Experiment No.4
Develop a scene in Unity that includes a cube, plane and sphere. Create a new material and texture separately for three Game objects. Change the color, material and texture of each Game object separately in the scene. Write a C# program in visual studio to change the color and material/texture of the game objects dynamically on button click.
Date of Performance:
Date of Submission:

AIM: -

Develop a scene in Unity that includes a cube, plane and sphere. Create a new material and texture separately for three Game objects. Change the color, material and texture of each Game object separately in the scene. Write a C# program in visual studio to change the color and material/texture of the game objects dynamically on button click.

OBJECTIVES: -

The objectives for this Unity project are to create a scene with cube, plane, and sphere objects, each with unique materials and textures. The goal is to dynamically change the color, material, and texture of these objects individually upon clicking a button, achieved through a C# program. Key steps include scene setup, material and texture creation, assigning them to objects, implementing color changes, coding in Visual Studio, creating a user interface button, rigorous testing, and proper documentation. Successfully completing these objectives will enhance proficiency in Unity, C# programming, and interactive content development, fostering a well-rounded skill set in game development.

THEORY: -

This Unity project aims to create an interactive 3D scene comprising cube, plane, and sphere objects.

1) Scene Composition:
    a) Unity Scene: Unity scenes serve as the primary environment where game objects and assets are organized and interact. In this project, you'll create a new Unity scene to build your 3D environment.
    b) GameObjects: GameObjects are the fundamental building blocks of Unity scenes. You'll add three types of GameObjects – cube, plane, and sphere – to your scene to create the visual elements.

2) Material and Texture Creation:
    a) Materials: In Unity, materials define how an object appears by specifying its color, transparency, and how it reacts to lighting. You'll create separate materials for the cube, plane, and sphere to give them unique visual properties.
    b) Textures: Textures are 2D images applied to materials to add surface details, patterns, or realism. You'll create textures to enhance the appearance of your objects.

3) Material and Texture Assignment:
    a) Material Assignment: Unity allows you to assign materials to GameObjects. You'll

assign the previously created materials to their respective cube, plane, and sphere GameObjects.

b) Texture Assignment: Similarly, you'll assign textures to the materials, ensuring that each object gets the desired visual texture.

4) Color Modification:
   a) Dynamic Color Change: Through C# scripting, you'll create a mechanism to change the color of each GameObject individually. This will involve modifying the material's color properties during runtime based on user interactions, such as button clicks.

5) Material/Texture Swapping:
   a) Dynamic Material/Texture Change: You'll use C# programming in Visual Studio to enable dynamic changes in the material or texture properties of the GameObjects. These changes will occur in response to specific user actions, such as clicking a button in the Unity scene.

Theoretical understanding and practical application of these objectives will equip developers with essential skills in Unity, 3D asset manipulation, material and texture management, C# scripting for interactivity, and user interface integration. Ultimately, this project contributes to proficiency in Unity game development and interactive content creation.

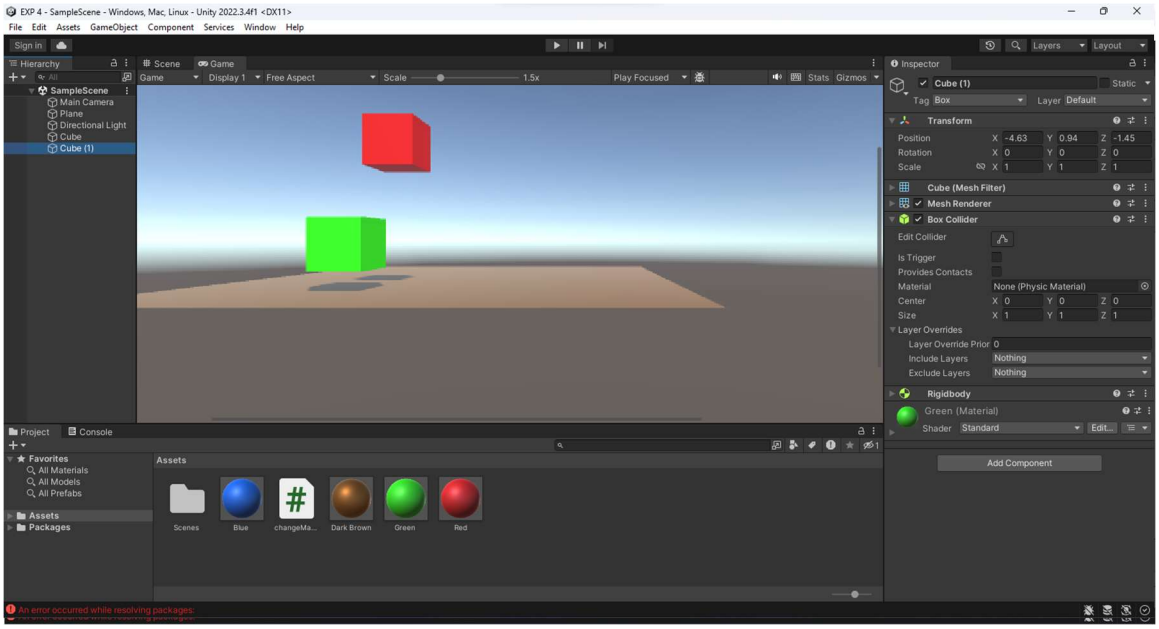
CODE: -

changematerial.cs

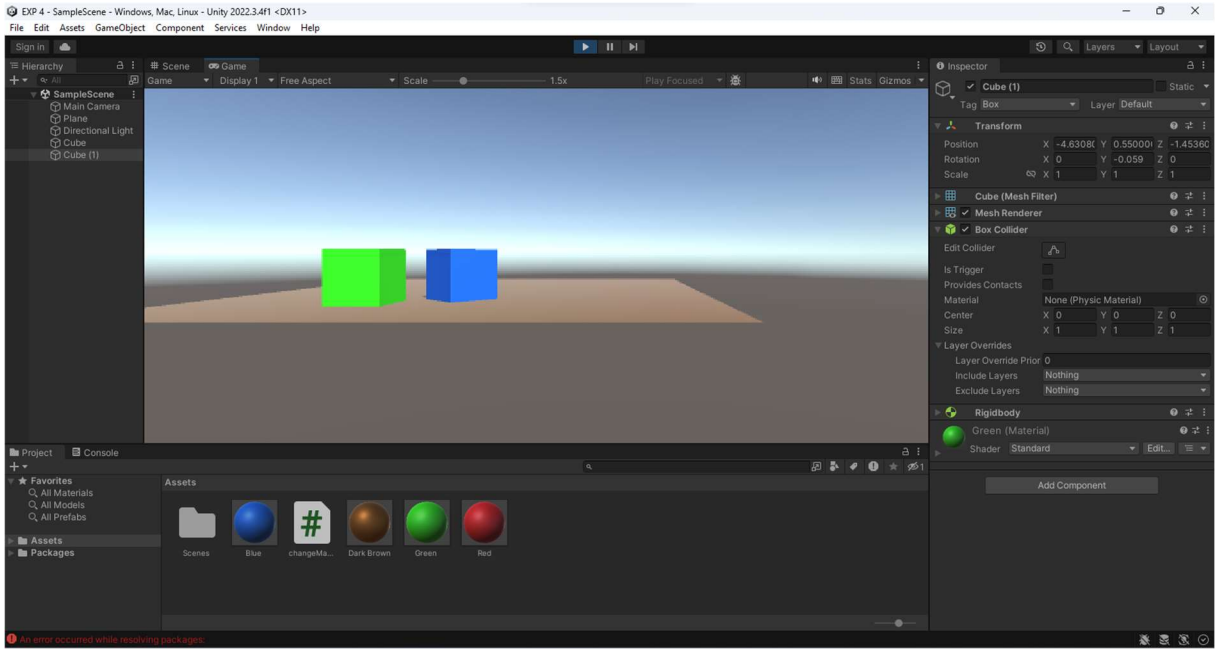
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class changeMaterial : MonoBehaviour
{
   public Material[] material;
   Renderer rend;
   void Start()
   {
      rend = GetComponent<Renderer>();
      rend.enabled = true;
      rend.sharedMaterial=material[0];
   }
```

```
    void OnCollisionEnter (Collision col)
    {
        if(col.gameObject.tag=="Box")
        {
            rend.sharedMaterial=material[1];
        }
        else
        {
            rend.sharedMaterial = material[2];
        }
    }
}
```

OUTPUT: -



After falling and on collision



CONCLUSION: -

This Unity project focused on constructing a dynamic 3D scene containing cube, plane, and sphere objects. For each object, distinct materials and textures were created and applied. The key achievement was enabling real-time adjustments to colour, material, and texture properties via a C# program in Visual Studio, initiated by a button click. The project illuminated core Unity concepts, including scene assembly, material management, and interactive scripting. It showcased the versatility of Unity for creating engaging user experiences. Ultimately, this endeavour bolstered Unity proficiency and the ability to craft interactive content, a valuable skill set applicable to diverse multimedia applications beyond gaming.