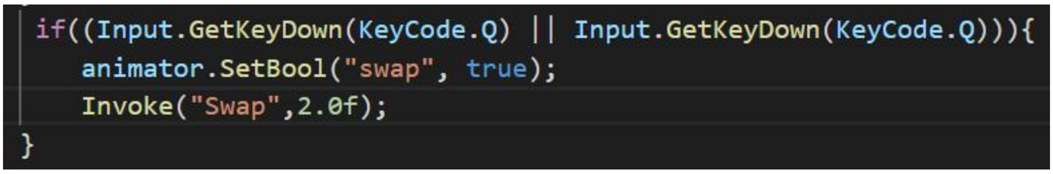


```
void OnTriggerEnter(Collider collider){
    if(collider.gameObject.name == "ammo"){
        print("reach ammo crater");
        magBulletsVal += 10;
    }
}
```

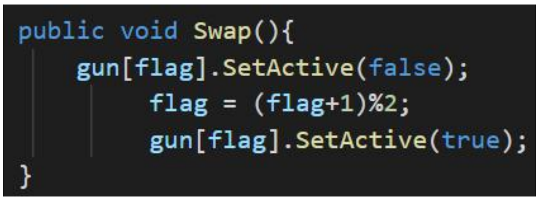
1.2.4 Swapping guns(added part, didn't show in demo)

Add swap animation to the animator, and run the animation when pressing Q





```
if((Input.GetKeyDown(KeyCode.Q) || Input.GetKeyDown(KeyCode.Q))){
    animator.SetBool("swap", true);
    Invoke("Swap",2.0f);
}
```



```
public void Swap(){
    gun[flag].SetActive(false);
        flag = (flag+1)%2;
        gun[flag].SetActive(true);
}
```