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 colour, material and texture of each Game object separately in the scene. Write a C# program in visual studio to change the colour and material/texture of the game objects dynamically on button click.

Theory:

Creating a scene in Unity with game objects (cube, plane, and sphere) and customizing their appearance involves understanding various concepts:

1. Game Objects: Game objects are the basic entities in Unity. They can represent characters, items, or other interactive elements in the scene. Each game object has a Transform component, which defines its position, rotation, and scale.

2. Materials: Materials are assets used to define the appearance of objects. They control how light interacts with the object's surface. Materials can have properties like color, shininess, transparency, and more.

3. Textures: Textures are 2D images that can be applied to materials. They can add visual details to objects and make them more realistic.

4. Changing Color, Material, and Texture:
   In Unity's Inspector panel, you can assign materials and textures to game objects by creating new materials and textures or using existing ones. You can change the color, material, and texture of a game object by modifying its renderer component.

5. C# Scripting:
   C# scripts in Unity enable you to add interactivity and functionality to your game objects. You can write scripts to change the color, material, and texture of game objects dynamically.
   Button clicks are typically handled using the Unity UI system, and you can write code to respond to button click events

Procedure:

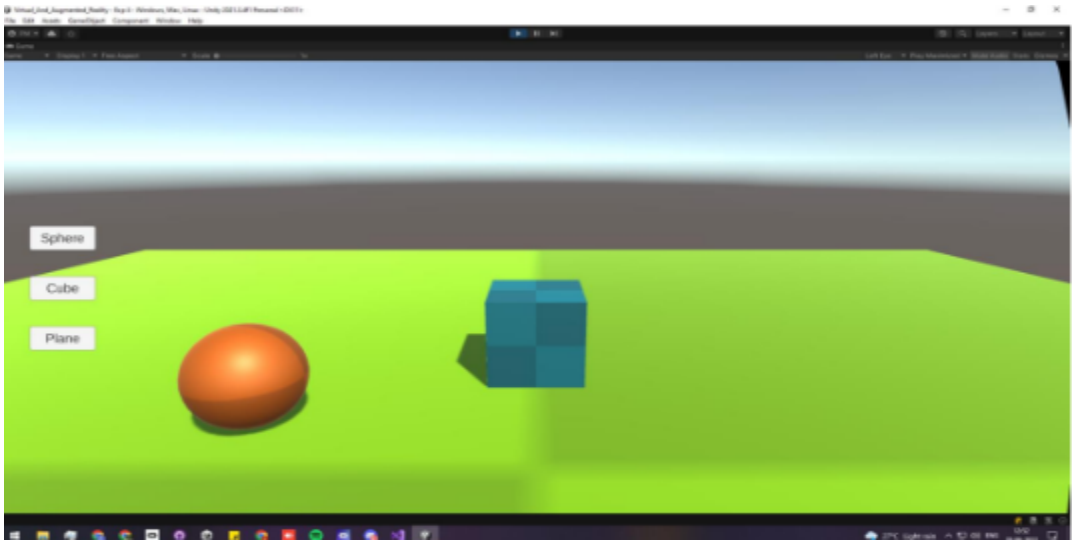
1. Scene creation:
   a. Create a scene named Exp 4 in the similar manner to experiment 3 with the same folder structure
   b. Create a cube, plane & sphere in the scene
   c. Position them & the camera in such a way that all of them are visible in the Game view in the similar manner.

2. Material creation:
   a. Material is an asset that is used to define how a gameobject with a renderer component will look in the game.
   b. Create a "Materials" folder inside the experiment 4 folder.
   c. Create a material inside the folder by right click -> Create-> Material.
   d. Click on the material & check if it has the standard shader assigned to it in the inspector.
   e. Name the material "Plane". Create 2 other materials for cube & sphere.

f. Give a different color to each material from the inspector by changing the albedo.

3. Assigning Materials:
   a. Go the Mesh Renderer component of each gameobject - cube, sphere, planEach
   b. GameObject will only have one material in it's materials list.
4. Creating Custom Components from code:
   a. Create la new folder, "Scripts" inside the experiment folder.
   b. Create > C# script
   c. Name the script "ColorChanger"
   d. Double click the script to open it up in Visual Studio.
5. Understanding MonoBehaviour:
   a. MonoBehaviour is the base class from which every Unity script derivels.
   b. If you need to add a script as a Component on a GameObject, you need to inherit the class from MonoBehaviour.
   c. When a script is created the Start & Update functions are already present in it by default.
6. Editing the script
   a. Remove the Update function from the script as we won't be needing that.
   b. We need to implement the color changing functionality inside this script.
   c. To do that we need a list of colors & textures that we will be changing on runtime at the button click.
   d. Write the script as following:
7. Creating User Interface (UI)
   a. Ul is the graphical interface where we can display information such as numbers, messages & add buttons for the user to interact with.
   b. Go back to Unity & in the hierarchy window, right click -> Ul -> Canvas.
   c. Now right click on the Canvas GameObject created & select Ul -> Button - TextMeshPro Name the gameobject "Color Changing Button"
   d. Click on the child text object of it & in the inspector in the text field type- "Switch Color

Results:



Conclusion:

In conclusion, the development of a Unity scene with dynamic color, material, and texture changes for game objects adds an interactive and engaging dimension to the world of 3D content creation. Unity's versatile platform allows you to create visually compelling scenes by customizing the appearance of game objects using materials and textures. This capability is especially valuable for game development, architectural visualization, and interactive simulations.The inclusion of C# scripting in Unity empowers developers to add interactivity to their projects. By writing scripts that respond to button clicks, you can dynamically alter the properties of game objects, such as their color, material, and texture. This interactive element enhances the user experience, making the scene more immersive and engaging.