

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Microsoft.DirectX;
using Microsoft.DirectX.Direct3D;

namespace GpPractical4
{
    public partial class Form1 : Form
    {
        Microsoft.DirectX.Direct3D.Device device;
        Microsoft.DirectX.Direct3D.Font font;
        Texture texture;
        public Form1()
        {
            InitializeComponent();
            InitDevice();
            LoadTexture();
            InitFont();
        }
        private void LoadTexture()
        {
            texture = TextureLoader.FromFile(device,
"C://Users//bot//source//repos//gppractical2//texture.png");
        }

        private void InitFont()
        {
            System.Drawing.Font f = new System.Drawing.Font("Arial", 12f,
FontStyle.Strikeout);
            font = new Microsoft.DirectX.Direct3D.Font(device, f);
        }

        public void InitDevice()
        {
            PresentParameters pp = new PresentParameters();
            pp.Windowed = true;
            pp.SwapEffect = SwapEffect.Discard;
            device = new Device(0, DeviceType.Hardware, this,
CreateFlags.HardwareVertexProcessing, pp);
        }
        private void Render()
        {
            device.Clear(ClearFlags.Target, Color.White, 0, 1);
            device.BeginScene();
            using (Sprite s = new Sprite(device))
            {

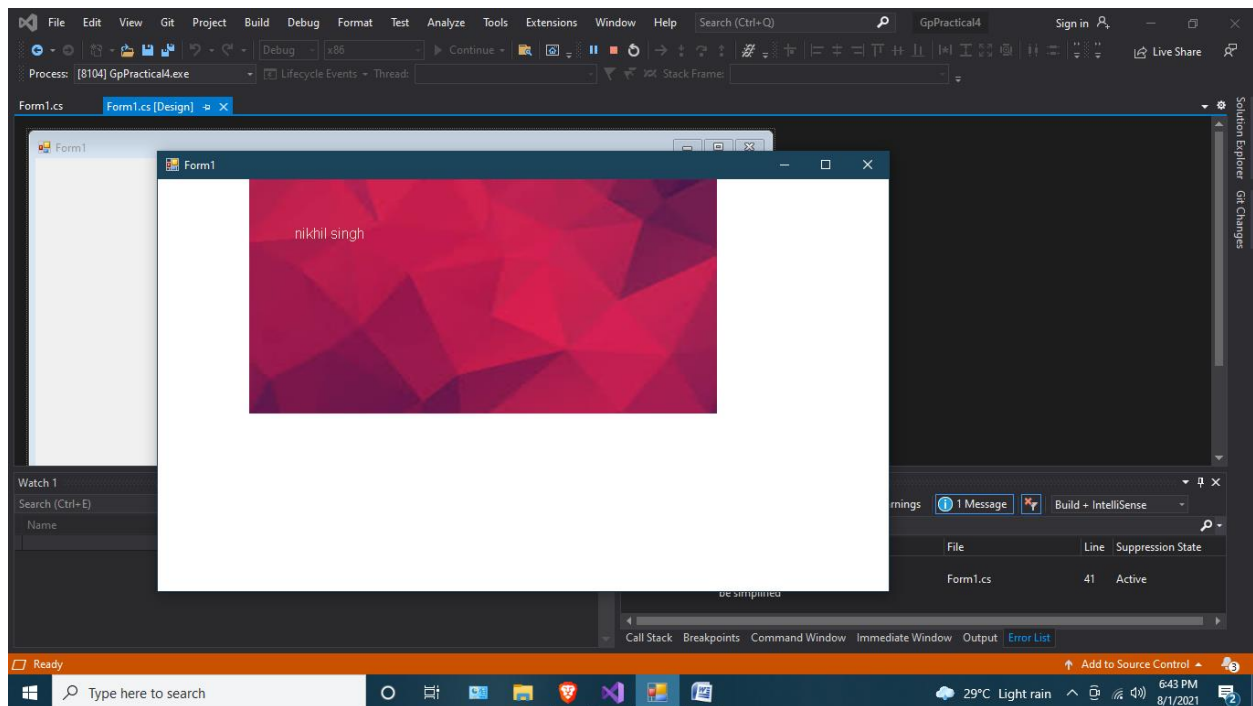
```

```

        s.Begin(SpriteFlags.AlphaBlend);
        s.Draw2D(texture, new Point(0, 0), 0, new Point(100, 0), Color.White);
        // s.Draw2D(texture, Rectangle.Empty, new Rectangle(0, 0, 500, 500), new
        Point(0,0),0, new Point(0,0), color.Violet);
        font.DrawText(s, "nikhil singh", new Point(150, 50), Color.Bisque);

        s.End();
    }
    device.EndScene(); device.Present();
}
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Render();
}
}

```



6B**Code:**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using Microsoft.DirectX.Direct3D;
using Microsoft.DirectX;

namespace GP_E
{
    public partial class Form1 : Form
    {
        private Device device;
        private PresentParameters pres;
        private Mesh mesh;
        private Material[] materials;
        private Texture[] textures;
        public Form1()
        {
            InitializeComponent();
        }
        public bool InitializeGraphics()
        {
            {
                pres = new PresentParameters();
                pres.Windowed = true;
                pres.SwapEffect = SwapEffect.Discard;
                pres.EnableAutoDepthStencil = true;
                pres.AutoDepthStencilFormat = DepthFormat.D16;
                device = new Device(0, DeviceType.Hardware, this,
                    CreateFlags.SoftwareVertexProcessing,
                    pres);
                device.RenderState.CullMode = Cull.None;
                CreateMesh(@"airplane 2.x");
                return true;
            }
        }
        public void CreateMesh(string path)
        {
            {
                ExtendedMaterial[] exMaterials;
                mesh = Mesh.FromFile(path, MeshFlags.SystemMemory, device,
                    out exMaterials);
                if (textures != null)
                {
                    DisposeTextures();
                }
                textures = new Texture[exMaterials.Length];
                materials = new Material[exMaterials.Length];

                for (int i = 0; i < exMaterials.Length; ++i)
                {
                    if (exMaterials[i].TextureFilename != null)
                    {
                        string texturePath = Path.Combine(Path.GetDirectoryName(path),
                            exMaterials[i].TextureFilename);
                        textures[i] = TextureLoader.FromFile(device, texturePath);
                    }
                }
            }
        }
    }
}

```

```

        materials[i] = exMaterials[i].Material3D;
        materials[i].Ambient = materials[i].Diffuse;
    }
}
public void DisposeTextures()
{
    if (textures == null)
    {
        return;
    }
    foreach (Texture t in textures)
    {
        if (t != null)
        {
            t.Dispose();
        }
    }
}
public void SetupMatrices()
{
    float yaw = Environment.TickCount / 500.0F;
    float pitch = Environment.TickCount / 500.0F;
    float roll = Environment.TickCount / 500.0F;
    device.Transform.World = Matrix.RotationYawPitchRoll(yaw, pitch, roll);
    device.Transform.View = Matrix.LookAtLH(new Vector3(0, 0, -8), new Vector3(0, 0, 0),
    new Vector3(0, 1, 0));
    device.Transform.Projection = Matrix.PerspectiveFovLH((float)Math.PI / 2.0F, 1.0F,
    1.0F, 12.0F);
}
public void SetupLights()
{
    device.RenderState.Lighting = true;
    device.Lights[0].Diffuse = Color.White;
    device.Lights[0].Specular = Color.White;
    device.Lights[0].Type = LightType.Directional;
    device.Lights[0].Direction = new Vector3(-1, -1, 3);
    device.Lights[0].Enabled = true;
    device.RenderState.Ambient = Color.FromArgb(0x00, 0x00, 0x00);
}
public void RenderMesh()
{
    for (int i = 0; i < materials.Length; ++i)
    {
        if (textures[i] != null)
        {
            device.SetTexture(0, textures[i]);
        }
        device.Material = materials[i];
        mesh.DrawSubset(i);
    }
}
public void Render()
{
    device.Clear(ClearFlags.Target | ClearFlags.ZBuffer, Color.SkyBlue, 1.0F, 0);
    device.BeginScene();
    SetupMatrices();
    SetupLights();
    RenderMesh();
    device.EndScene();
    device.Present();
}

```

```
    }  
    public void DisposeGraphics()  
    {  
        DisposeTextures();  
        device.Dispose();  
    }  
  
    private void Form1_Paint(object sender, PaintEventArgs e)  
    {  
        Render();  
    }  
}
```

Output:

