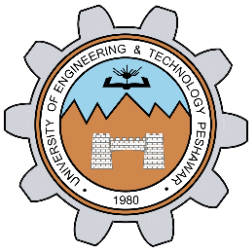
Unity API (Input Axis, Raycast and Instantiation Methods)

LAB # 6



Fall 2024

CSE-411L Intro to Game Development Lab

Submitted by: **Ali Asghar**

Registration No.: **21PWCSE2059**

Class Section: **A**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Submitted to:

Engr. Abdullah Hamid

Date:

21st December 2024

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Objective:

In this lab we further explored the Unity API.

Tasks:

- Open/create a Unity scene.

- Create a player cube that moves forward, backward, left, and right using axis.

- The camera in the scene should be set to a top-down view.

- The player should be able to shoot bullets the bullets must destroy themselves after 2 seconds.

- Create 3 enemies on the plane.

- When the bullets hit the enemies, the enemies must turn first yellow and then red and the then destroy themselves with 1 second interval between each behavior .

- Create an array of these strings (Dead!, Killed!. Defeated!) when the enemy dies any of these messages should print on console “Randomly”

Code:

Player class

Player.cs | EnemyManager.cs | Enemy.cs | Bullet.cs

Assets > Labs > Lab6 > Scripts > Player.cs > Player > Update

```
using UnityEngine;
namespace Lab6{
    0 references
    public class Player : MonoBehaviour{
        2 references
        RaycastHit hit;
        1 reference
        public GameObject bulletPrefab;
        2 references
        GameObject bulletContainer;
        4 references
        public Transform spawnPoint;
        2 references
        public float movSpd = 1;

        // Update is called once per frame
        0 references
        void Update(){
            float h = Input.GetAxis("Horizontal")*movSpd*Time.deltaTime;
            float v = Input.GetAxis("Vertical")*movSpd*Time.deltaTime;

            transform.Translate(new Vector3(h, 0, v));

            if (Physics.Raycast(spawnPoint.position,
                Vector3.forward, out hit, Mathf.Infinity)){
```

```
16
17              if (Physics.Raycast(spawnPoint.position,
18                  Vector3.forward, out hit, Mathf.Infinity)){
19                  Debug.DrawRay(spawnPoint.position,
20                                Vector3.forward * hit.distance,
21                                Color.red);
22              }
23
24              if (Input.GetKeyDown(KeyCode.F)){
25                  bulletContainer = Instantiate(bulletPrefab,
26                                                spawnPoint.position,
27                                                spawnPoint.rotation);
28                  Rigidbody bullet_rb = bulletContainer.GetComponent<Rigid
29                  bullet_rb.AddForce(transform.forward * 10000f);
30              }
31          }
32      }
33  }
```

EnemyManager class

Player.cs | EnemyManager.cs

Assets > Labs > Lab6 > Scripts > EnemyManager.cs > EnemyManager > showMessage

```
1   using UnityEngine;
2
    2 references
3   public class EnemyManager : MonoBehaviour{
        3 references
4       public static EnemyManager instance;
        2 references
5       public string[] deathMessages;
6
        0 references
7       void Awake(){
8           if (instance == null)
9               instance = this;
10          else
11              Destroy(gameObject);
12      }
13
        1 reference
14      public void showMessage(){
15          int index = Random.Range(0,deathMessages.Length);
16          Debug.Log(deathMessages[index]);
17      }
18  }
19
```

Enemy class

Player.cs | EnemyManager.cs | Enemy.cs

Assets > Labs > Lab6 > Scripts > Enemy.cs > Enemy

```
1   using UnityEngine;
2
    0 references
3   public class Enemy : MonoBehaviour{
4
        0 references
5       void OnCollisionEnter(Collision other){
6           var rendrer = gameObject.GetComponent<MeshRenderer>();
7           rendrer.material.color = Color.yellow;
8           InvokeRepeating(nameof(DieEffect),1f,0.5f);
9       }
10
        1 reference
11      void DieEffect(){
12          Destroy(gameObject);
13          EnemyManager.instance.showMessage();
14      }
15  }
```

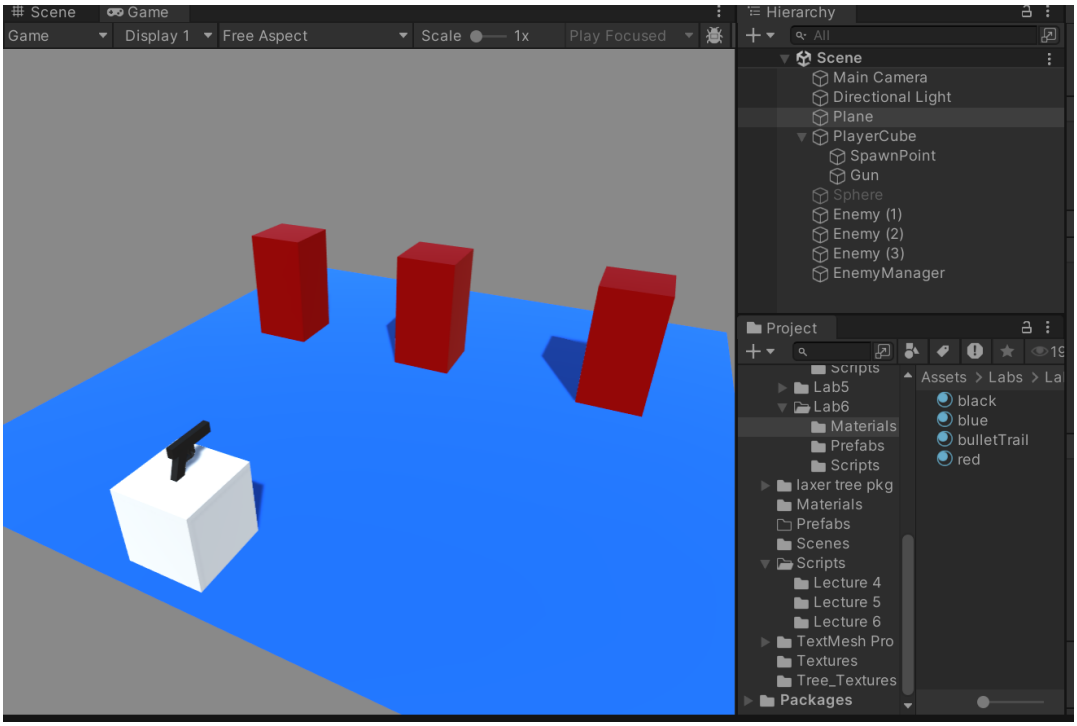
Enemy class

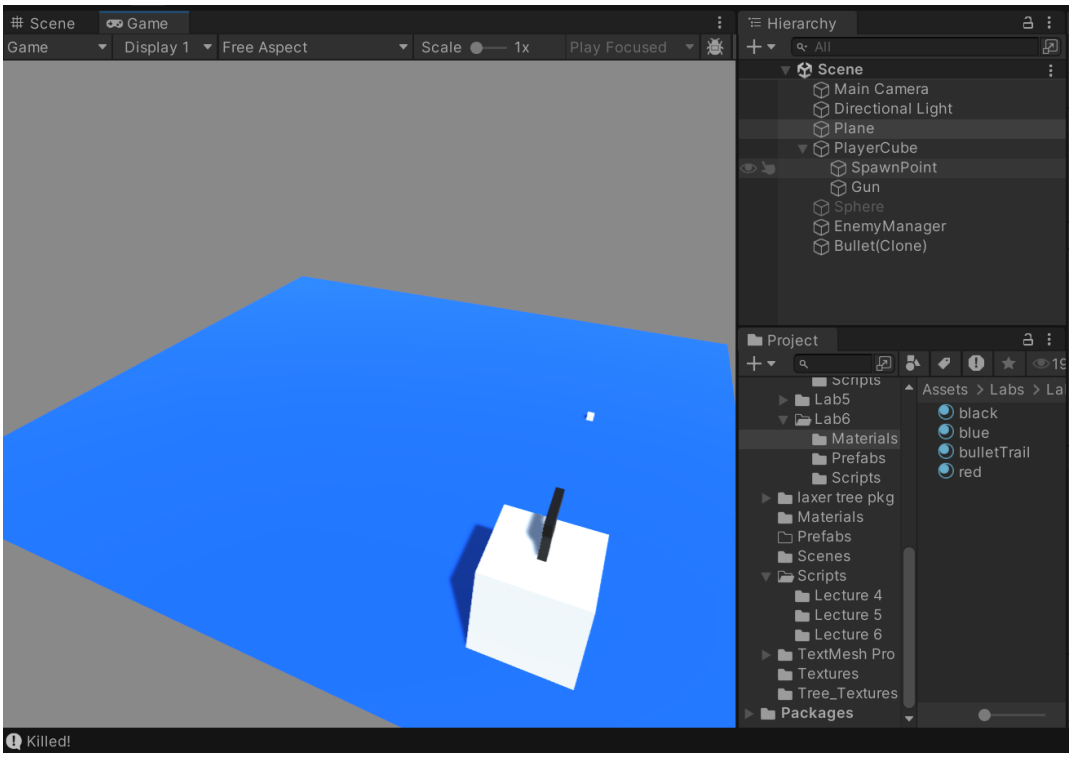
Player.cs | EnemyManager.cs | Enemy.cs | Bullet.cs

Assets > Labs > Lab6 > Scripts > Bullet.cs > Bullet > Start

```
1   using UnityEngine;
2
    0 references
3   public class Bullet : MonoBehaviour{
4       // Start is called before the first frame update
        0 references
5       void Start(){
6           Destroy(gameObject,2f);
7       }
8   }
```

Output:


Scene
Game
Game
Display 1
Free Aspect
Scale
1x
Play Focused
Hierarchy
All
Scene
Main Camera
Directional Light
Plane
PlayerCube
SpawnPoint
Gun
Sphere
Enemy (1)
Enemy (2)
Enemy (3)
EnemyManager
Project
Assets > Labs > La
black
blue
bulletTrail
red
Lab5
Lab6
Materials
Prefabs
Scripts
laxer tree pkg
Materials
Prefabs
Scenes
Scripts
Lecture 4
Lecture 5
Lecture 6
TextMesh Pro
Textures
Tree_Textures
Packages


Scene
Game
Game
Display 1
Free Aspect
Scale
1x
Play Focused
Hierarchy
All
Scene
Main Camera
Directional Light
Plane
PlayerCube
SpawnPoint
Gun
Sphere
EnemyManager
Bullet(Clone)
Project
Assets > Labs > La
black
blue
bulletTrail
red
Lab5
Lab6
Materials
Prefabs
Scripts
laxer tree pkg
Materials
Prefabs
Scenes
Scripts
Lecture 4
Lecture 5
Lecture 6
TextMesh Pro
Textures
Tree_Textures
Packages
Killed!