**Practical no 1**

**AIM:** Create a 2D UFO Game using the Unity Engine.

**Code:**

**playerController.cs**

|  |
| --- |
| using System.Collections;  using UnityEngine;  using System.Collections.Generic;  using UnityEngine.UI;  public class PlayerController : MonoBehaviour {  public float speed;  public Text countText;  public Text winText;  private Rigidbody2D rb2d;  private int count;  void Start()  {  rb2d = GetComponent<Rigidbody2D>();  count = 0;  winText.text = "";  SetCountText ();  }  void FixedUpdate()  {  float moveHorizontal = Input.GetAxis("Horizontal");  float moveVertical = Input.GetAxis("Vertical");  Vector2 movement = new Vector2(moveHorizontal, moveVertical);  rb2d.AddForce(movement \* speed);  }  void OnTriggerEnter2D(Collider2D other)  {  if (other.gameObject.CompareTag ("PickUp"))  {  other.gameObject.SetActive (false);  count = count + 1;  SetCountText ();  }  }  void SetCountText()  {  countText.text="Count: " + count.ToString();  if (count >= 12) {  winText.text = "You Win!!!" ;  }  }  } |

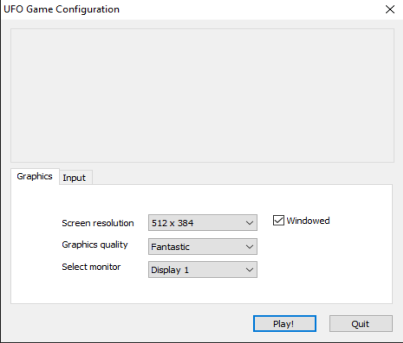
**CameraController.cs**

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class CameraController : MonoBehaviour {  public GameObject player ;  private Vector3 offset;  //us this for initialization  void Start()  {  offset = transform.position - player.transform.position;  }  //update is called once per frame  void LateUpdate()  {  transform.position=player.transform.position+offset;  }  } |

**Rotator.cs**

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Rotator : MonoBehaviour {  // Update is called once per frame  void Update ()  {  transform.Rotate (new Vector3 (0, 0, 45) \* Time.deltaTime);  }  } |

**output**







**Practical no 2**

**AIM:** Setup DirectX 11, Window Framework and Initialize Direct3D Device

**Steps:-**

1. Create a new project and select a windows form application(.Net Framework 2.0-3.5).
2. Right click on the properties → click on open → click build → select platform target → x86 or add new
3. Click on view code on form 1(design) or press F7.
4. Go to the solution explorer → right click on project name → select add reference .
5. Click on browse and add the required dll files.
6. Code the required files.
7. Add the paint method for changing the appearance .
8. Change the window name and icon if possible.
9. Disable the Exception Settings option such as LoaderLock.
10. Run the app.

**Code:-**

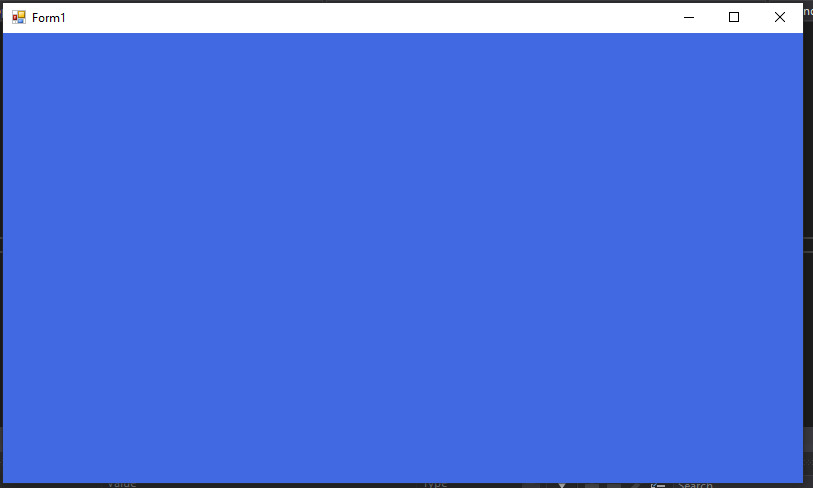
**Program.cs file**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Windows.Forms;  namespace WindowsFormsApp5  {  static class Program  {  /// <summary>  /// The main entry point for the application.  /// </summary>  [STAThread]  static void Main()  {  Application.EnableVisualStyles();  Application.SetCompatibleTextRenderingDefault(false);  Application.Run(new Form1());  }  }  } |

**Form1.cs file**

|  |
| --- |
| 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;  namespace WindowsFormsApp5  {  public partial class Form1 : Form  {  Microsoft.DirectX.Direct3D.Device device;  public Form1()  {  InitializeComponent();  InitDevice();  }  private void InitDevice()  {  PresentParameters pp = new PresentParameters();  pp.Windowed = true;  pp.SwapEffect = SwapEffect.Discard;  device = new Device(0, DeviceType.Hardware, this, CreateFlags.HardwareVertexProcessing, pp);  }  public void Render()  {  device.Clear(ClearFlags.Target, Color.RoyalBlue, 0, 1);  device.Present();  }  private void Form1\_Paint(object sender, PaintEventArgs e)  {  Render();  }  }  } |

**Output**

****

**Practical no 3**

**AIM:** Buffers, Shaders and HLSL (Draw a triangle/rectangle using Direct3D 11)

**Steps:-**

1. Create a new project and select a windows form application(.Net Framework 2.0-3.5).
2. Right click on the properties → click on open → click build → select platform target → x86 or add new
3. Click on view code on form 1(design) or press F7.
4. Go to the solution explorer → right click on project name → select add reference .
5. Click on browse and add the required dll files.
6. Code the required files.
7. Add the paint method for changing the appearance .
8. Change the window name and icon if possible.
9. Disable the Exception Settings option such as LoaderLock.
10. Run the app.

**Code:-**

**Program.cs file**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Windows.Forms;  using Microsoft.DirectX.Direct3D;  namespace WindowsFormsApp6  {  static class Program  {    [STAThread]  static void Main()  {  Application.EnableVisualStyles();  Application.SetCompatibleTextRenderingDefault(false);  Application.Run(new Form1());  }  }  } |

**Form1.cs file**

|  |
| --- |
| 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;  using Microsoft.DirectX.Direct3D;  namespace WindowsFormsApp6  {  public partial class Form1 : Form  {  private Device device;  // Microsoft.DirectX.Direct3D.Device device;  private CustomVertex.PositionColored[] vertex = new CustomVertex.PositionColored [3];  public Form1()  {  InitializeComponent();  }  private void Form1\_Paint(object sender, PaintEventArgs e)  {  device.Clear(ClearFlags.Target, Color.Black, 1, 0);  device.BeginScene();  device.VertexFormat = CustomVertex.PositionColored.Format;  device.DrawUserPrimitives(PrimitiveType.TriangleList, vertex.Length / 3, vertex);  device.EndScene();  device.Present();  }  private void Form1\_Load(object sender, EventArgs e)  {  PresentParameters pp = new PresentParameters();  pp.Windowed = true;  pp.SwapEffect = SwapEffect.Discard;  device = new Device(0, DeviceType.Hardware, this, CreateFlags.HardwareVertexProcessing, pp);  device.Transform.Projection = Matrix.PerspectiveFovLH(3.14f / 4 , device.Viewport.Width / device.Viewport.Height, 1f , 1000f);  device.Transform.View = Matrix.LookAtLH(new Vector3(0, 0, 20), new Vector3(), new Vector3(0, 1, 0));  device.RenderState.Lighting = false;  vertex[0] = new CustomVertex.PositionColored(new Vector3(0, 0, 0), Color.Green.ToArgb());  vertex[1] = new CustomVertex.PositionColored(new Vector3(4, 0, 0), Color.White.ToArgb());  vertex[2] = new CustomVertex.PositionColored(new Vector3(2, 4, 0), Color.Orange.ToArgb());  }  }  } |

**Output:**

