Name:	Manas Parab
Roll No & Branch:	CSE-40
Class/Sem:	BE/VII
Experiment No.:	04
Title:	To 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 visualstudio to change the colour and material/texture of the game objects dynamically on button click
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim :-

To 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:-

Unity is a versatile game development platform that enables the creation of interactive 3D and 2D experiences. In this project, we establish a Unity scene featuring a cube, plane, and sphere while emphasizing crucial concepts like materials, textures, C# scripting, object manipulation, and user interaction. Materials define object appearance and texture application, with each object possessing its material. C# scripts enable dynamic property

changes, accessed through GameObjects, and handled via button clicks in Unity's UI system. Proper resource management and Play mode testing are integral parts of this project, serving as a foundation for developing engaging Unity applications and games.

Procedure:-

Step 1: Create an object

Select the GameObject menu in the menu bar. The GameObject Menu has several objects to create a game. Select the 3D object and pick the cube option.

The cube object will be displayed in the Scene View.

Increase the object size using the Scaling tool. Pull the arrow for the x-axis, y-axis, and z-axis. The object size can be increased.

Step 2: Create materials

Select the Assets menu in the menu bar. Select "Create" and pick the "Material" option. The material is displayed in the assets. Now change the material name to red. Select the Material (red) and click the color box. The color box is displayed on the left side. Choose the red color. The color is applied to your material (red).

Drag the red and drop it in the cube. The object can be changed into a red

color.Step 3: Create shaders and textures

Right-click on the assets, Select Import the asset.
Already I have brick wall images. If you want to apply for any other images, you can apply them. Now I have selected a brick wall image. Click to the Import button.

The brick wall image is added to assets. Select the image drag and drop the image into the object. The object can be changed fully to a brick wall. Now the new material added into the assets.

Now delete the brick wall image and double click on the new material. Delete the brick wall material. Select the material, press F2, and rename the new material to Textures.

Now, double click on the textures, open the brick wall image, and drag and drop the image into Textures.

Give the name for the Brick wall.

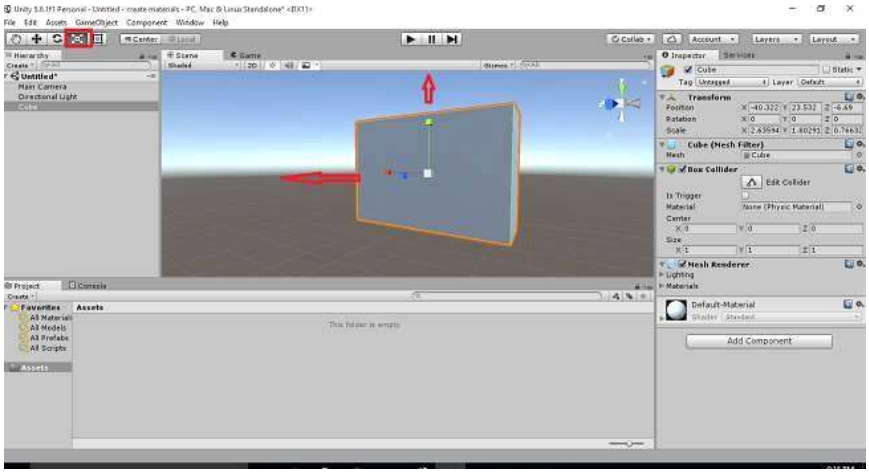
Select the Brick wall and go to the standard list box. Choose the Legacy shaders and pick the Diffuse option. When you select diffuse, the window is displayed like this.

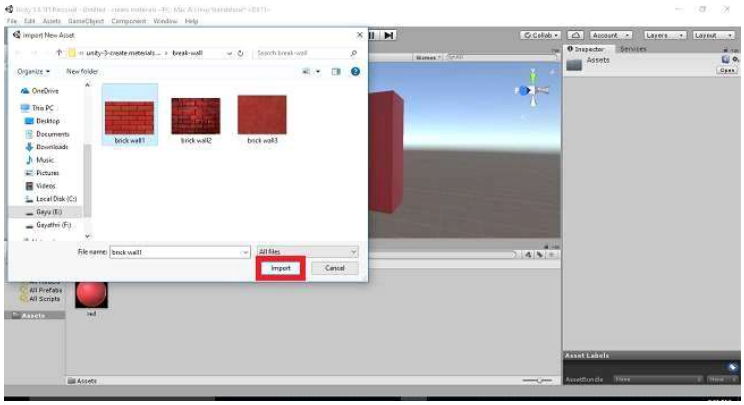
Click on the select option on your right side. Select the Texture window open on your leftside. Double click on the Brick wall. The brick wall material can be changed.
Pick the red marked tool. Press the alt key and scroll the mouse. You can see all sides in your object with a brick wall fully applied.

Step 4

Now, select the red material and go to a standard list box. Choose the Legacy shaders and picka specular option.
The object can be displayed in more shine.

Result:-





Conclusion:-

Creating a scene in Unity with a cube, plane, and sphere, and assigning separate materials and textures to each object, demonstrates versatile asset customization. Implementing a C# scriptin Visual Studio to dynamically change the color, material, and texture on button click enhances interactivity and showcases Unity's scripting capabilities. This project underscores Unity's flexibility in creating dynamic, user-controlled 3D environments for a wide range of applications, from gaming to simulations and more.