Title: Collision of one mobile and one static object

EXPERIMENT NO: 8 **Name:** SHAKYA JAY JITENDRA

Date: 8-9-2025 **Reg. No:** 24BCG10123

Aim: To study the effect of collisions between a mobile object (sphere) and a static object (cube) in Unity physics by varying their relative masses.

Description/Concept:

A sphere with a Rigidbody is launched forward with an initial velocity.

A cube acts as the static/massive object placed in front of the sphere.

Three different cases are tested by adjusting Rigidbody Mass values:

Equal mass (m1 = m2)

Mobile object heavier (m1 >> m2)

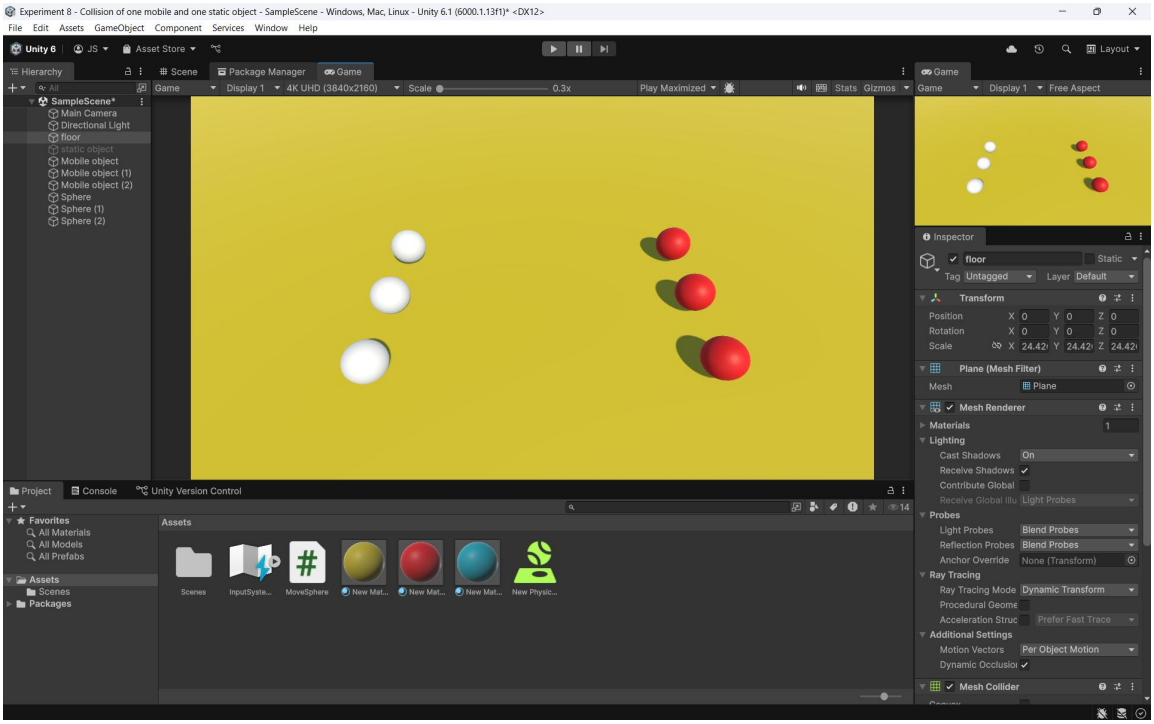
Mobile object lighter (m1 << m2)

The experiment is performed in Unity using the built-in Rigidbody physics system and a simple script to apply velocity to the sphere.

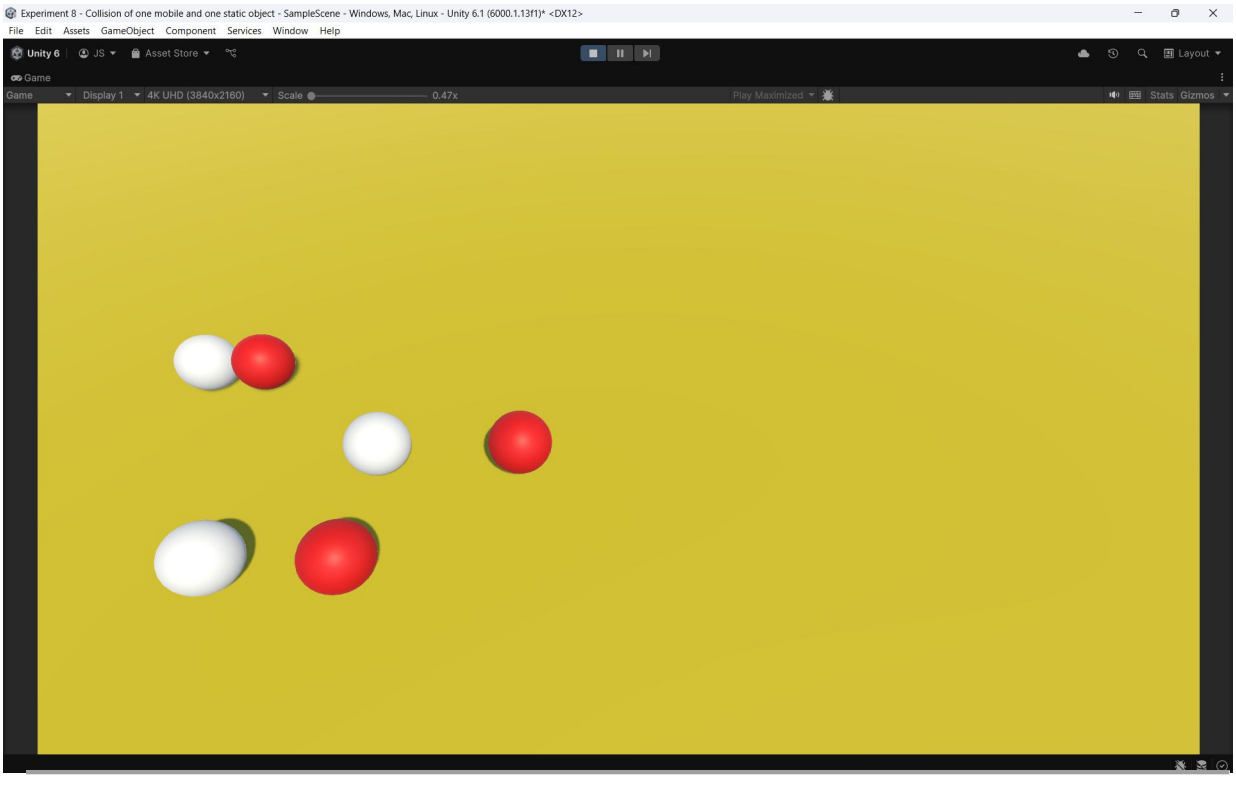
Program/Coding:

```
using UnityEngine;

public class MoveSphere : MonoBehaviour
{
    public float speed = 10f;     // Set in Inspector
    private Rigidbody rb;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
        rb.linearVelocity = transform.forward * speed;   // Push sphere forward
    }

}
```

Before Collision:



After Collision:



Output:

Output (Observed Behavior in Unity)

Case i: m1 = m2
On collision, both the sphere and cube move apart.
Motion is distributed equally, similar to billiard balls colliding.

Case ii: m1 >> m2
The sphere (mobile) continues almost unaffected.
The cube (static) is pushed strongly.

Case iii: m1 << m2
The sphere bounces back or stops.
The cube remains almost unaffected (like a ball hitting a wall)

Conclusion:

Unity's physics engine correctly simulates Newton's laws of motion and conservation of momentum.

The outcome of the collision depends directly on the mass ratio of the mobile and static objects.

Heavier objects dominate collisions, while lighter objects tend to bounce or stop

Equal mass → both move.

Mobile heavier → static object gets displaced.

Mobile lighter → mobile bounces back.

Result:

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