Title: Falling Down of Objects - Gravity Effect

EXPERIMENT NO: 5 **Name:** SHAKYA JAY JITENDRA

Date:14-8-2025 **Reg. No:** 24BCG10123

Aim: To implement and observe a custom gravity system in Unity using physics forces, replacing the built-in gravity to control the rate at which objects fall.

Description/Concept:

In this Unity experiment, the goal is to create a script that allow object to adjust the fall of gravity
This allows fine-tuning of the gravity value for specific objects, enabling effects such as:
Faster or slower falling.
Gravity in different directions (e.g., upward or sideways for special effects).
Per-object gravity control instead of a global setting.

Program/Coding:

```
using UnityEngine;

public class CustomGravity : MonoBehaviour
{
    public float gravity = -9.81f;
    private Rigidbody rb;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
        if (rb == null)
        {
            rb = gameObject.AddComponent<Rigidbody>();
        }

        // Disable built-in gravity since we are applying it manually
        rb.useGravity = false;
    }

    void FixedUpdate()
    {
        Vector3 gravityForce = new Vector3(0, gravity * rb.mass, 0);
        rb.AddForce(gravityForce);
    }

    void OnCollisionEnter(Collision collision)
    {
        Debug.Log("Collided with " + collision.gameObject.name);
    }
}
```

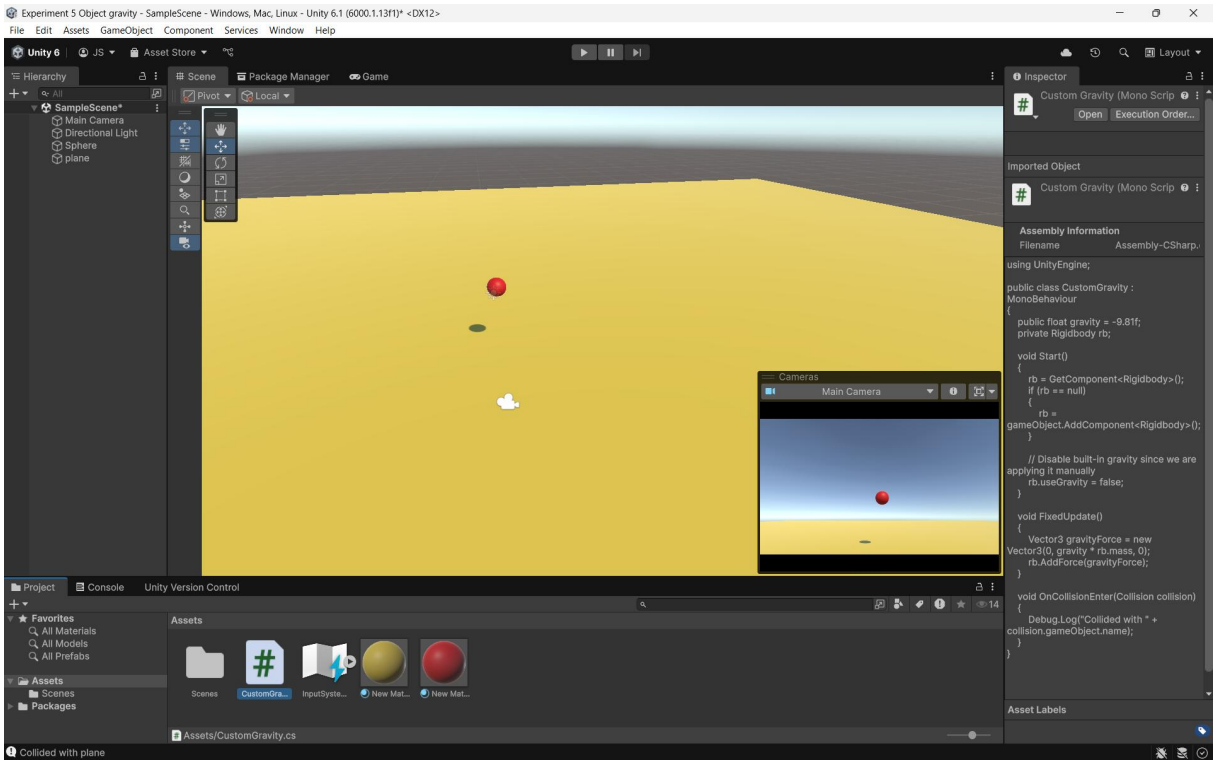
Output:

Ojects fall toward the ground at a rate determined by the custom gravity value.

Changing gravity to a higher negative value makes objects fall faster; smaller negative values slow down the fall.

Positive gravity values make objects float upward.

Console logs appear when objects collide with the ground or other obstacles, showing "Collided with <ObjectName>".



Conclusion:

This experiment demonstrates that Unity's built-in gravity can be overridden and customized per object.
By applying manual forces in FixedUpdate(), we gain precise control over

Fall speed,Gravity direction

Per-object gravitational behavior
Such an approach is useful in games requiring unique physics (e.g., space environments, underwater levels, or gameplay where different objects have different gravity strengths).

Result:

Script was created for gravity of object were successfully implemented and tested in Unity Game Engine 6.1 Their application in game mechanics was understood through coding and console outputs.