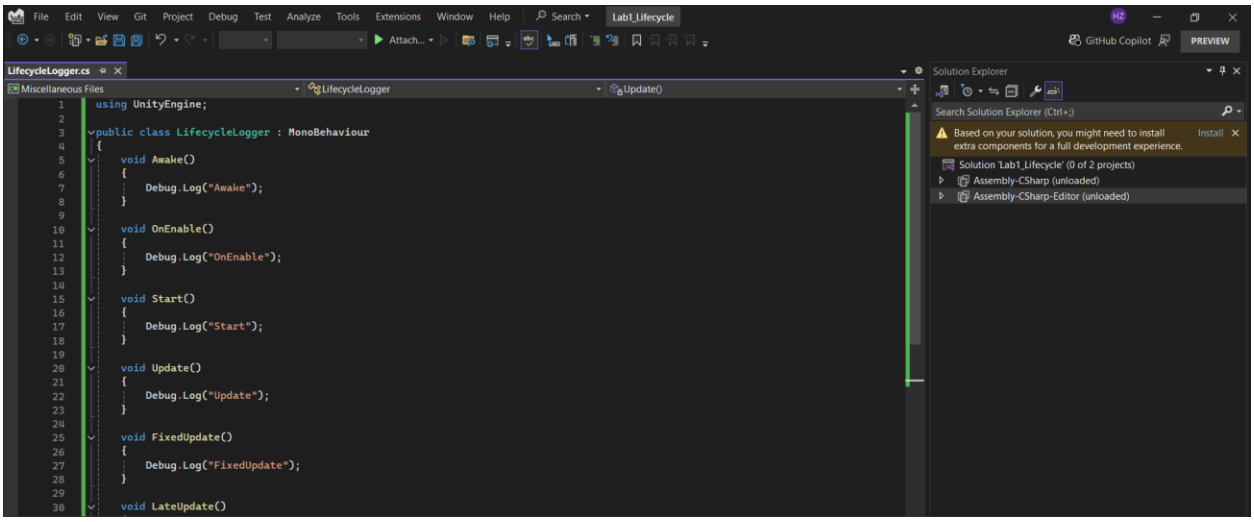
LAB THỰC HÀNH – CHƯƠNG 3: UNITY SCRIPTING

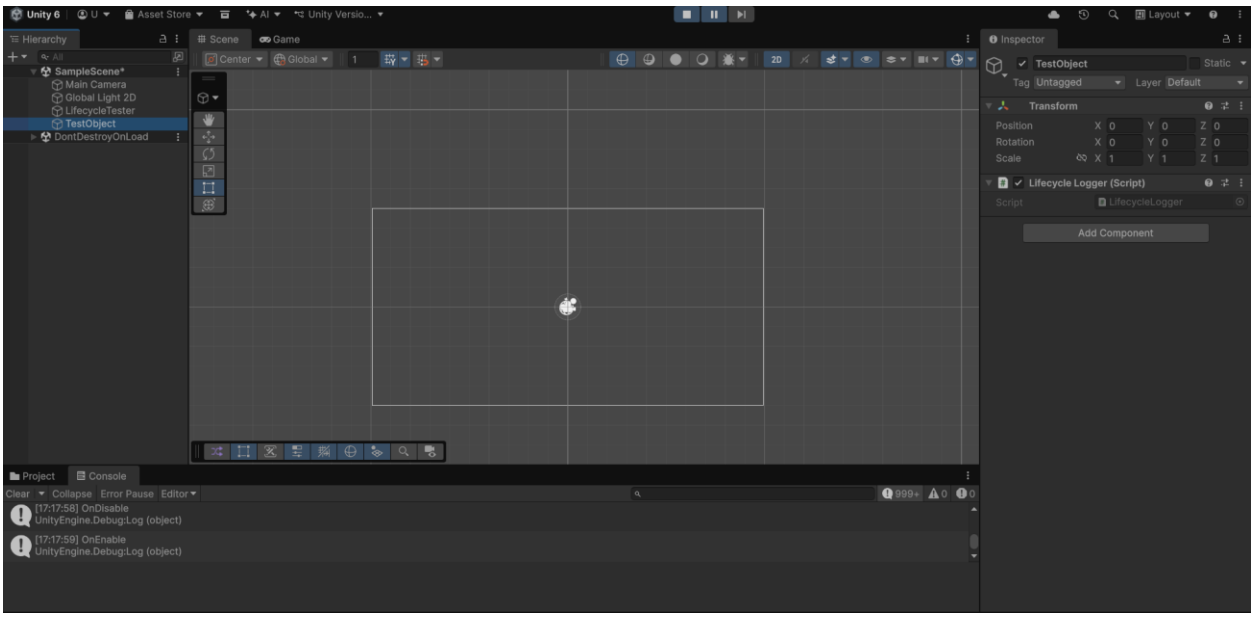
Lab 1 – Component Lifecycle Debugger

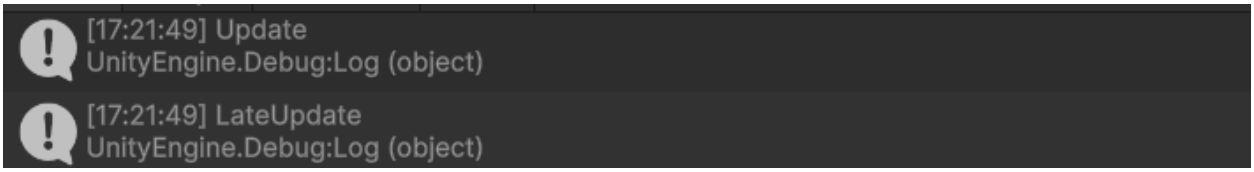
Tạo script



Toggle Active:

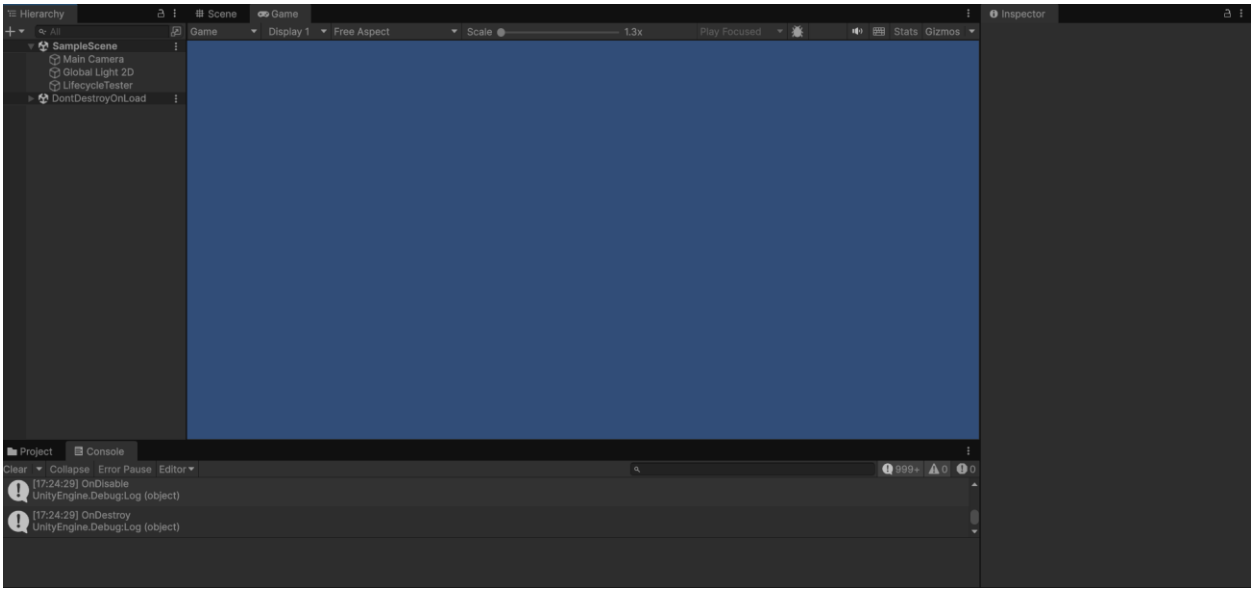
Click **TestObject** trong **Hierarchy**





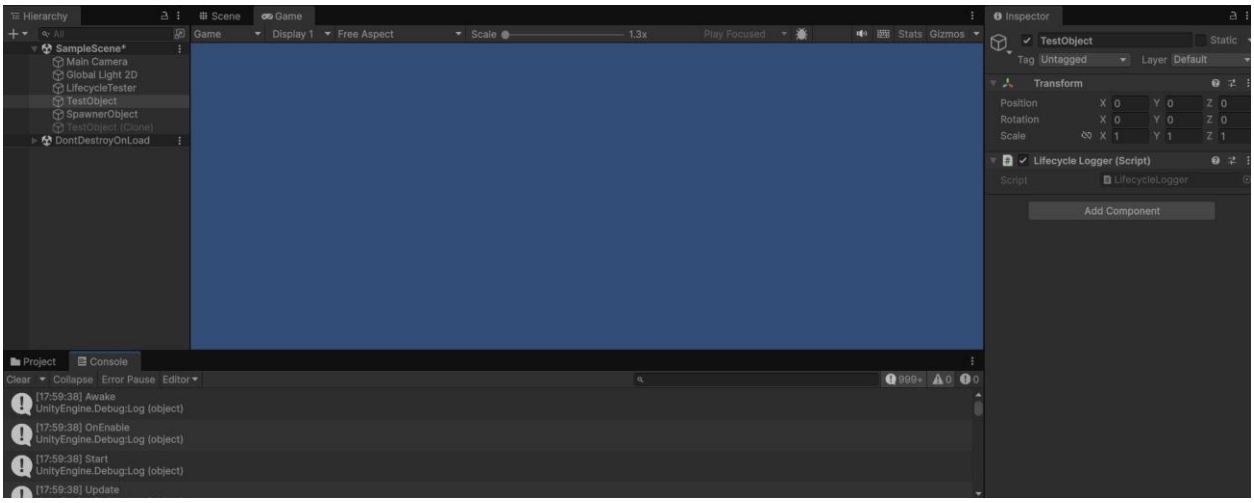
DESTROY OBJECT:

- Chọn **TestObject** trong Hierarchy chọn **Delete**



INSTANTIATE OBJECT:

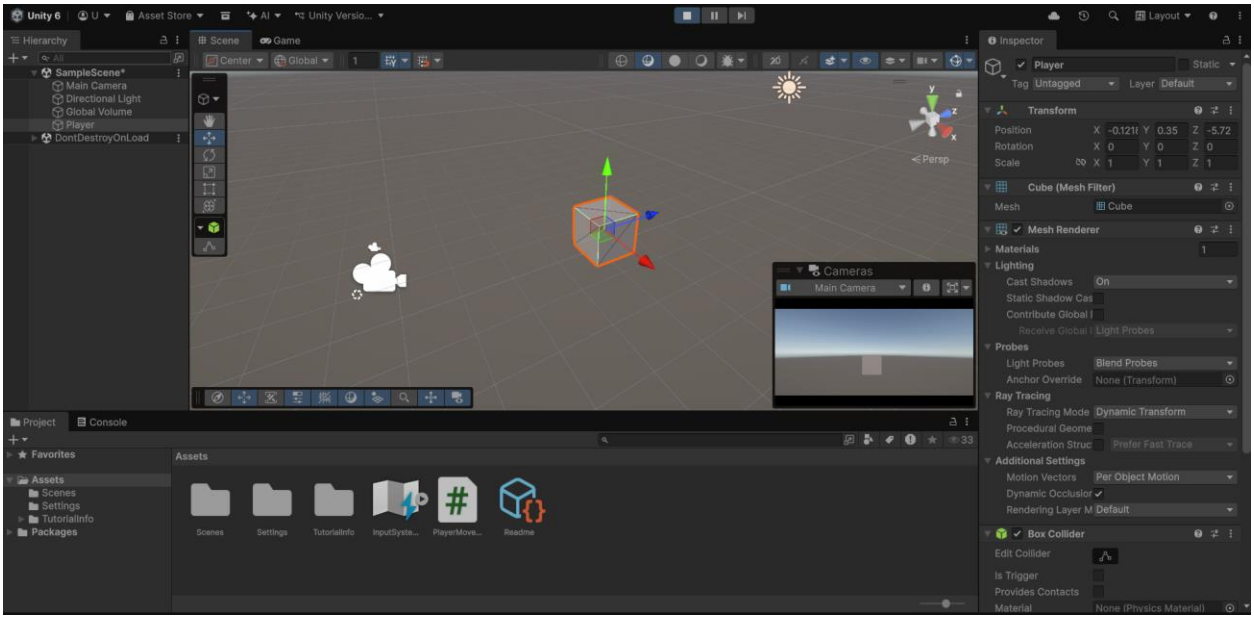
Tạo SpawnerObject và kéo **TestObject (Prefab)** vào ô Prefab



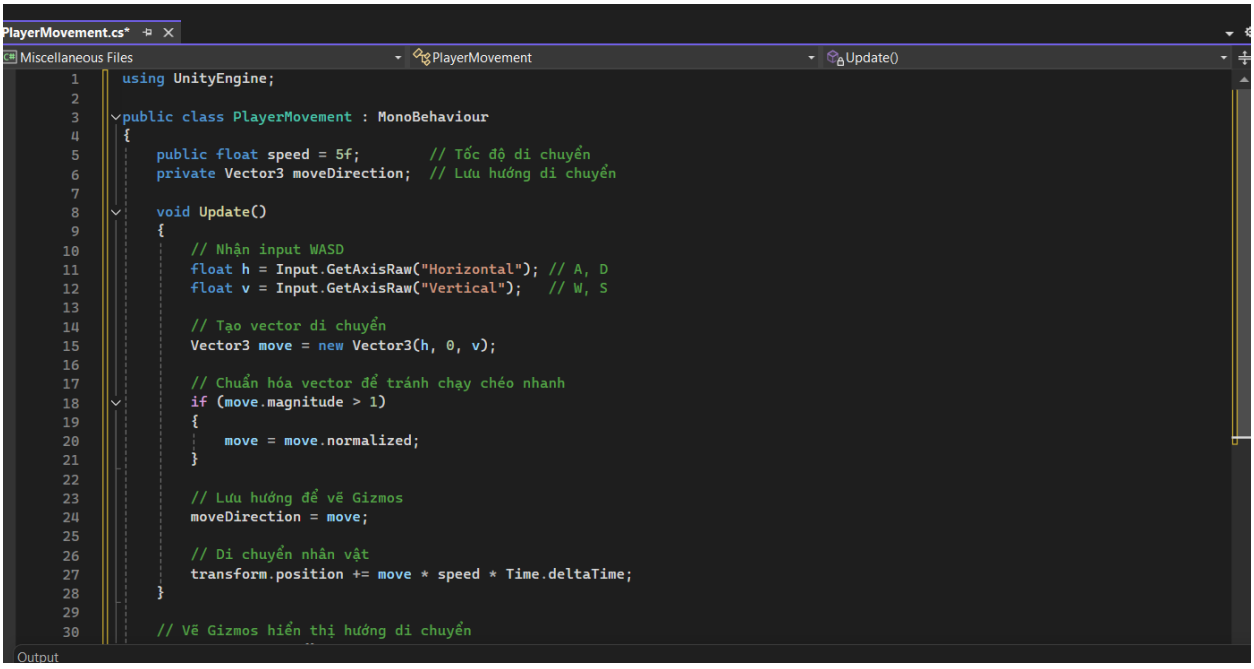
Lab 2 – Vector Movement & Gizmos

Mục tiêu :Điều khiển nhân vật bằng WASD, chuẩn hóa vector để tránh chạy chéo nhanh:

Tạo Player



Tạo PlayerMovement



```
using UnityEngine;

public class PlayerMovement : MonoBehaviour
{
    public float speed = 5f;        // Tốc độ di chuyển
    private Vector3 moveDirection;  // Lưu hướng di chuyển

    void Update()
    {
        // Nhận input WASD
        float h = Input.GetAxisRaw("Horizontal"); // A, D
        float v = Input.GetAxisRaw("Vertical");   // W, S

        // Tạo vector di chuyển
        Vector3 move = new Vector3(h, 0, v);

        // Chuẩn hóa vector để tránh chạy chéo nhanh
        if (move.magnitude > 1)
        {
            move = move.normalized;
        }

        // Lưu hướng để vẽ Gizmos
        moveDirection = move;

        // Di chuyển nhân vật
        transform.position += move * speed * Time.deltaTime;
    }

    // Vẽ Gizmos hiển thị hướng di chuyển
```

Đã điều khiển nhân vật bằng WASD , chuẩn hóa vector để tránh chạy chéo nhanh và Gizmos hiển thị hướng di chuyển.

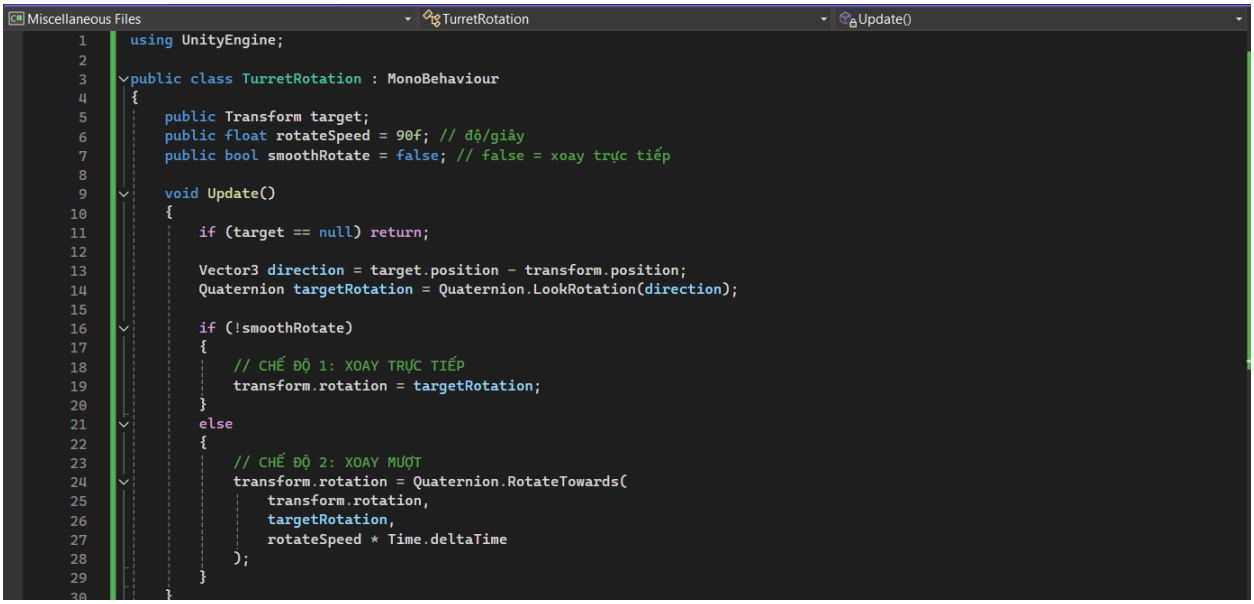
Normalize**:** là đưa vector về độ dài bằng **1** nhưng **giữ nguyên hướng**, giúp nhân vật di chuyển **cùng tốc độ** dù đi thẳng hay đi chéo, tránh trường hợp nhấn nhiều phím làm chạy nhanh hơn.

Lab 3 – Quaternion Rotation

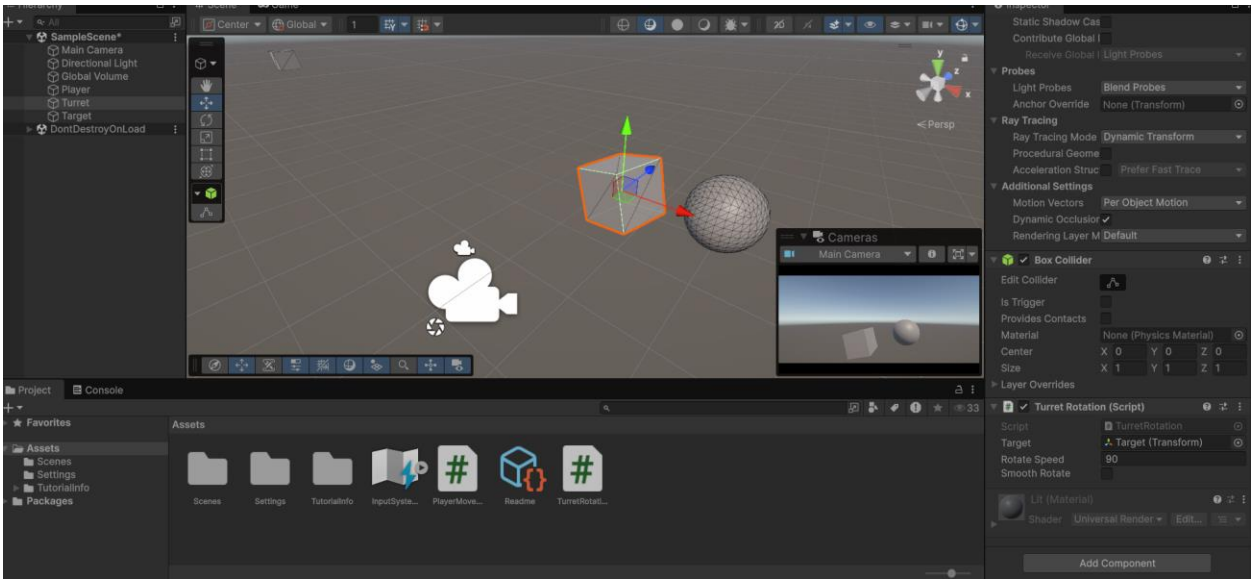
Tạo **Cube** → rename: Turret

Tạo **Sphere** → rename: Target

Tạo script: **TurretRotation.cs**



Turret xoay nhìn target bằng LookAt và RotateTowards/Slerp.



20260120-1532-16.1239605.mp4

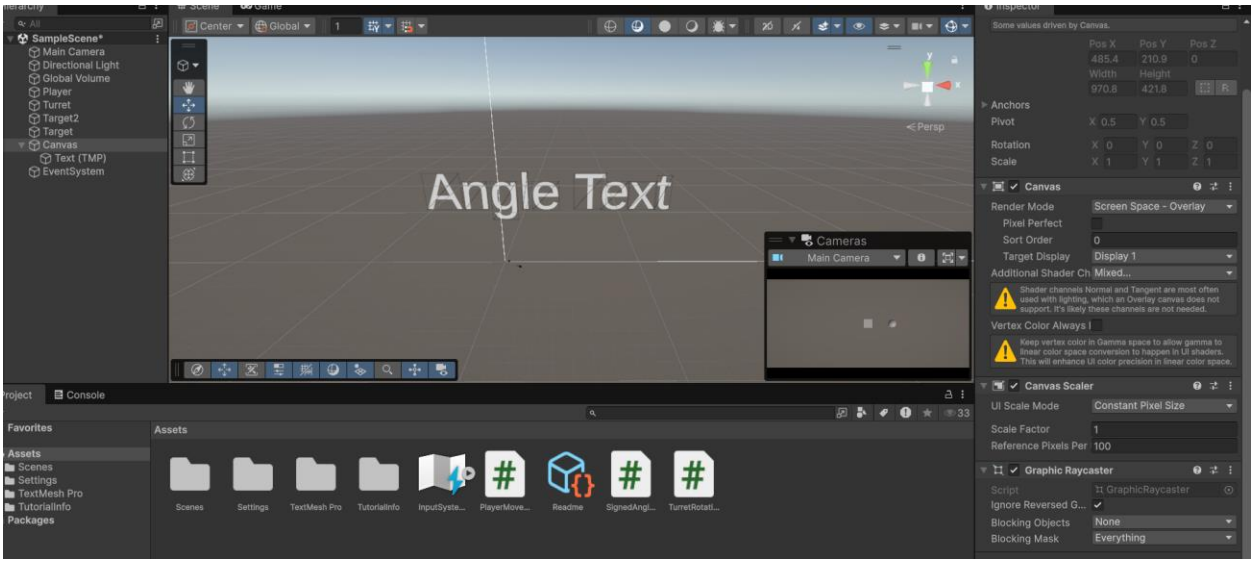
Lab 4 – Signed Angle (2D/Topdown)

Tạo “Target”

Tạo SignedAngleRotate.cs:

```
using UnityEngine;

public class SignedAngleRotate : MonoBehaviour
{
    public Transform target;
    public float currentAngle;

    void Update()
    {
        if (target == null) return;

        // Vector hướng hiện tại của Player (trục Z)
        Vector3 forward = transform.forward;

        // Vector từ Player tới Target
        Vector3 toTarget = target.position - transform.position;

        // Chỉ xét mặt phẳng XZ (topdown)
        forward.y = 0;
        toTarget.y = 0;

        // Tính góc SignedAngle
        currentAngle = Vector3.SignedAngle(forward, toTarget, Vector3.up);

        // Xoay Player theo góc đó
        transform.Rotate(0, currentAngle * Time.deltaTime, 0);
    }
}
```

Chọn **Text (TMP)** → Inspector: Angle: 0



Di chuyển **Target**

Angle: -23.4

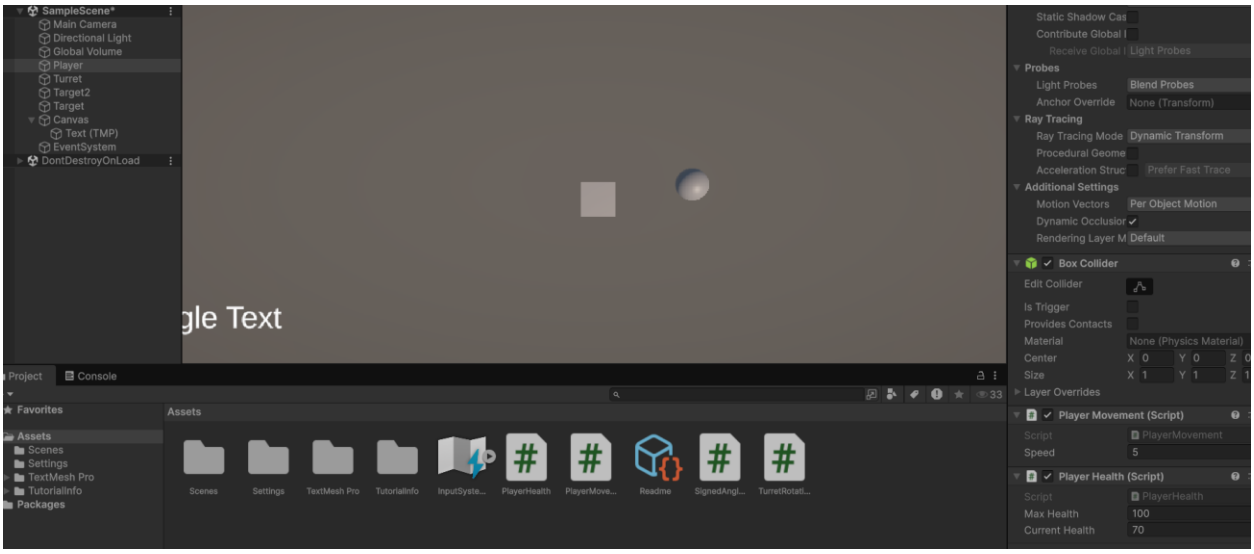
Angle: 45.1

LAB 5 – OBSERVER PATTERN (C# EVENT)

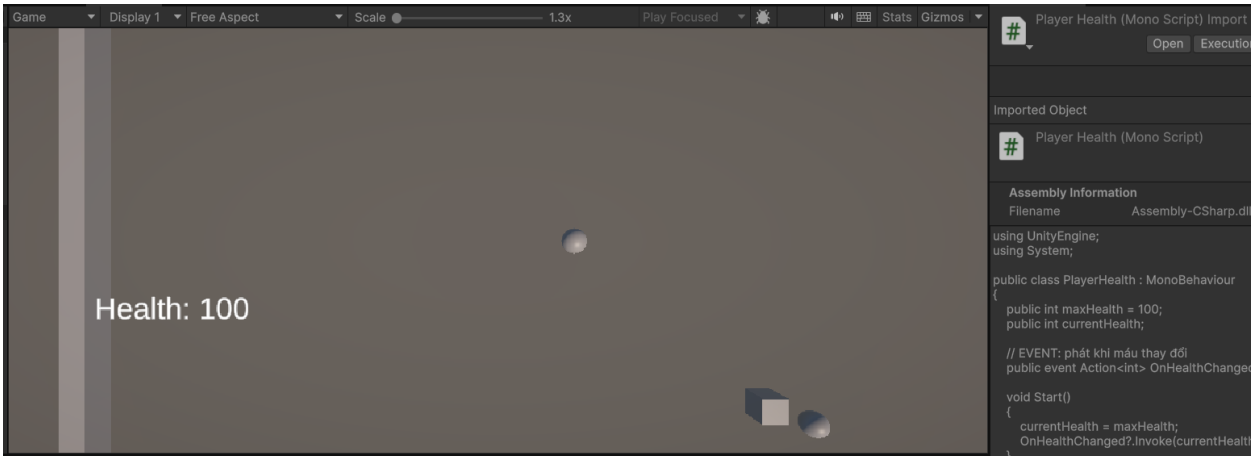
TẠO PLAYER HEALTH (SUBJECT)

Tạo Script PlayerHealth

```
scellaneous Files                    PlayerHealth                    maxHealth
1   using UnityEngine;
2   using System;
3
4   public class PlayerHealth : MonoBehaviour
5   {
6       public int maxHealth = 100;
7       public int currentHealth;
8
9       // EVENT: phát khi máu thay đổi
10      public event Action<int> OnHealthChanged;
11
12      void Start()
13      {
14          currentHealth = maxHealth;
15          OnHealthChanged?.Invoke(currentHealth);
16      }
17
18      void Update()
19      {
20          // Nhấn H để trừ máu
21          if (Input.GetKeyDown(KeyCode.H))
22          {
23              TakeDamage(10);
24          }
25      }
26
27      void TakeDamage(int damage)
28      {
29          currentHealth -= damage;
30          currentHealth = Mathf.Clamp(currentHealth, 0, maxHealth);
```



TEXT HEALTH



Trong Inspector:

- Text: Health
- Font Size: 28
- Auto Size: **OFF**

Tạo GameOver Text và Khi HP = 0:

LAB 6 – OBSERVER PATTERN (UNITYEVENT)

làm giống Lab 5, nhưng KHÔNG dùng event C#, chỉ dùng UnityEvent.

```
using UnityEngine;
using UnityEngine.Events;

public class PlayerHealthUE : MonoBehaviour
{
    public int health = 100;

    public UnityEvent<int> OnHealthChanged;

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.H))
        {
            health -= 10;
            health = Mathf.Clamp(health, 0, 100);

            OnHealthChanged.Invoke(health);
        }
    }
}
```