**Practical No. 7:**

**Aim: Programmable Diffuse Lightning in a rectangle using Direct3D 11**

**1)**After **completing the steps from the initialization file(“base setup.pdf”)**, now **open**

**“Form1.cs” file** in your project, and **code** the part where it is commented as **//OUR CODE*-----Form1.cs------***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using Microsoft.DirectX; //OUR CODE

using Microsoft.DirectX.Direct3D; //OUR CODE

namespace p8

{

publicpartialclassForm1 : Form

{

Microsoft.DirectX.Direct3D.Device device; //OUR CODE

public Form1()

{

InitializeComponent();

InitDevice(); //OUR CODE

}

private void InitDevice() //OUR CODE

{

PresentParameters pp = newPresentParameters();

pp.Windowed = true;

pp.SwapEffect = SwapEffect.Discard;

device=new Device(0, DeviceType.Hardware, this, CreateFlags.SoftwareVertexProcessing, pp); device.RenderState.Lighting = false;

device.Transform.Projection = Matrix.PerspectiveFovLH(3.14f / 4, device.Viewport.Width / device.Viewport.Height, 1f, 100f);

device.Transform.View = Matrix.LookAtLH(new Vector3(0, 0, 10), new Vector3(), new Vector3(0, 1, 0));

device.RenderState.Lighting = true; device.Lights[0].Type = LightType.Directional; device.Lights[0].Diffuse = Color.Plum;

device.Lights[0].Direction = newVector3(0.8f, 0, -1); device.Lights[0].Enabled = true;

}

private void Render() //OUR CODE

{

CustomVertex.PositionNormalColored[] vertexes = new

CustomVertex.PositionNormalColored[6]; //6 here is the number of vectors you've defined

//triangle1

vertexes[0] = new CustomVertex.PositionNormalColored(new Vector3(0, 2, 1), new Vector3(1, 0, 1), Color.Red.ToArgb());

vertexes[1] = new CustomVertex.PositionNormalColored(new Vector3(0, -2, 1), new Vector3(-1, 0, 1), Color.Red.ToArgb());

vertexes[2] = new CustomVertex.PositionNormalColored(new Vector3(2, -2,

1), new Vector3(-1, 0, 1), Color.Red.ToArgb());

//triangle2

vertexes[3] = new CustomVertex.PositionNormalColored(new Vector3(2, -2,

1), new Vector3(-1, 0, 1), Color.Red.ToArgb());

vertexes[4] = new CustomVertex.PositionNormalColored(new Vector3(2, 2, 1), new Vector3(-1, 0, 1), Color.Red.ToArgb());

vertexes[5] = new CustomVertex.PositionNormalColored(new Vector3(0, 2, 1), new Vector3(1, 0, 1), Color.Red.ToArgb());

device.Clear(ClearFlags.Target, Color.Cyan, 1.0f, 0); device.BeginScene();

device.VertexFormat = CustomVertex.PositionNormalColored.Format; device.DrawUserPrimitives(PrimitiveType.TriangleList, vertexes.Length / 3, vertexes);

//in the above line(vertexes.Length=6, so 6/3=2) hence, 1 for single triangle & 2 for double triangle device.EndScene();

device.Present();

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

Render(); //OUR CODE

}

}

}

**2)**Click the **Start button** to **run file**> (you’ll see the following **OUTPUT** of the window with the color you specified).

