Lab # 06

Unity API (Input Axis, Raycast and Instantiation Methods)



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:

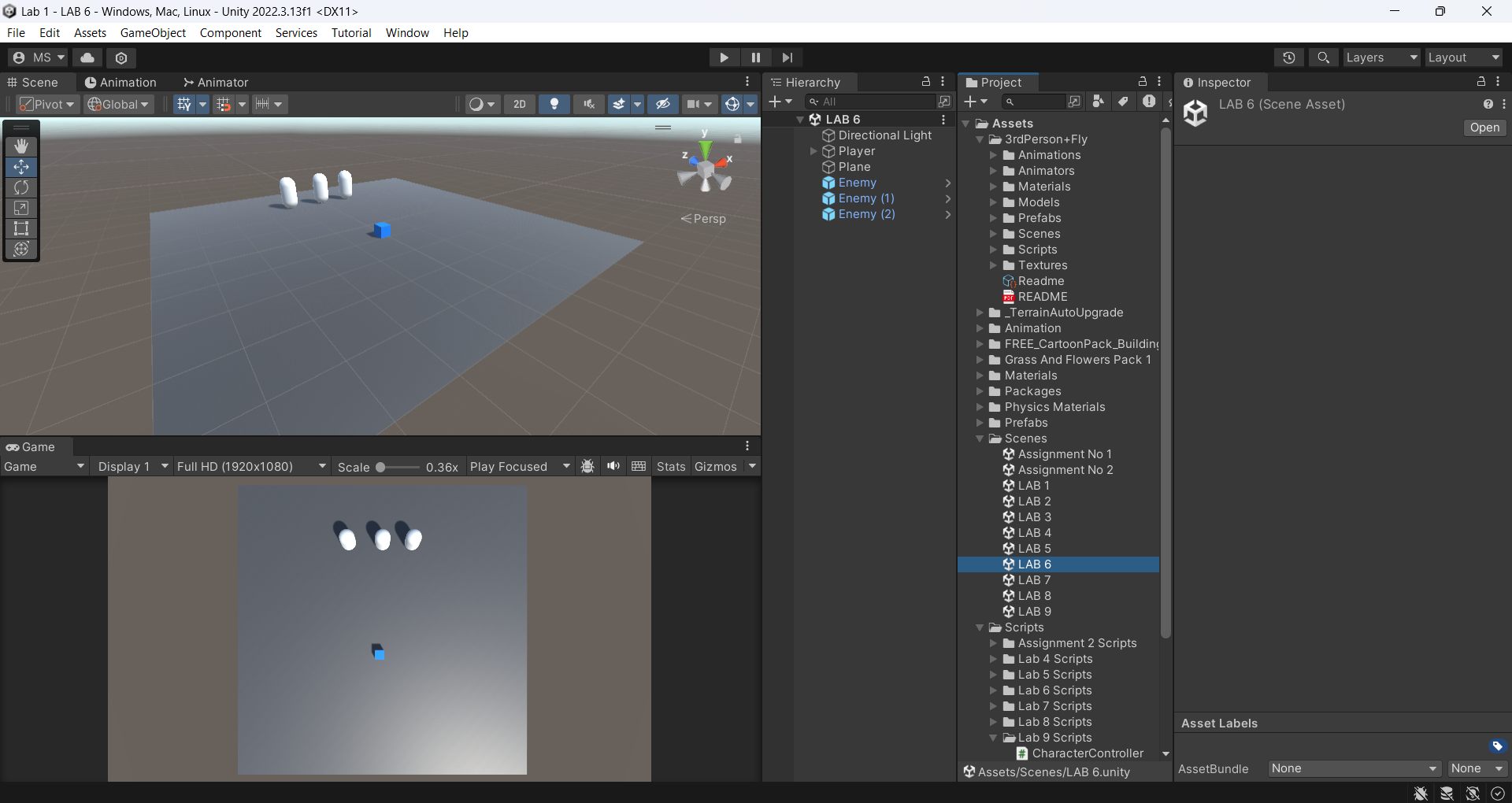
Engr. Abdullah Hamid

(15 Dec 2024)

Department of Computer systems engineering

University of Engineering and Technology, Peshawar

1. **Created a Unity Scene:**
   1. Opened a new Unity project and created a 3D scene.
   2. Added essential GameObjects to the scene, such as a plane to act as the game area.



1. **Player Setup:**
   1. Created a cube to represent the player and added a Rigidbody component for physics.
   2. Implemented movement for the player using Unity's Input system with the following controls:
   3. Forward/Backward: Controlled by the Vertical Axis (W/S or Up/Down Arrow keys).
   4. Left/Right: Controlled by the Horizontal Axis (A/D or Left/Right Arrow keys).

**Player Movement Code:**

using System.Collections;

using System.Collections.Generic;

using UnityEditor;

using UnityEngine;

public class PlayerMovement : MonoBehaviour

{

[SerializeField]

private float speed = 1f;

[SerializeField]

GameObject bullet;

// Update is called once per frame

void Update()

{

if (Input.GetKey(KeyCode.W))

{

transform.Translate(Vector3.forward \* speed \* Time.deltaTime);

}

if (Input.GetKey(KeyCode.S))

{

transform.Translate(Vector3.back \* speed \* Time.deltaTime);

}

if (Input.GetKey(KeyCode.A))

{

transform.Translate(Vector3.left \* speed \* Time.deltaTime);

}

if (Input.GetKey(KeyCode.D))

{

transform.Translate(Vector3.right \* speed \* Time.deltaTime);

}

if ((Input.GetKeyDown(KeyCode.Space)))

{

Instantiate(bullet,transform.position,Quaternion.identity);

}

}

}

1. **Camera Setup:**
   1. Positioned the camera to provide a top-down view of the game area.
   2. Configured the camera to follow the player's movements for better visibility.
2. **Shooting Mechanism:**
   1. Added functionality for the player to shoot bullets by pressing Spacebar
   2. Bullets:
   3. Spawn at the player’s position with a forward force.
   4. Automatically destroy themselves after 2 seconds to optimize performance.

**Bullet Code:**

using UnityEngine;

public class bullet : MonoBehaviour

{

Rigidbody rb;

float bulletSpeed = 10;

// Start is called before the first frame update

void Start()

{

rb = GetComponent<Rigidbody>();

}

// Update is called once per frame

void Update()

{

rb.AddForce(Vector3.forward \* bulletSpeed);

Destroy(gameObject, 2);

}

}

1. **Enemy Setup:**
   1. Placed 3 enemy cubes on the plane as target objects.
   2. Implemented the following behavior for enemies upon being hit by a bullet:
   3. Turn yellow immediately after being hit.
   4. After a 1-second delay, turn red.
   5. Finally, destroy themselves with another 1-second delay.
2. **Console Messages:**
   1. Created an array containing three strings:
   2. "Dead!", "Killed!", "Defeated!".
   3. Implemented a random selection mechanism to print one of these messages in the Unity console whenever an enemy is destroyed.

**Enemy Code:**

using UnityEngine;

public class Enemy : MonoBehaviour

{

string[] message = { "Dead!", "Killed!", "Defeated!" };

private void OnTriggerEnter(Collider other)

{

if (other.gameObject.CompareTag("bullet"))

{

gameObject.GetComponent<MeshRenderer>().material.color = Color.yellow;

Invoke("ChangeColor", .5f);

Destroy(gameObject,1);

Debug.Log(message[Random.Range(0,message.Length)]);

}

}

private void ChangeColor()

{

gameObject.GetComponent<MeshRenderer>().material.color = Color.red;

}

}

