IT 4001: Virtual Reality

Assignment 5: Gaze Control

Task 1: Kill Target Script (60 pts)

- Follow along in class and the Gaze Control instructions (available on Canvas), build the basics of the project
- Create an empty game object "GameController" and attach "KillTarget" script below to it. The script provided below is incomplete. Complete the script at places marked with `//ToDo`.

```
using UnityEngine;
using System.Collections;

public class KillTarget : MonoBehaviour {
 public GameObject target;
 public float timeToSelect = 3.0f;
 public ParticleSystem hitEffect;
 public GameObject killEffect;

 private float countDown;


    // Use this for initialization
    void Start () {

     countDown = timeToSelect;
     hitEffect.enableEmission = false;
    }

    // Update is called once per frame
    void Update () {

     Transform camera = Camera.main.transform;

     Ray ray = new Ray(camera.position, camera.transform.rotation * Vector3.forward);

     RaycastHit hit;

     if (Physics .Raycast (ray,out hit) && (hit. collider .gameObject == target ))
     {
         if (countDown <0f )
         {
             //ToDo: Kill target. Instantiate kill Effect. Reset the countdown.

         }
         else
         {
```

```
                //ToDo: Decrement countdown with Time.deltaTime amount. Enable the hitEffect,
and place it at hit.ponit.

            }

        }
        else
        {
              //ToDo: Reset countdown. Disable hitEffect.

        }

      }
}
```

Some helpful information:

- In KillTarget.cs, instantiate killEffect using *Instantiate ( )* method:

  ```
  Instantiate(killEffect, target.transform.position, target.transform.rotation);
  ```

  https://unity3d.com/learn/tutorials/topics/scripting/instantiate

  https://docs.unity3d.com/ScriptReference/Object.Instantiate.html

- Set hitEffect at the location of hit:

  ```
  hitEffect.transform.position = hit.point;
  ```

- To enable/disable hitEffect emission, use:

  ```
  hitEffect.enableEmission = true; //true or false
  ```

  However, this enableEmission is obsolete. The current API
  https://docs.unity3d.com/ScriptReference/ParticleSystem.EmissionModule-enabled.html

  ```
  var emission = hitEffect.emission;
  emission.enabled = true;
  ```

  Adding particle effects

  Now, to populate the public variables (target, hitEffect and killEffect) in the script, perform the following steps:

  1. First, you need the ParticleSystems package that comes with Unity standard assets. If you do not have them, navigate to **Assets** | **Import Package** | **ParticleSystems**, choose **All**, and then click on **Import**.

2. Select GameController from the **Hierarchy** panel and go to the **Kill Target (Script)** pane in the **Inspector** panel.

3. Drag the Ethan object from the **Hierarchy** panel onto the **Target** field.

4. From the main menu bar, navigate to **GameObject** | **Particle System** and name it SparkEmitter.

5. Reselect GameController and drag SparkEmitter onto the **Hit Effect** field.

We created a default particle system that will be used as the sparks emitter. You need to set that up to your liking. The following steps get you started, and you can play with it as you desire,

1. Select SparkEmitter from the **Hierarchy** panel.
2. And in its **Inspector** panel, under **Particle System**, set the following values:
    - °° **Start Size**: 0.15,
    - °° **Start Color**: pick a red/orange color
    - °° **Start Lifetime**: 0.3,
    - °° **Max Particles**: 50.
3. Under **Emission**, set **Rate**: 100.
4. Under **Shape**, set **Shape**: **Sphere** and **Radius**: 0.01.
5. For performance, under **Renderer**, set **Cast Shadows**: **Off**, **Receive Shadows**: uncheck, and **Reflection Probes**: **Off**.

To learn more, visit:

https://docs.unity3d.com/Manual/ParticleSystemModules.html

https://unity3d.com/learn/tutorials/topics/graphics/particle-system

6. In the **Project** panel, find the Explosion prefab in Assets/Standard Assets/ParticleSystems/Prefabs, drag the Explosion prefab onto the **Kill Effect** field.

- Click play and test your project. At this point, you should have the following features through KillTarget.cs: looking at Ethan to hit him with the line-of-sight ray gun; sparks are emitted when the gun hits target; after 3 sec of being hit, Ethan is killed. You can easily test these features before moving to the next task by disabling the scripts on WalkTarget, so Ethan doesn't move.

Task 2: Game Design (40 pts)

- Enable RandomPosition script on WalkTarget. Ethan will follow the random moving WalkTarget.
- Modify *KillTarget.cs*, instead of calling function *Destroy (target)* that destroys the target game object - Ethan, you are to respawn Ethan at a new random location, and reward player 1 point.
- Improve the environment by adding some interesting environment objects – such as trees from asset store.

***Bonus (10 pts)*.** Display the score points using UI.

Turn in your completed project (zipped folder) on Canvas. You are required to demo your project to TA/Instructor.