Lab # 07



Fall 2024

**CSE-411L Intro to Game Development Lab**

Submitted by: MUHAMMAD SADEEQ

Registration No.: 21PWCSE2028

Section: C

“On my honor, as a student of the University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work”

Submitted to:

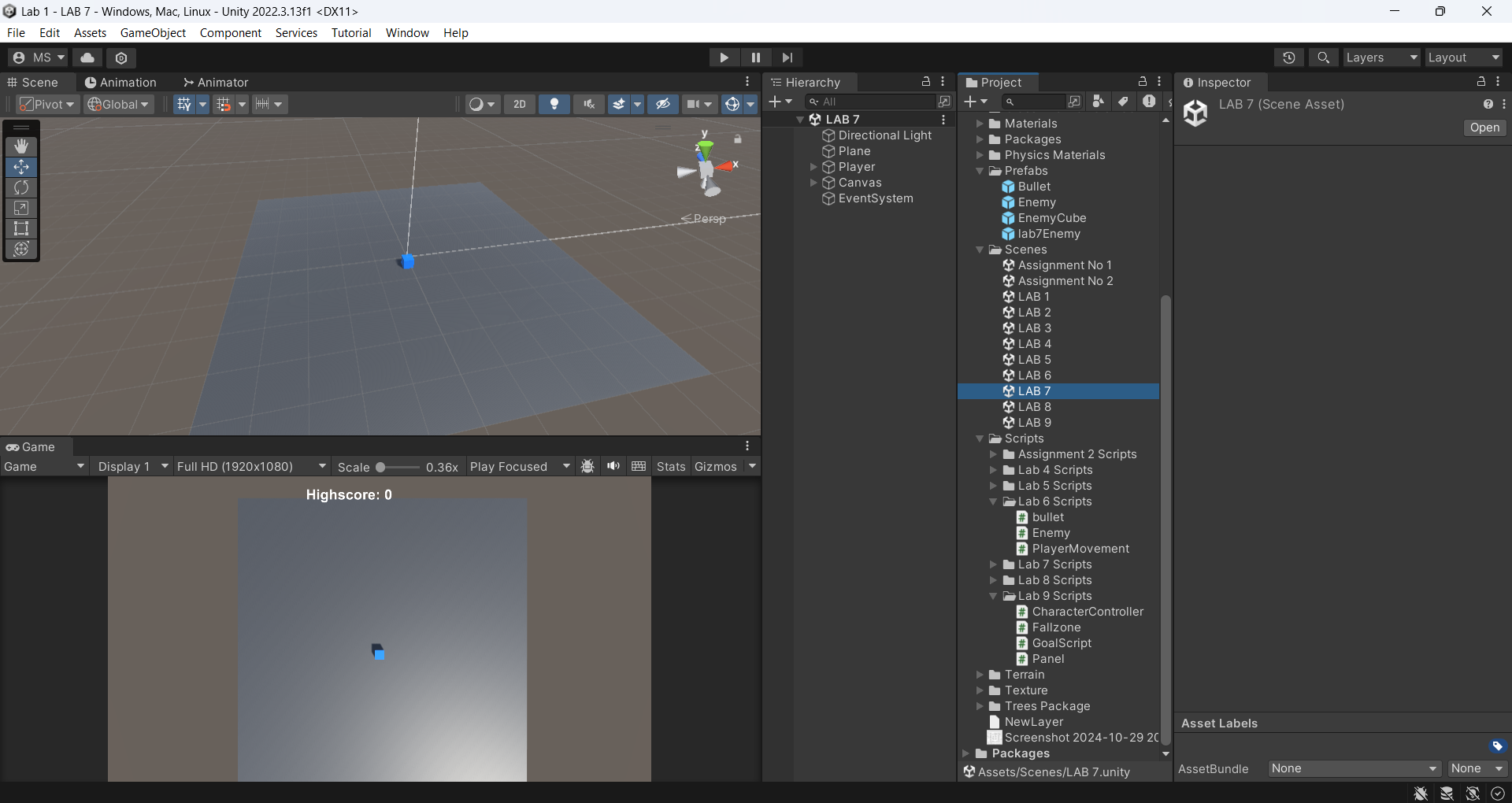
Engr. Abdullah Hamid

(15 Dec 2024)

Department of Computer systems engineering

University of Engineering and Technology, Peshawar

1. **Created a Unity Scene:**
   1. Opened Unity and created a new scene.



1. **Player Setup:**
   1. Added a cube to represent the player.
   2. Implemented movement for the player cube using Unity's Input.GetAxis() for horizontal and vertical axis inputs, allowing the player to move:
   3. Forward and backward.
   4. Left and right.

**Player Controller Code:**

using System.Collections;

using System.Collections.Generic;

using Unity.VisualScripting;

using UnityEditor;

using UnityEngine;

using UnityEngine.UI;

public class PlayerController : MonoBehaviour

{

public float moveX, moveZ;

public float speed = 10;

float centerToEdge = 48f;

[SerializeField]

private GameObject enemyCube;

[SerializeField]

private GameObject GameOverPanel;

RaycastHit hit;

int scoreNumb = 0;

public Text scoreText;

public Text Highscore;

// Start is called before the first frame update

private void Start()

{

if(Time.timeScale == 0) { Time.timeScale = 1; }

SetColor(gameObject, Color.blue);

InvokeRepeating("EnemySpawn", 3f, 3f);

}

// Update is called once per frame

void Update()

{

float moveX = Input.GetAxis("Horizontal");

float moveZ = Input.GetAxis("Vertical");

Vector3 move = new Vector3(moveX, 0, moveZ).normalized;

transform.Translate(move \* speed \* Time.deltaTime, Space.World);

scoreText.text = $"Score : {scoreNumb}";

if (Input.GetMouseButtonDown(0))

{

Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);

if (Physics.Raycast(ray, out hit))

{

if (hit.transform.gameObject.CompareTag("Enemy"))

{

scoreNumb++;

Destroy(hit.collider.gameObject,1f);

hit.collider.gameObject.GetComponent<EnemyController>().isstopped = true;

}

}

}

if (Time.timeScale == 0)

{

ScoreManager.GameOver(scoreNumb);

Highscore.text = ScoreManager.highscore.ToString();

GameOverPanel.SetActive(true);

}

}

void EnemySpawn()

{

Vector3 randomLocationOnPlane = new Vector3(Random.Range(-centerToEdge, centerToEdge), 1.5f, Random.Range(-centerToEdge, centerToEdge));

Instantiate(enemyCube, randomLocationOnPlane, Quaternion.identity);

}

private void SetColor(GameObject obj, Color color)

{

Renderer renderer = obj.GetComponent<Renderer>();

if (renderer != null)

{

renderer.material.color = color;

}

}

}

1. **Camera Setup:**
   1. Configured a camera to provide a top-down (bird's-eye) view of the game area.
2. **Enemy Instantiation:**
   1. Set up functionality to randomly instantiate enemy cubes on the plane after every 3-second interval.
   2. Ensured that enemies spawn only within the boundaries of the plane.
3. **Initialized Game Objects:**
   1. At the start of the game:
   2. Set the player cube's color to blue.
   3. Set all enemy cubes' colors to red.
   4. Programmed the enemies to move towards the player upon spawning.

**Enemy Controller Code:**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class EnemyController : MonoBehaviour

{

float enemySpeed = 5f;

[SerializeField] GameObject Player;

public bool isstopped = false;

// Start is called before the first frame update

void Start()

{

Player = GameObject.FindWithTag("Player");

SetColor(gameObject, Color.red);

}

private void Update()

{

if (!isstopped)

{

MoveEnemyTowardsPlayer(gameObject);

}

}

private void MoveEnemyTowardsPlayer(GameObject enemy)

{

Vector3 direction = (Player.transform.position - enemy.transform.position).normalized;

enemy.transform.position += direction \* enemySpeed \* Time.deltaTime;

}

private void SetColor(GameObject obj, Color color)

{

Renderer renderer = obj.GetComponent<Renderer>();

if (renderer != null)

{

renderer.material.color = color;

}

}

private void OnCollisionEnter(Collision collision)

{

if (collision.gameObject.CompareTag("Player"))

{

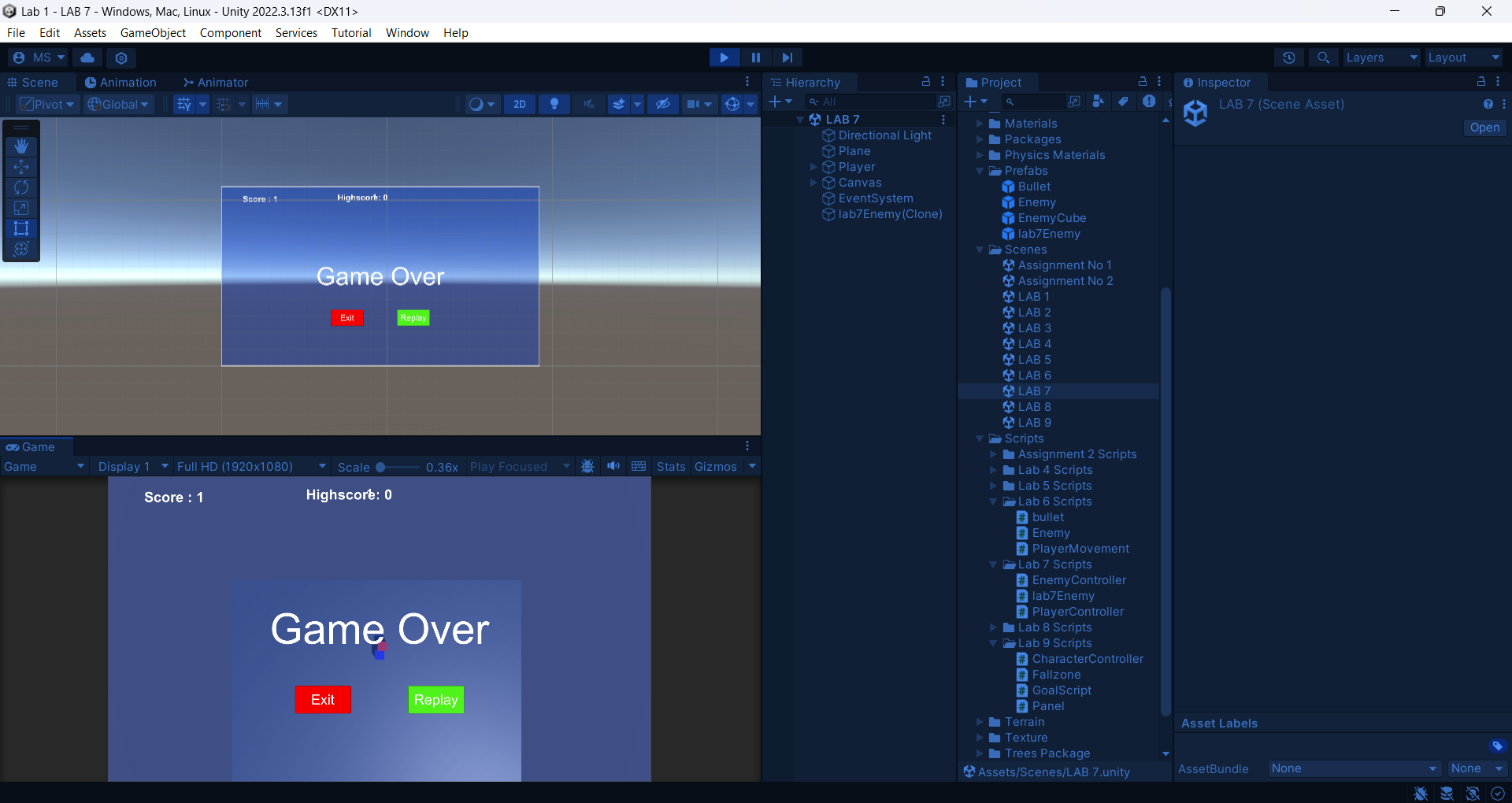
Time.timeScale = 0f;

}

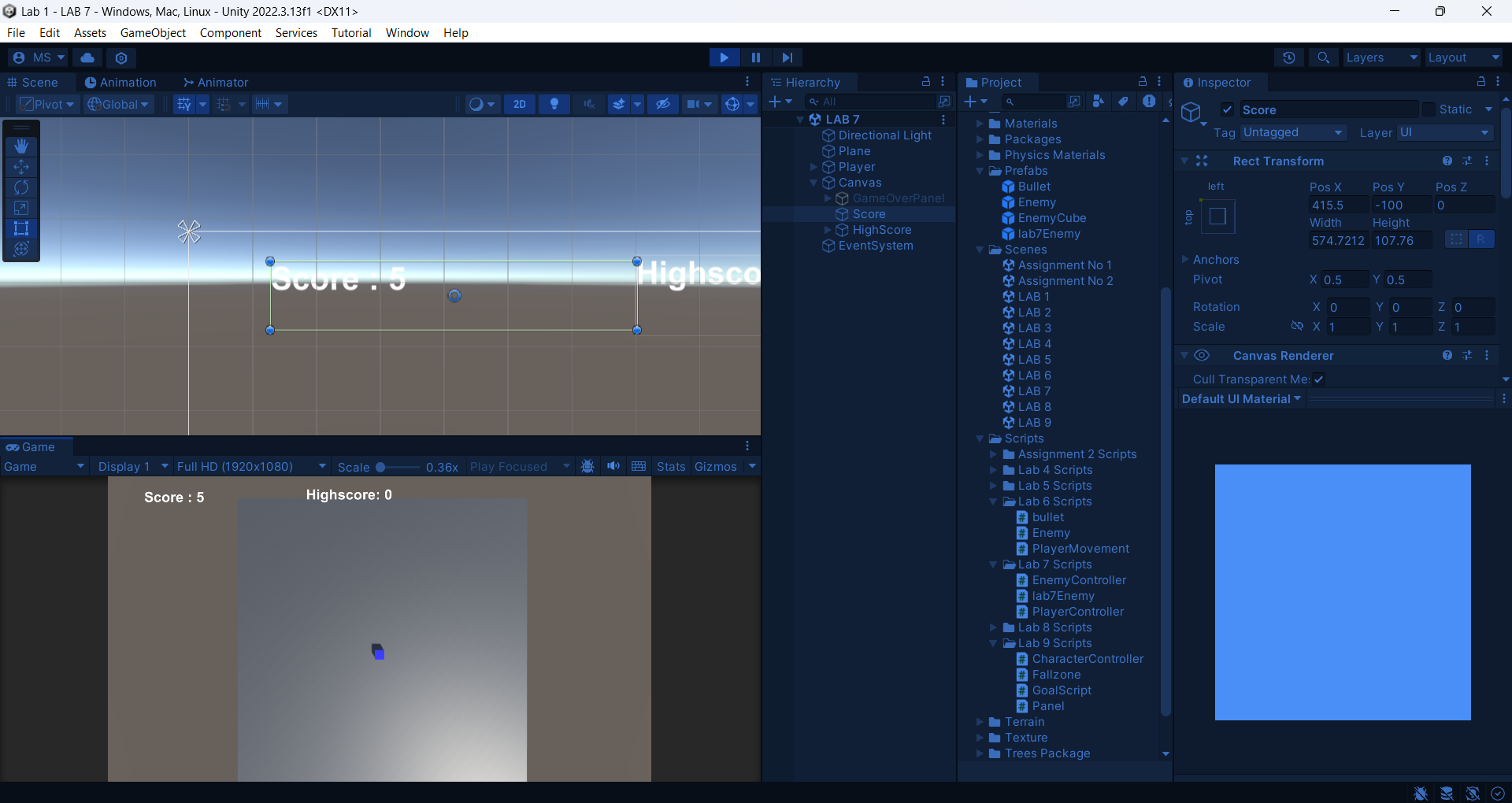
}

}

1. **Game Over Condition:**
   1. Implemented the following behavior for when an enemy collides with the player:
   2. The game pauses.
   3. A "Game Over" message is displayed on the screen.



1. **Enemy Interaction:**
   1. Programmed the enemies to react when clicked:
   2. The movement of the clicked enemy stops.
   3. The enemy is destroyed after a 1-second delay.
2. **Scoring System:**
   1. Added a score system that displays the current score on the screen.
   2. The score increases each time an enemy is destroyed by the player.

****