**Practical 6:**

Aim: Specular Lightning (Programmable Spot Lightning using Direct3D 11)

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.Direct3D; //OUR CODE using Microsoft.DirectX; //OUR CODE

namespace p11

{

public partial class Form1 : Form

{

private Device device; //OUR CODE private float angle = 0f; //OUR CODE public Form1()

{

InitializeComponent();

InitDevice(); //OUR CODE

this.SetStyle(ControlStyles.AllPaintingInWmPaint | ControlStyles.Opaque, true); //OUR CODE

}

private void InitDevice() //OUR CODE

{

PresentParameters pp = new PresentParameters(); pp.Windowed = true;

pp.SwapEffect = SwapEffect.Discard;

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

device.RenderState.CullMode = Cull.None; device.RenderState.Lighting = true; device.Lights[0].Type = LightType.Spot; device.Lights[0].Range = 4; device.Lights[0].Position = new Vector3(0, -1, 0f); device.Lights[0].Enabled = true;

}

private void Render() //OUR CODE

{

device.Transform.Projection = Matrix.PerspectiveFovLH((float)Math.PI / 4, this.Width / this.Height, 1f, 50f);

device.Transform.View = Matrix.LookAtLH(new Vector3(0, 0, 30), new Vector3(1, 0, 0), new Vector3(0, 5, 0));

CustomVertex.PositionNormalColored[] vertices = new CustomVertex.PositionNormalColored[6];

vertices[0].Position = new Vector3(10f, 12f, 0f); vertices[0].Normal = new Vector3(0, 2, 0.5f);

vertices[0].Color = Color.Yellow.ToArgb();

vertices[1].Position = new Vector3(-5f, 5f, 0f); vertices[1].Normal = new Vector3(0, 2, 0.5f); vertices[1].Color = Color.Blue.ToArgb();

vertices[2].Position = new Vector3(5f, 5f, -1f); vertices[2].Normal = new Vector3(0, 0, 0.5f); vertices[2].Color = Color.Pink.ToArgb();

vertices[3].Position = new Vector3(5f, -5f, -1f); vertices[3].Normal = new Vector3(0, 0, 0.5f); vertices[3].Color = Color.Green.ToArgb();

vertices[4].Position = new Vector3(10f, 12f, 0f); vertices[4].Normal = new Vector3(0, 0, 0.5f); vertices[4].Color = Color.Green.ToArgb();

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

Vector3 v;

device.VertexFormat = CustomVertex.PositionNormalColored.Format; device.Transform.World = Matrix.Translation(-5, -10 \* 1 / 3, 0) \*

Matrix.RotationAxis(new Vector3(), 0);

Console.WriteLine(device.Transform.World.ToString());

device.DrawUserPrimitives(PrimitiveType.TriangleStrip, 3, vertices); device.EndScene();

device.Present(); this.Invalidate();

}

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

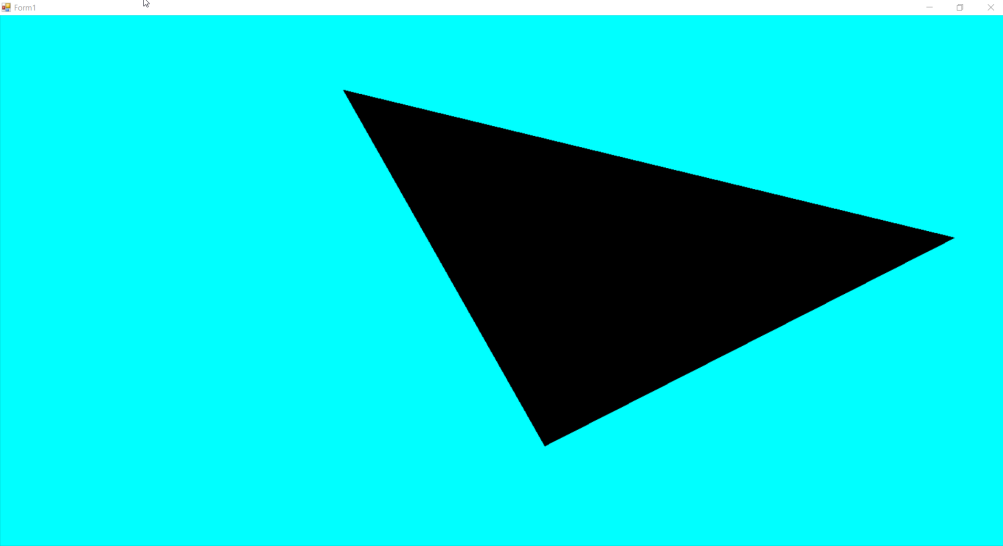
{

Render(); //OUR CODE

}

}

}



**Practical 7:**

Aim: Roll ball Tutorial

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class Move : MonoBehaviour

{

public Rigidbody rb;

public float h, v, speed = 10.0f; int count = 0;

public Text ct;

// Start is called before the first frame update

void Start()

{

rb = GetComponent<Rigidbody>();

count = 0;

ct.text = "";

}

// Update is called once per frame

void Update()

{

h = Input.GetAxis("Horizontal");

v = Input.GetAxis("Vertical");

}

public void FixedUpdate()

{

rb.AddForce(new Vector3(h, 0.0f, v) \* speed);

}

private void OnTriggerEnter(Collider other)

{

if (other.gameObject.CompareTag("coin"))

{

other.gameObject.SetActive(false); count++;

}

if (other.gameObject.CompareTag("coin1"))

{

other.gameObject.SetActive(false); count++;

}

if (count == 2)

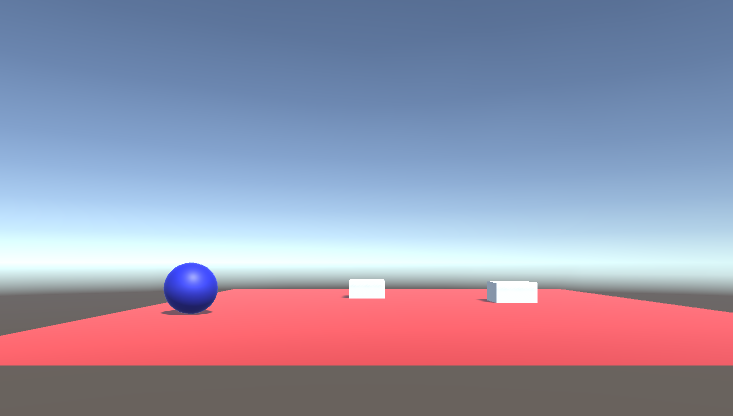
{

ct.text = "You won the game";

}

}

}





**Practical 8:**

Aim: UFO Tutorial

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class Move : MonoBehaviour

{

public Rigidbody2D rb;

public float h, v, speed = 10.0f; int count = 0;

public Text ct,tot;

// Start is called before the first frame update

void Start()

{

rb = GetComponent<Rigidbody2D>();

count = 0;

ct.text = "";

}

// Update is called once per frame

void Update()

{

h = Input.GetAxis("Horizontal");

v = Input.GetAxis("Vertical");

}

public void FixedUpdate()

{

rb.AddForce(new Vector3(h,0.0f,v)\*speed);

}

private void OnTriggerEnter(Collider other)

{

if (other.gameObject.CompareTag("Pickup1"))

{

other.gameObject.SetActive(false); count++;

}

if (other.gameObject.CompareTag("Pickup2"))

{

other.gameObject.SetActive(false); count++;

}

if (other.gameObject.CompareTag("Pickup3"))

{

other.gameObject.SetActive(false); count++;

}

if (other.gameObject.CompareTag("Pickup4"))

{

other.gameObject.SetActive(false); count++;

}

Tot.text=”count “+count;

if(count == 4)

{

ct.text = "You won the ufo game";

}

}

}

