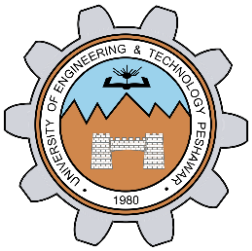
Unity API (QOL API, Timescale, Vector3 Methods, Unity UI)

LAB # 7



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:

24th December 2024

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Objective:

In this lab we further explored the Unity API.

Tasks:

1. **Open or create a Unity scene.**
2. **Create a player cube.**
    - Implement movement controls for the player cube using the axis input. This should allow the player to move forward, backward, left, and right.
3. **Set up the camera.**
    - Configure the camera to a top-down view, providing a bird's-eye perspective of the scene.
4. **Instantiate enemy cubes.**
    - After a 3-second interval, randomly instantiate enemy cubes on the plane.
    - Ensure that the instantiation process is confined to the plane's boundaries.
5. **Initialize game objects.**
    - At the start of the game:
        - Set the player cube's color to blue.
        - Set the enemy cubes' color to red.
        - Initiate enemy movement towards the player.
6. **Implement game over condition.**
    - When an enemy collides with the player:
        - Pause the game.
        - Display a "Game Over" message on the screen.
7. **Handle enemy interaction.**
    - When an enemy is clicked:
        - Stop the enemy's movement.
        - Destroy the enemy after a 1-second delay.
8. **Implement a score system.**
    - Display the current score on the screen.
    - Increment the score each time an enemy is destroyed.

Code:

EnemySpawner.cs

GameController.cs | EnemySpawner.cs

Assets > Labs > Lab7 > Scripts > EnemySpawner.cs > ...

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

namespace Lab7{
0 references
public class EnemySpawner : MonoBehaviour
{
    [Tooltip("Object Prefab to spawn")]
    2 references
    public GameObject enemyPrefab;

    [Tooltip("Number of objects spawn per second")]
    1 reference
    public float spawnRate = 1f;

    [Tooltip("Delay before 1st initial Spawn")]
    1 reference
    public float initialSpawnDelay = 1f;

    3 references
    private GameObject player;

    6 references
    public List<Vector2> points;
    4 references
    private  Bounds bounds;
    // Start is called before the first frame update
    0 references
    void Start()
    {
        player = GameObject.FindGameObjectWithTag("Player");
        float spawnFreq = 1/spawnRate;
        InvokeRepeating(nameof(SpawnOnCells),initialSpawnDelay,spawnFreq);
        Bounds bounds = gameObject.GetComponent<MeshRenderer>().bounds;
    }

    0 references
    void SpawnEnemyOnBounds()
    {
        float x = Random.Range(bounds.min.x + 0.5f, bounds.max.x- 0.5f);
        float z = Random.Range(bounds.min.z + 0.5f, bounds.max.z- 0.5f);
        Vector3 spawnPos = new Vector3(x,1f,z);
        var enemySpawned = Instantiate(enemyPrefab,
```

GameController.cs | EnemySpawner.cs

Assets > Labs > Lab7 > Scripts > EnemySpawner.cs > ...

```
6   public class EnemySpawner : MonoBehaviour
30      void SpawnEnemyOnBounds()
35          var enemySpawned = Instantiate(enemyPrefab,
36                                         spawnPos,Quaternion.identity);
37      }
38
        1 reference
39      void SpawnOnCells(){
40          Bounds bounds = gameObject.GetComponent<MeshRenderer>().bounds;
41          //Get Points
42          for (int x = (int)bounds.min.x + 1; x < (int)bounds.max.x; x++){
43              for (int z = (int)bounds.min.z + 1; z < (int)bounds.max.z; z++){
44                  if (Vector3.Distance(new Vector3(x, 0f, z), player.gameObject.transform.position) > 4){
45                      points.Add(new Vector2(x,z));
46                  }
47              }
48          }
49
50          //Vector3 spawnPos = points.OrderBy(x => Random.value).FirstOrDefault();
51          int i = Random.Range(0,points.Count);
52          Vector2 spawnPos = points[i];
53
54          var enemySpawned = Instantiate(enemyPrefab,
55                                         new Vector3(spawnPos.x, 0.5f,spawnPos.y),
56                                         Quaternion.identity);
57
58      }
59
        0 references
60      void OnDrawGizmos(){
61
62          for (int i = 0; i < points.Count; i++) {
63              Vector3 cellPosition =  new Vector3(points[i].x, 0f,points[i].y);
64
65              if (Vector3.Distance(cellPosition, player.gameObject.transform.position) > 4){
66                  Gizmos.DrawSphere(cellPosition, 0.25f);
67                  Gizmos.DrawWireCube(cellPosition, Vector3.one);
68              }
69          }
70      }
71  }
72
73
```

GameController.cs

GameController.cs | EnemySpawner.cs

Assets > Labs > Lab7 > Scripts > GameController.cs > GameController

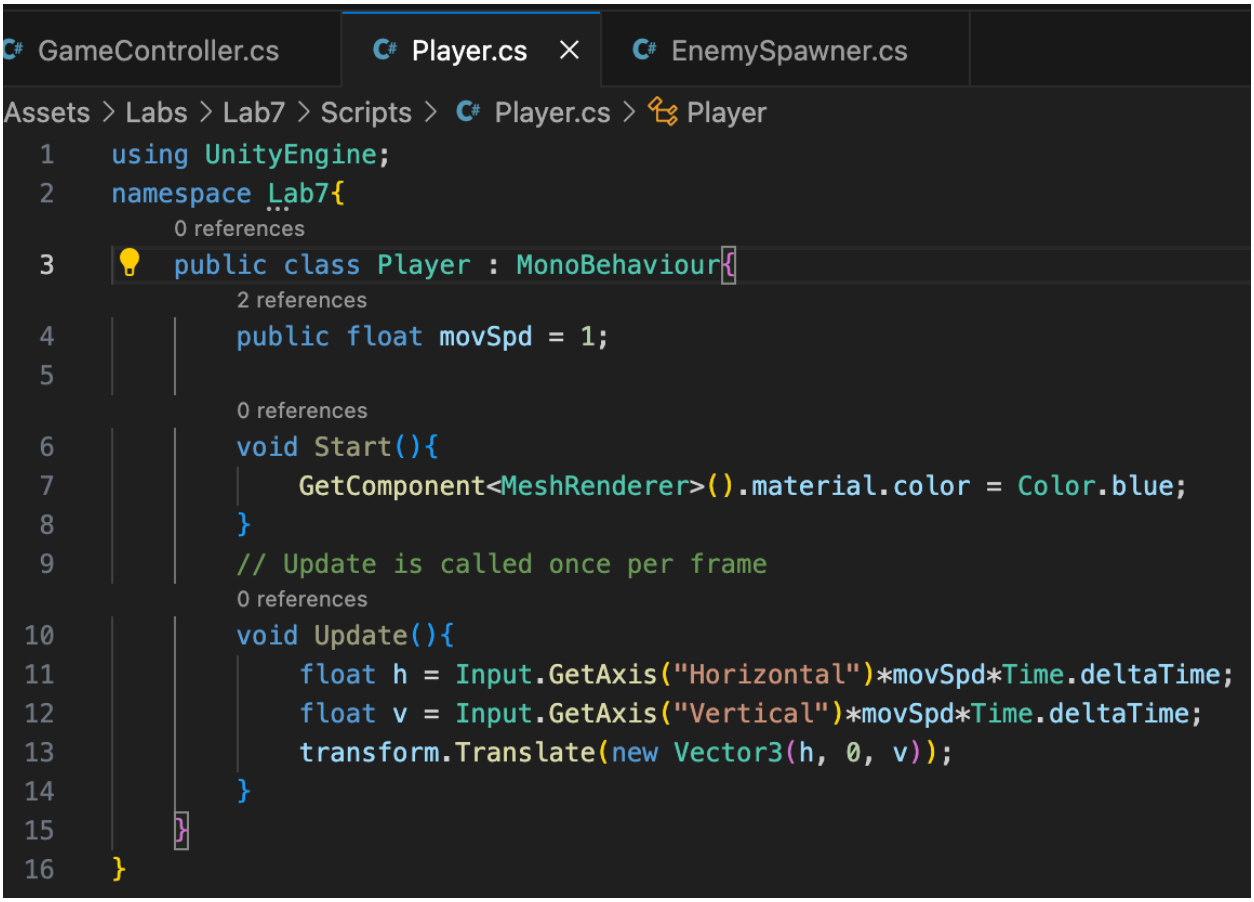
```
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

2 references
public class GameController : MonoBehaviour
{
    6 references
    RaycastHit hit;
    1 reference
    public Text ScoreText;
    1 reference
    public Text HighScoreText;
    2 references
    public Text CountDownText;
    3 references
    public int Counter = 3;
    2 references
    public GameObject gameOverPanel;

    // Start is called before the first frame update
    0 references
    void Start()
    {
        gameOverPanel.SetActive(false);
        InvokeRepeating(nameof(CountDown),0f, 1f);
        HighScoreText.text = "HighScore: " + ScoreManager.HighScore;
    }

    0 references
    void Update(){

        ScoreText.text = "Score: " + ScoreManager.score;
        if (Input.GetMouseButton(0)){ //Total 7
            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
            Debug.DrawRay(ray.origin, ray.direction);
            if (Physics.Raycast(ray, out hit)){

            //Debug.Log($"RAY{hit.transform.name}");
                if (hit.transform.gameObject.CompareTag("Enemy")){
                    hit.transform.GetComponent<FollowerEnemy>().StopEnemy();
                    Destroy(hit.transform.GetComponent<Rigidbody>());
                    Destroy(hit.transform.GetComponent<BoxCollider>());
                    Destroy(hit.transform.gameObject, 1f);
```

GameController.cs | EnemySpawner.cs

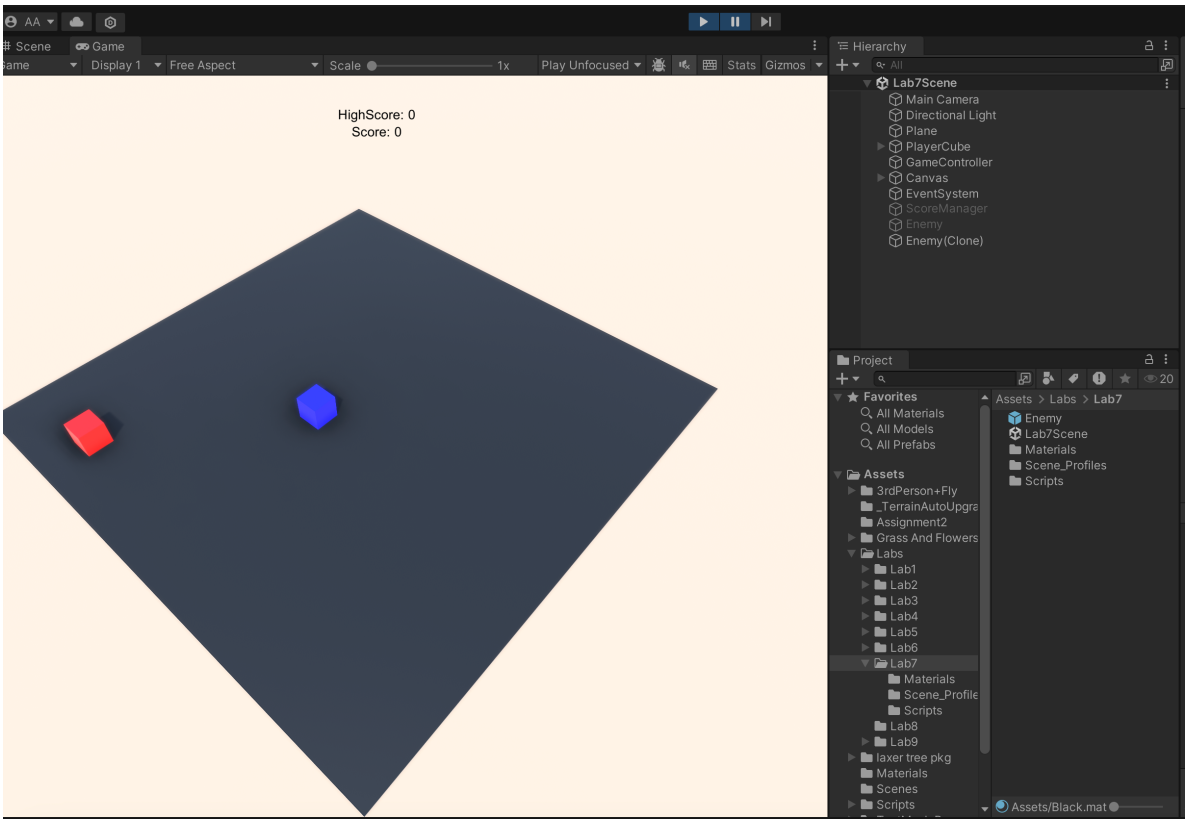
Assets > Labs > Lab7 > Scripts > GameController.cs > GameController > Update

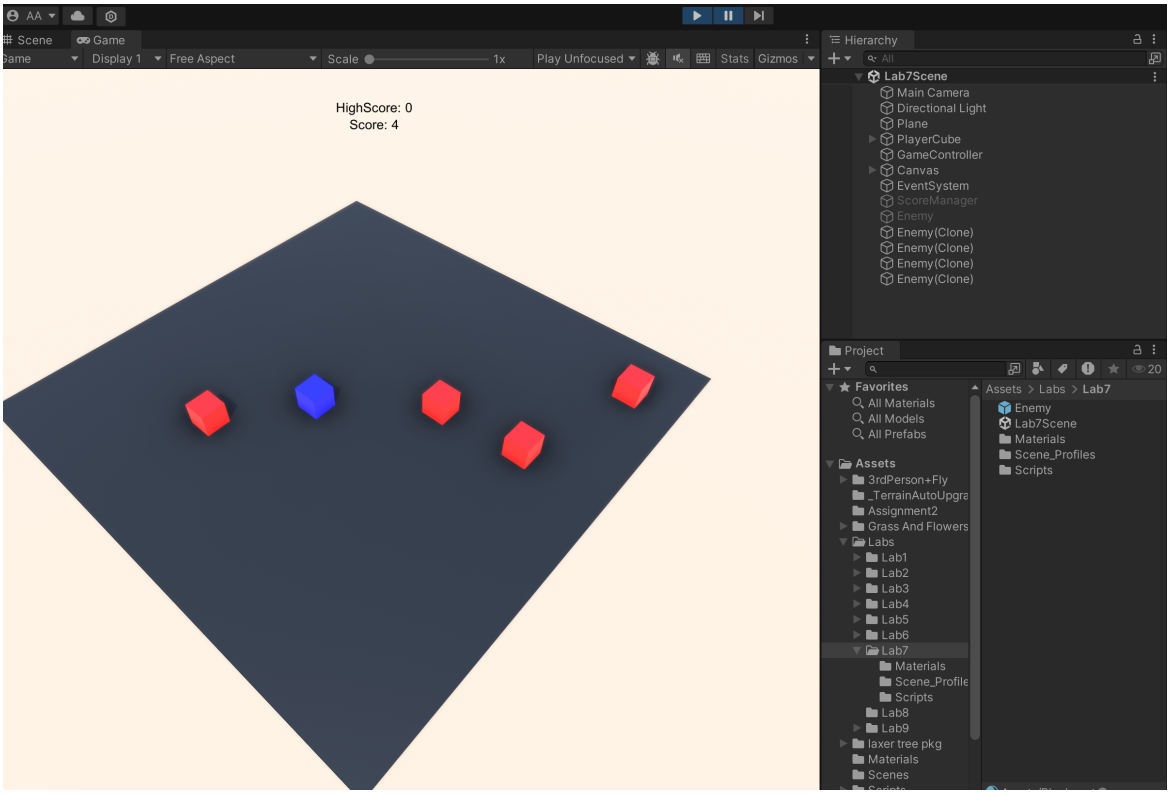
```
5   public class GameController : MonoBehaviour
22      void Update(){
36                      ScoreManager.score++;
37                  }
38              }
39          }
40      }
        1 reference
41      public void gameOver(){
42          gameOverPanel.SetActive(true);
43          Time.timeScale = 0;
44          ScoreManager.SetHighScore();
45      }
46
        0 references
47      public void gameUnPause(){
48          Time.timeScale = 1;
49      }
        2 references
50      public void CountDown(){
51          CountDownText.text = Counter.ToString();
52          Counter--;
53          if (Counter < 0){
54              CancelInvoke(nameof(CountDown));
55              CountDownComplete();
56          }
57      }
        1 reference
58      public void CountDownComplete(){
59          CountDownText.gameObject.SetActive(false);
60      }
61
        0 references
62      public void RestartGame(){
63          Time.timeScale = 1;
64          SceneManager.LoadScene(SceneManager.GetActiveScene().name);
65      }
        0 references
66      public void ExitGame(){
67          Time.timeScale = 1;
68          SceneManager.LoadScene(0);
69      }
70  }
```

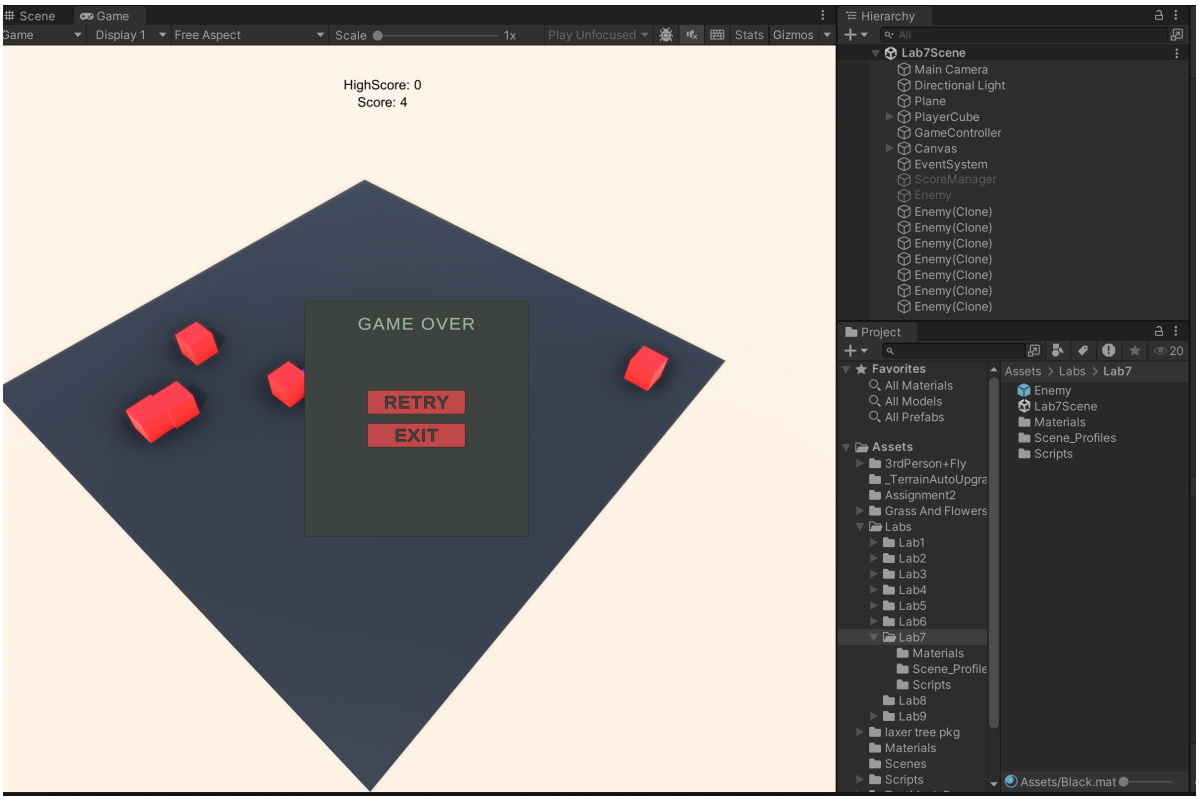
Player.cs

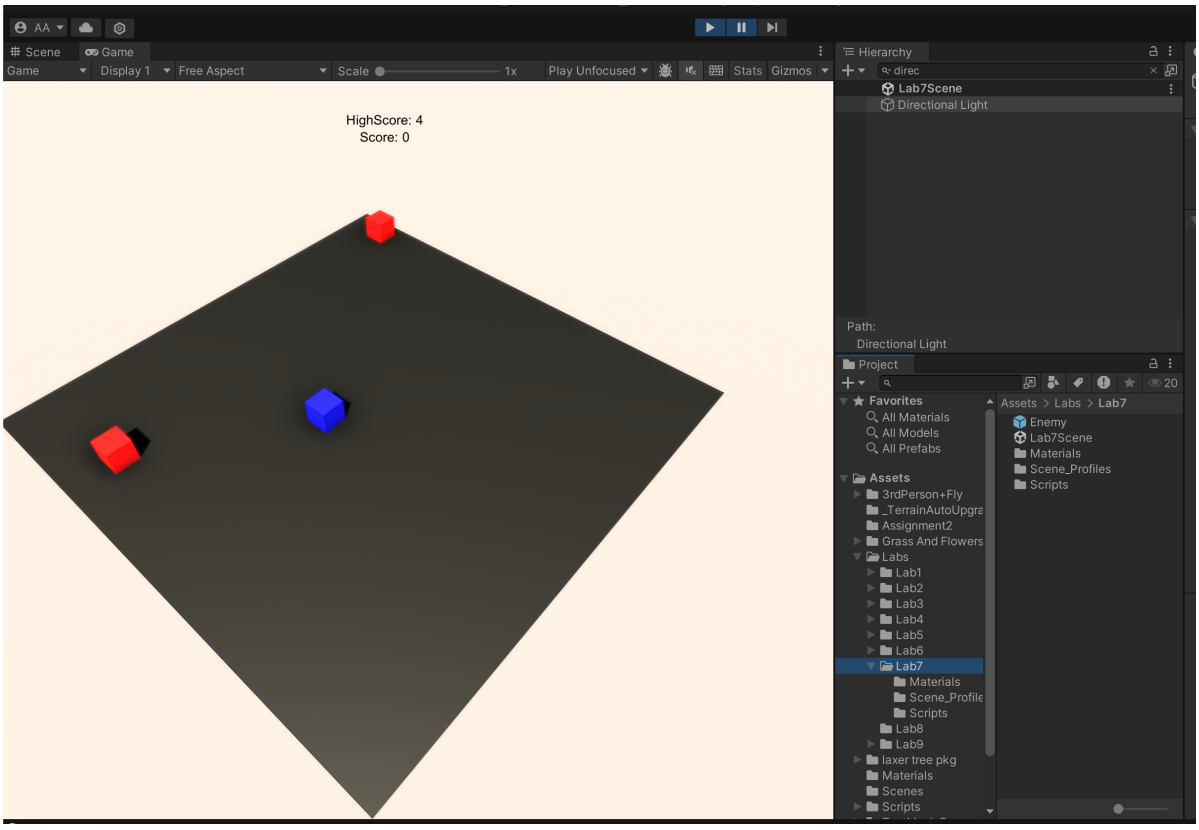

GameController.cs
Player.cs
EnemySpawner.cs
Assets > Labs > Lab7 > Scripts > Player.cs > Player
using UnityEngine;
namespace Lab7{
0 references
public class Player : MonoBehaviour{
2 references
public float movSpd = 1;
0 references
void Start(){
GetComponent<MeshRenderer>().material.color = Color.blue;
}
// 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));
}
}
}

Output:


HighScore: 0
Score: 0
Hierarchy
Lab7Scene
Main Camera
Directional Light
Plane
PlayerCube
GameController
Canvas
EventSystem
ScoreManager
Enemy
Enemy(Clone)
Project
Favorites
All Materials
All Models
All Prefabs
Assets
Assets > Labs > Lab7
Enemy
Lab7Scene
Materials
Scene_Profiles
Scripts
Assets/Black.mat


HighScore: 0
Score: 4
Hierarchy
Lab7Scene
Main Camera
Directional Light
Plane
PlayerCube
GameController
Canvas
EventSystem
ScoreManager
Enemy
Enemy(Clone)
Enemy(Clone)
Enemy(Clone)
Enemy(Clone)
Project
Favorites
All Materials
All Models
All Prefabs
Assets
Assets > Labs > Lab7
Enemy
Lab7Scene
Materials
Scene_Profiles
Scripts


Scene
Game
Display 1
Free Aspect
Scale
1x
Play Unfocused
Stats
Gizmos
HighScore: 0
Score: 4
GAME OVER
RETRY
EXIT
Hierarchy
Lab7Scene
Main Camera
Directional Light
Plane
PlayerCube
GameController
Canvas
EventSystem
ScoreManager
Enemy
Enemy(Clone)
Enemy(Clone)
Enemy(Clone)
Enemy(Clone)
Enemy(Clone)
Enemy(Clone)
Enemy(Clone)
Project
Favorites
All Materials
All Models
All Prefabs
Assets
3rdPerson+Fly
_TerrainAutoUpgra
Assignment2
Grass And Flowers
Labs
Lab1
Lab2
Lab3
Lab4
Lab5
Lab6
Lab7
Materials
Scene_Profile
Scripts
Lab8
Lab9
laxer tree pkg
Materials
Scenes
Scripts
Assets > Labs > Lab7
Enemy
Lab7Scene
Materials
Scene_Profiles
Scripts
Assets/Black.mat


AA
Scene
Game
Display 1
Free Aspect
Scale
1x
Play Unfocused
Stats
Gizmos
HighScore: 4
Score: 0
Hierarchy
direc
Lab7Scene
Directional Light
Path:
Directional Light
Project
Favorites
All Materials
All Models
All Prefabs
Assets
3rdPerson+Fly
_TerrainAutoUpgra
Assignment2
Grass And Flowers
Labs
Lab1
Lab2
Lab3
Lab4
Lab5
Lab6
Lab7
Materials
Scene_Profile
Scripts
Lab8
Lab9
laxer tree pkg
Materials
Scenes
Scripts
Assets > Labs > Lab7
Enemy
Lab7Scene
Materials
Scene_Profiles
Scripts