**MALAD KANDIVALI EDUCATION SOCIETY'S**
**NAGINDAS KHANDWALA COLLEGE OF COMMERCE,**
**ARTS & MANAGEMENT STUDIES & SHANTABEN NAGINDAS**
**KHANDWALA COLLEGE OF SCIENCE**
**MALAD [W], MUMBAI – 64**
**(AUTONOMOUS)**

**(Reaccredited 'A' Grade by NAAC)**
**(AFFILIATED TO UNIVERSITY OF MUMBAI)**
**(ISO 9001:2015)**

# CERTIFICATE

**Name: Mr./Ms. Kuldeep Shushil Patel**

**Roll No: 574** **Programme**: BSc CS **Semester**: V

This is certified to be a bonafide record of practical works done by the above student in the college laboratory for the course **GAME PROGRAMMING** (Course Code: **1857UCSPR**) for the partial fulfillment of Fifth Semester of BSc CS during the academic year 2020-2021.

The journal work is the original study work that has been duly approved in the year 2020-2021 by the undersigned.

**External Examiner**

**Mr. Ashish Modi**
**(Subject-In-Charge)**

**Date of Examination:** **(College Stamp)**

**Subject: GAME PROGRAMMING** **Class: T. Y. BSc. CS**

# Index

# Practical no 1

## Aim : setup DirectX11, window freamework and initialize Direct3D Device.

**Step 1 =** given the name of the project gp_prac1_65 and select the framework 2.0 in property section and selected the path to store my project.

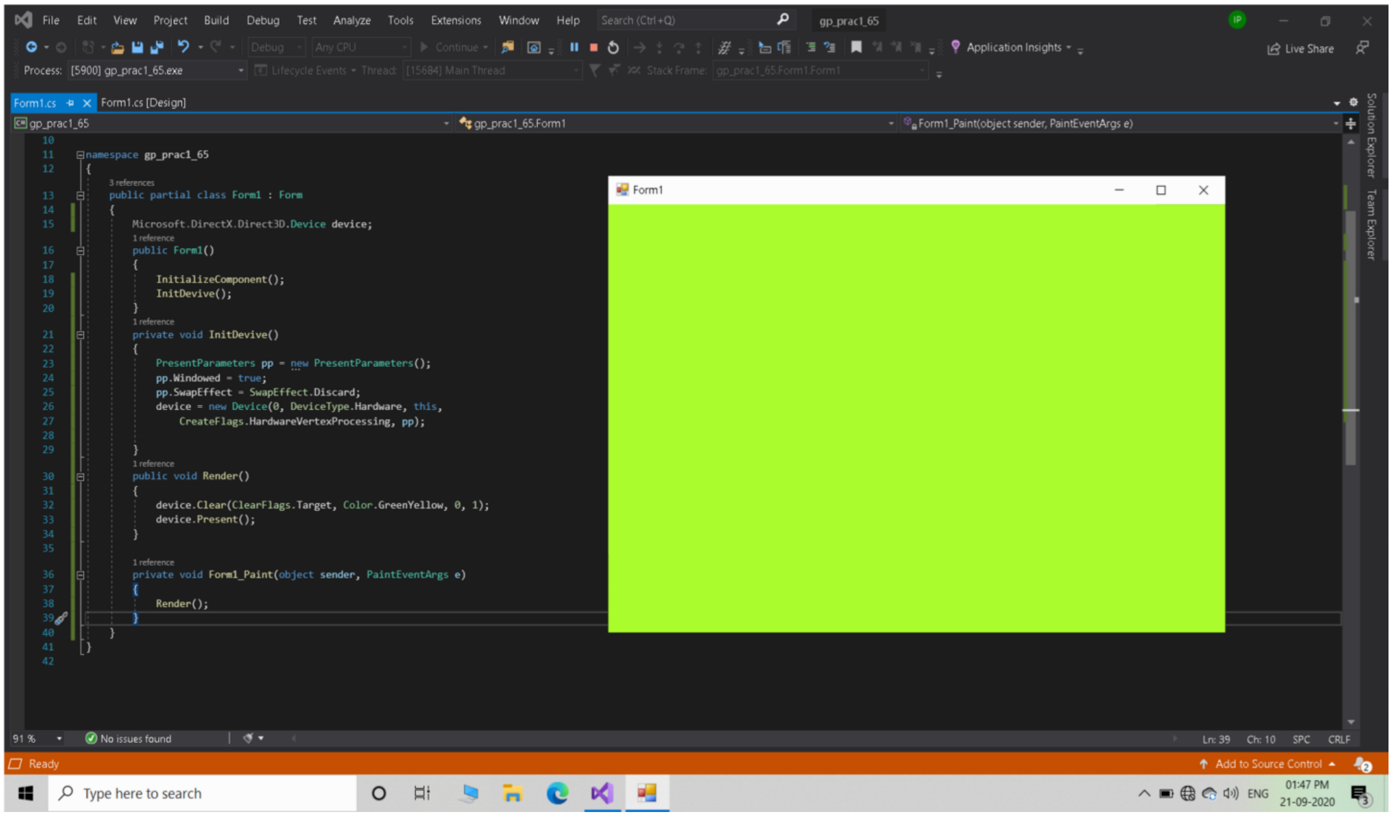
**Step2 =** in solution explorer right clicked on the property name **property - open – add – build – plateform target – select the x86**

**Step 3 =** now we have to add the references so in solution explorer click on **reference(right click) – add refernce – browse – this pc – local c drive – windows – microsoft.net – direct X managed code – 1.0.2902.0 – press ctrl and select (microsoft.directx , microsoft.directx.direct3d , Microsoft.DirectX.Direct3DX.dll) - add – ok**

**Step4 =** right click of form1 window and click on view code and the in property event list – click on paint option than it will create the form1_paint refernce.

**Step 5 =** to uncheck the loder lock option click on **debug – windows – exception settings – In down click on managed debugging assistence – uncheck the loader lock there.**

**Step 6 =** type the code and click on **start (button)** or **debug – start debugging. Step 7** = we will get the output as below



```
namespace gp_prac1_65
{
    public partial class Form1 : Form
    {
        Microsoft.DirectX.Direct3D.Device device;
        public Form1()
        {
            InitializeComponent();
            InitDevive();
        }
        private void InitDevive()
        {
            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.GreenYellow, 0, 1);
            device.Present();
        }

        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            Render();
        }
    }
}
```

## Practical no. 2

**Aim** = draw a triangle using Direct3D 11 .

**Step 1 =** given the name of the project gp_prac2_65 and select the framework 2.0 in property section and selected the path to store my project.

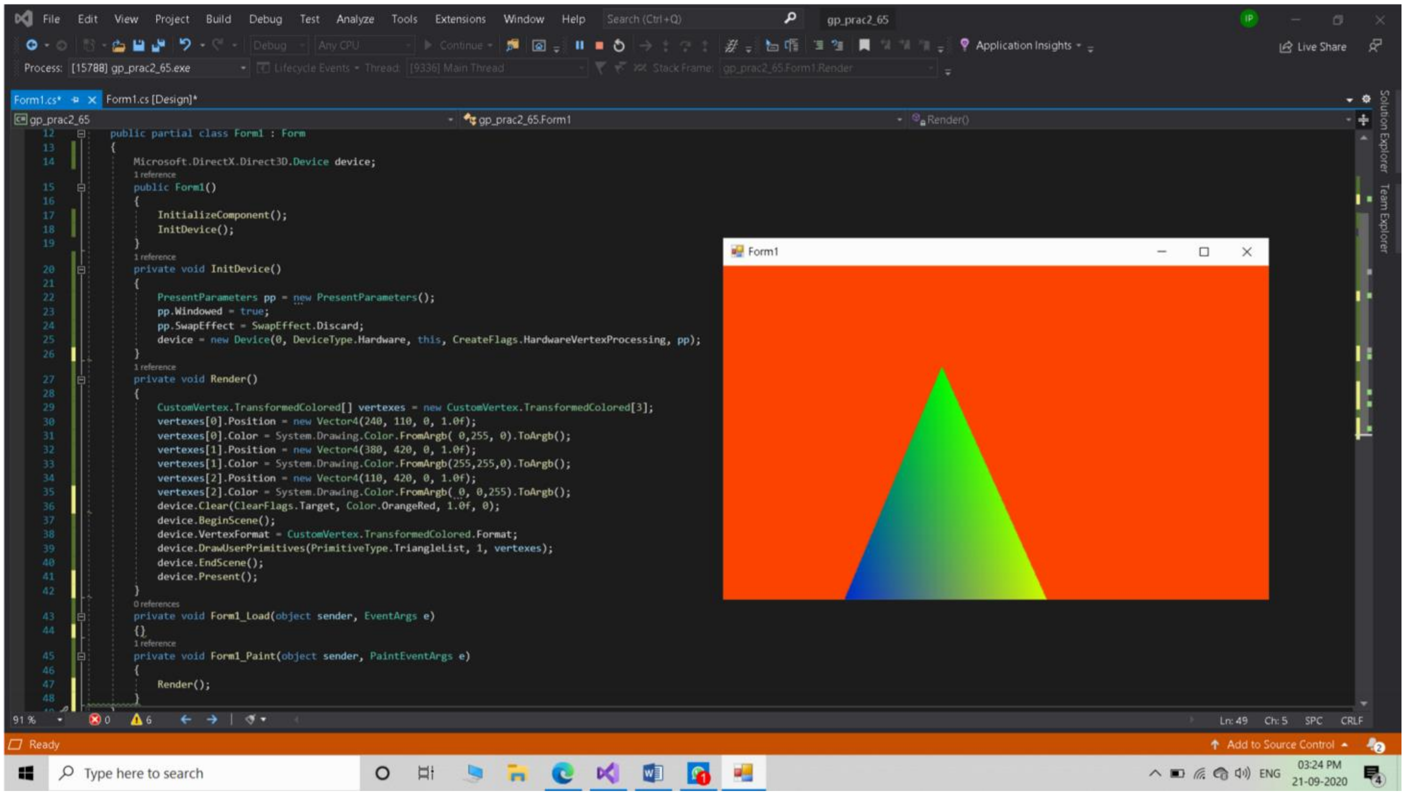
**Step2 =** in solution explorer right clicked on the property name **property - open – add – build – plateform target – select the x86**

**Step 3 =** now we have to add the references so in solution explorer click on **reference(right click) – add refernce – browse – this pc – local c drive – windows – microsoft.net – direct X managed code – 1.0.2902.0 – press ctrl and select (microsoft.directx , microsoft.directx.direct3d , Microsoft.DirectX.Direct3DX.dll) - add – ok**

**Step4 =** right click of form1 window and click on view code and the in property event list – click on paint option than it will create the form1_paint refernce.

**Step 5 =** to uncheck the loder lock option click on **debug – windows – exception settings – In down click on managed debugging assistence – uncheck the loader lock there.**

**Step 6 =** type the code and click on **start (button)** or **debug – start debugging. Step 7** = we will get the output as below



```
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);
    }
    private void Render()
    {
        CustomVertex.TransformedColored[] vertexes = new CustomVertex.TransformedColored[3];
        vertexes[0].Position = new Vector4(240, 110, 0, 1.0f);
        vertexes[0].Color = System.Drawing.Color.FromArgb( 0,255, 0).ToArgb();
        vertexes[1].Position = new Vector4(380, 420, 0, 1.0f);
        vertexes[1].Color = System.Drawing.Color.FromArgb(255,255,0).ToArgb();
        vertexes[2].Position = new Vector4(110, 420, 0, 1.0f);
        vertexes[2].Color = System.Drawing.Color.FromArgb( 0, 0,255).ToArgb();
        device.Clear(ClearFlags.Target, Color.OrangeRed, 1.0f, 0);
        device.BeginScene();
        device.VertexFormat = CustomVertex.TransformedColored.Format;
        device.DrawUserPrimitives(PrimitiveType.TriangleList, 1, vertexes);
        device.EndScene();
        device.Present();
    }
    private void Form1_Load(object sender, EventArgs e)
    {}
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        Render();
    }
```

# Practical no 3

## **Aim =** Texture the triangle using Direct3D 11 .

**Step 1 =** given the name of the project gp_prac3_65 and select the framework 2.0 in property section and selected the path to store my project.

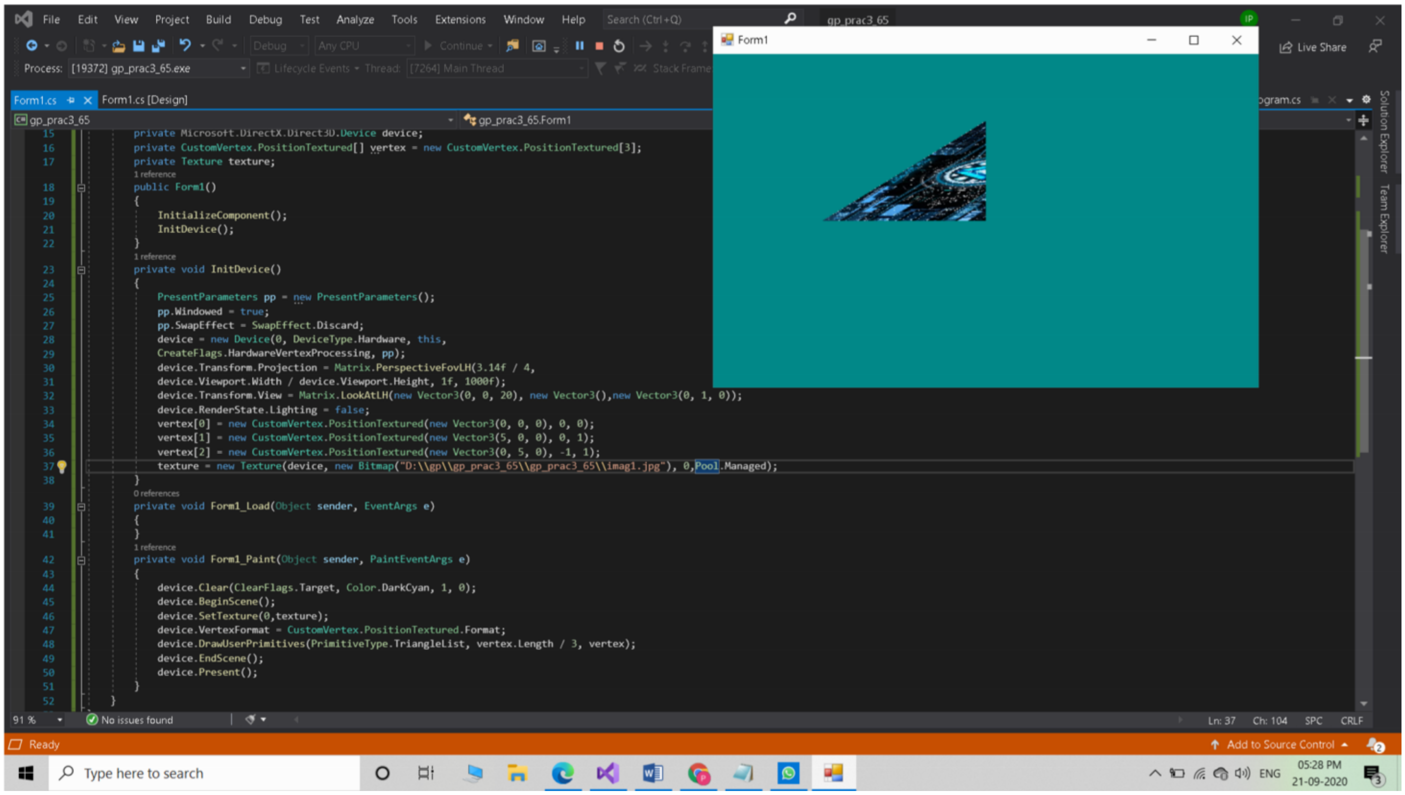
**Step2 =** in solution explorer right clicked on the property name **property - open – add – build – plateform target – select the x86**

**Step 3 =** now we have to add the references so in solution explorer click on **reference(right click) – add refernce – browse – this pc – local c drive – windows – microsoft.net – direct X managed code – 1.0.2902.0 – press ctrl and select (microsoft.directx , microsoft.directx.direct3d , Microsoft.DirectX.Direct3DX.dll) - add – ok**

**Step4 =** right click of form1 window and click on view code and the in property event list – click on paint option than it will create the form1_paint refernce.

**Step 5 =** to uncheck the loder lock option click on **debug – windows – exception settings – In down click on managed debugging assistence – uncheck the loader lock there.**

**Step 6 =** type the code and click on **start (button)** or **debug – start debugging. Step 7**
= we will get the output as below



```
        private Microsoft.DirectX.Direct3D.Device device;
        private CustomVertex.PositionTextured[] vertex = new CustomVertex.PositionTextured[3];
        private Texture texture;
        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);
            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.PositionTextured(new Vector3(0, 0, 0), 0, 0);
            vertex[1] = new CustomVertex.PositionTextured(new Vector3(5, 0, 0), 0, 1);
            vertex[2] = new CustomVertex.PositionTextured(new Vector3(0, 5, 0), -1, 1);
            texture = new Texture(device, new Bitmap("D:\\gp\\gp_prac3_65\\gp_prac3_65\\imag1.jpg"), 0,Pool.Managed);
        }
        private void Form1_Load(Object sender, EventArgs e)
        {
        }
        private void Form1_Paint(Object sender, PaintEventArgs e)
        {
            device.Clear(ClearFlags.Target, Color.DarkCyan, 1, 0);
            device.BeginScene();
            device.SetTexture(0,texture);
            device.VertexFormat = CustomVertex.PositionTextured.Format;
            device.DrawUserPrimitives(PrimitiveType.TriangleList, vertex.Length / 3, vertex);
            device.EndScene();
            device.Present();
        }
    }
```

# Practical no. 4

## Aim : Programmable Diffuse Lightning using Direct3D 11 .

**Step 1 =** given the name of the project gp_prac4_65 and select the framework 2.0 in property section and selected the path to store my project.

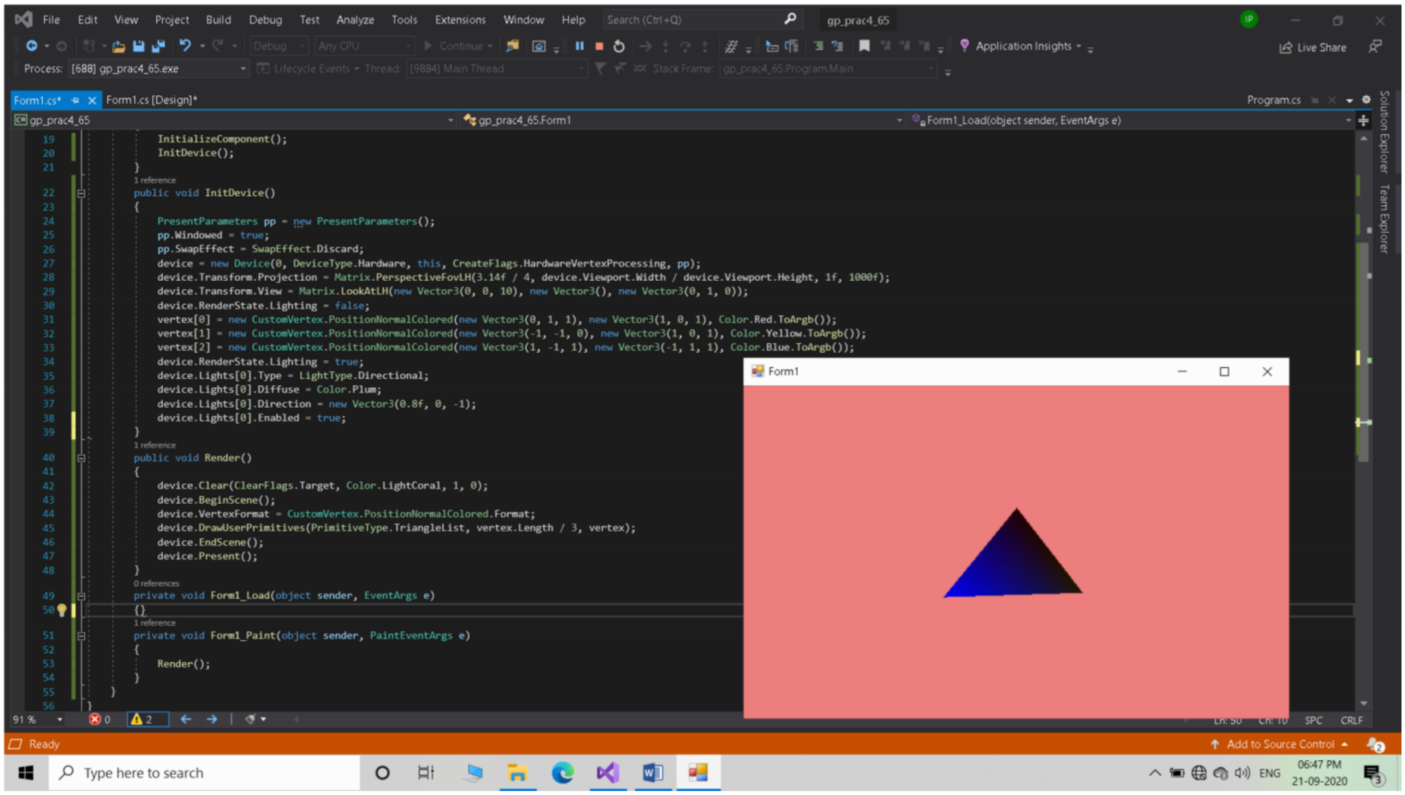
**Step2 =** in solution explorer right clicked on the property name **property - open – add – build – plateform target – select the x86**

**Step 3 =** now we have to add the references so in solution explorer click on **reference(right click) – add refernce – browse – this pc – local c drive – windows – microsoft.net – direct X managed code – 1.0.2902.0 – press ctrl and select (microsoft.directx , microsoft.directx.direct3d , Microsoft.DirectX.Direct3DX.dll) - add – ok**

**Step4 =** right click of form1 window and click on view code and the in property event list – click on paint option than it will create the form1_paint refernce.

**Step 5 =** to uncheck the loder lock option click on **debug – windows – exception settings – In down click on managed debugging assistence – uncheck the loader lock there.**

**Step 6 =** type the code and click on **start (button)** or **debug – start debugging. Step 7** = we will get the output as below



```
            InitializeComponent();
            InitDevice();
        }
        public void InitDevice()
        {
            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, 10), new Vector3(), new Vector3(0, 1, 0));
            device.RenderState.Lighting = false;
            vertex[0] = new CustomVertex.PositionNormalColored(new Vector3(0, 1, 1), new Vector3(1, 0, 1), Color.Red.ToArgb());
            vertex[1] = new CustomVertex.PositionNormalColored(new Vector3(-1, -1, 0), new Vector3(1, 0, 1), Color.Yellow.ToArgb());
            vertex[2] = new CustomVertex.PositionNormalColored(new Vector3(1, -1, 1), new Vector3(-1, 1, 1), Color.Blue.ToArgb());
            device.RenderState.Lighting = true;
            device.Lights[0].Type = LightType.Directional;
            device.Lights[0].Diffuse = Color.Plum;
            device.Lights[0].Direction = new Vector3(0.8f, 0, -1);
            device.Lights[0].Enabled = true;
        }
        public void Render()
        {
            device.Clear(ClearFlags.Target, Color.LightCoral, 1, 0);
            device.BeginScene();
            device.VertexFormat = CustomVertex.PositionNormalColored.Format;
            device.DrawUserPrimitives(PrimitiveType.TriangleList, vertex.Length / 3, vertex);
            device.EndScene();
            device.Present();
        }
        private void Form1_Load(object sender, EventArgs e)
        {}
        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            Render();
        }
    }
}
```

## Practical no 5

**Aim :** Loading models into DirectX11 and rendering.

**Step 1 =** given the name of the project gp_prac5_65 and select the framework 2.0 in property section and selected the path to store my project.

**Step2 =** in solution explorer right clicked on the property name **property - open – add – build – plateform target – select the x86**

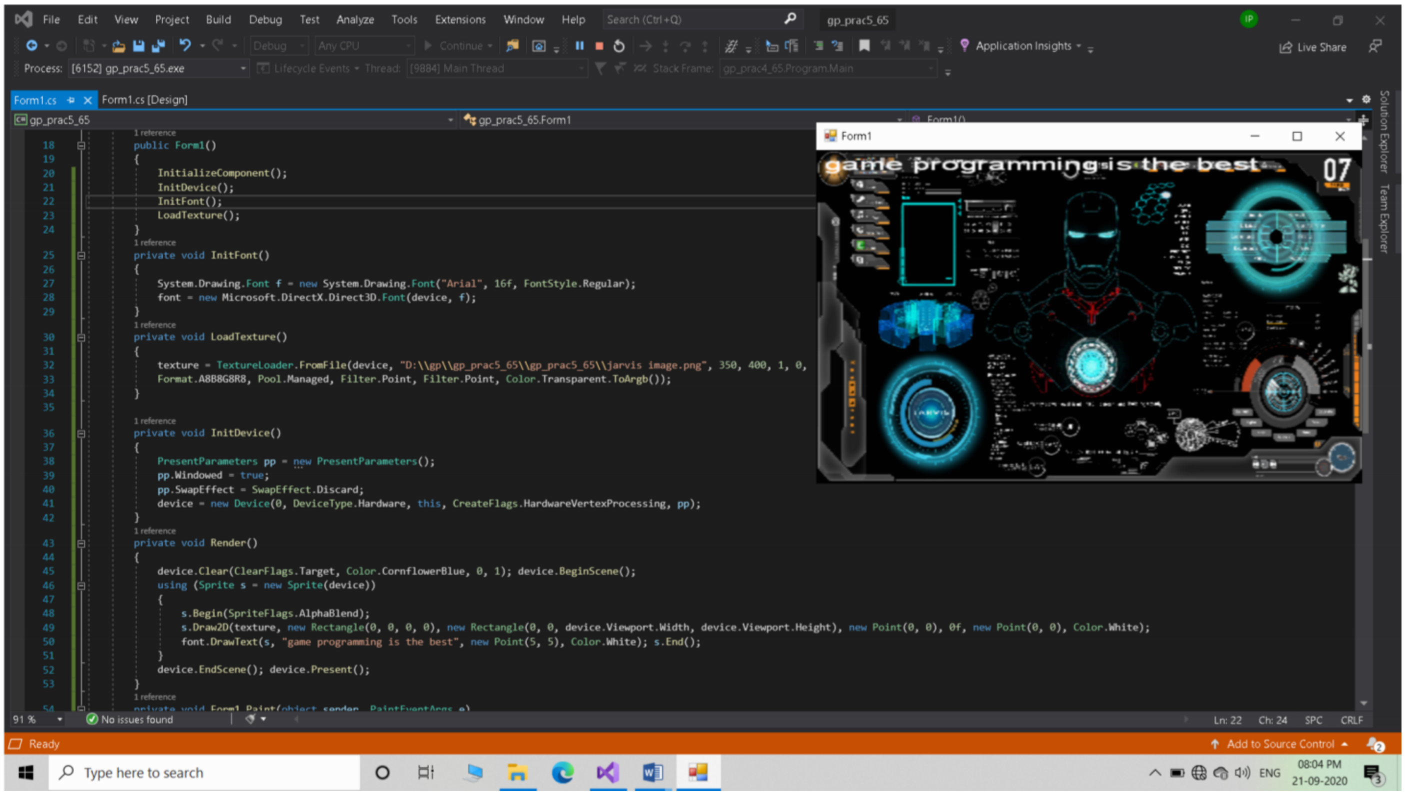
**Step 3 =** now we have to add the references so in solution explorer click on **reference(right click) – add refernce – browse – this pc – local c drive – windows – microsoft.net – direct X managed code – 1.0.2902.0 – press ctrl and select (microsoft.directx , microsoft.directx.direct3d , Microsoft.DirectX.Direct3DX.dll) - add – ok**

**Step4 =** right click of form1 window and click on view code and the in property event list – click on paint option than it will create the form1_paint refernce.

**Step 5 =** to uncheck the loder lock option click on **debug – windows – exception settings – In down click on managed debugging assistence – uncheck the loader lock there.**

**Step 6 =** type the code and click on **start (button)** or **debug – start debugging.**

**Step 7 =** we will get the output as below

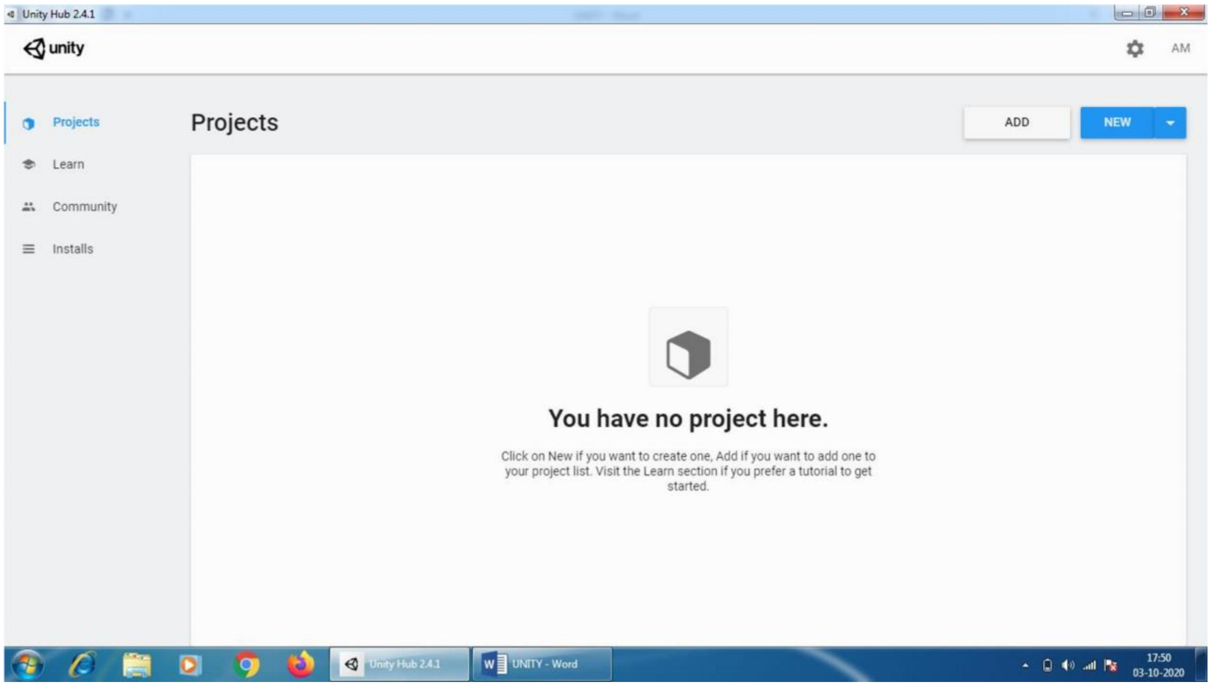


```csharp
        public Form1()
        {
            InitializeComponent();
            InitDevice();
            InitFont();
            LoadTexture();
        }
        private void InitFont()
        {
            System.Drawing.Font f = new System.Drawing.Font("Arial", 16f, FontStyle.Regular);
            font = new Microsoft.DirectX.Direct3D.Font(device, f);
        }
        private void LoadTexture()
        {
            texture = TextureLoader.FromFile(device, "D:\\gp\\gp_prac5_65\\gp_prac5_65\\jarvis image.png", 350, 400, 1, 0,
            Format.A8B8G8R8, Pool.Managed, Filter.Point, Filter.Point, Color.Transparent.ToArgb());
        }

        private 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.CornflowerBlue, 0, 1); device.BeginScene();
            using (Sprite s = new Sprite(device))
            {
                s.Begin(SpriteFlags.AlphaBlend);
                s.Draw2D(texture, new Rectangle(0, 0, 0, 0), new Rectangle(0, 0, device.Viewport.Width, device.Viewport.Height), new Point(0, 0), 0f, new Point(0, 0), Color.White);
                font.DrawText(s, "game programming is the best", new Point(5, 5), Color.White); s.End();
            }
            device.EndScene(); device.Present();
        }
        private void Form1_Paint(object sender, PaintEventArgs e)
```

# Practical no 6

## Aim : Implementing 2D UFO

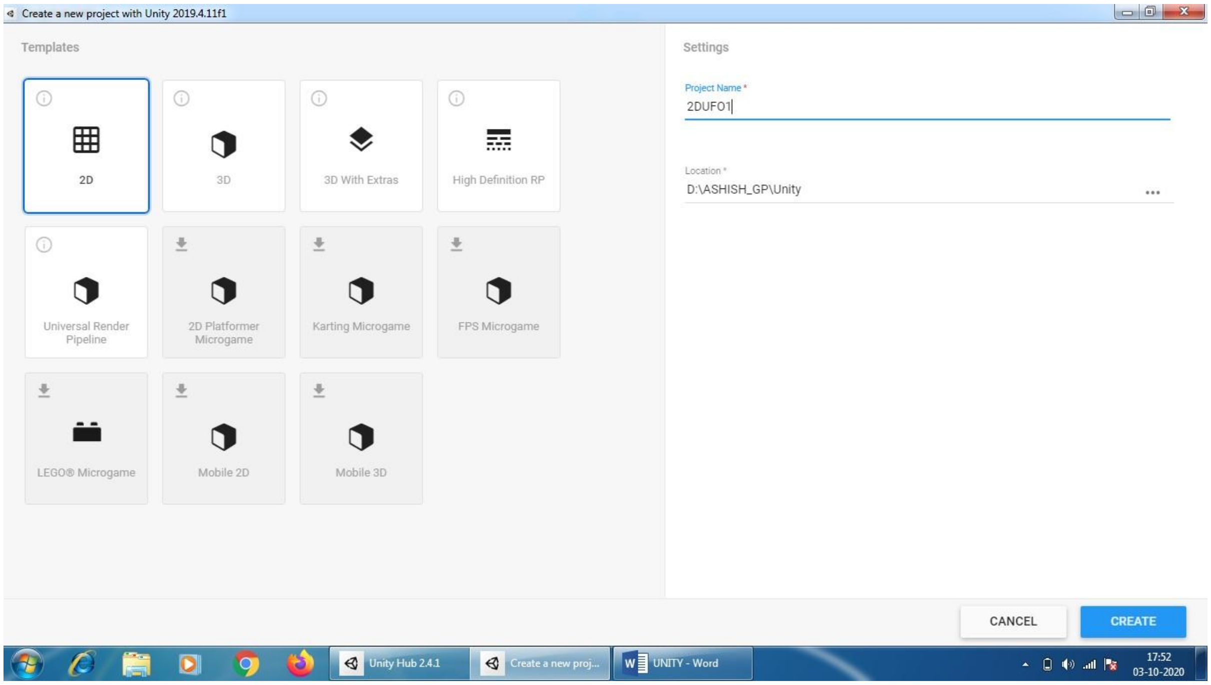
**Step 1 :** Under Projects Tab, Click on NEW



You have no project here.

Click on New if you want to create one, Add if you want to add one to your project list. Visit the Learn section if you prefer a tutorial to get started.

**Step 2 :** Click on 2D, Give Project Name, Path and Click on CREATE



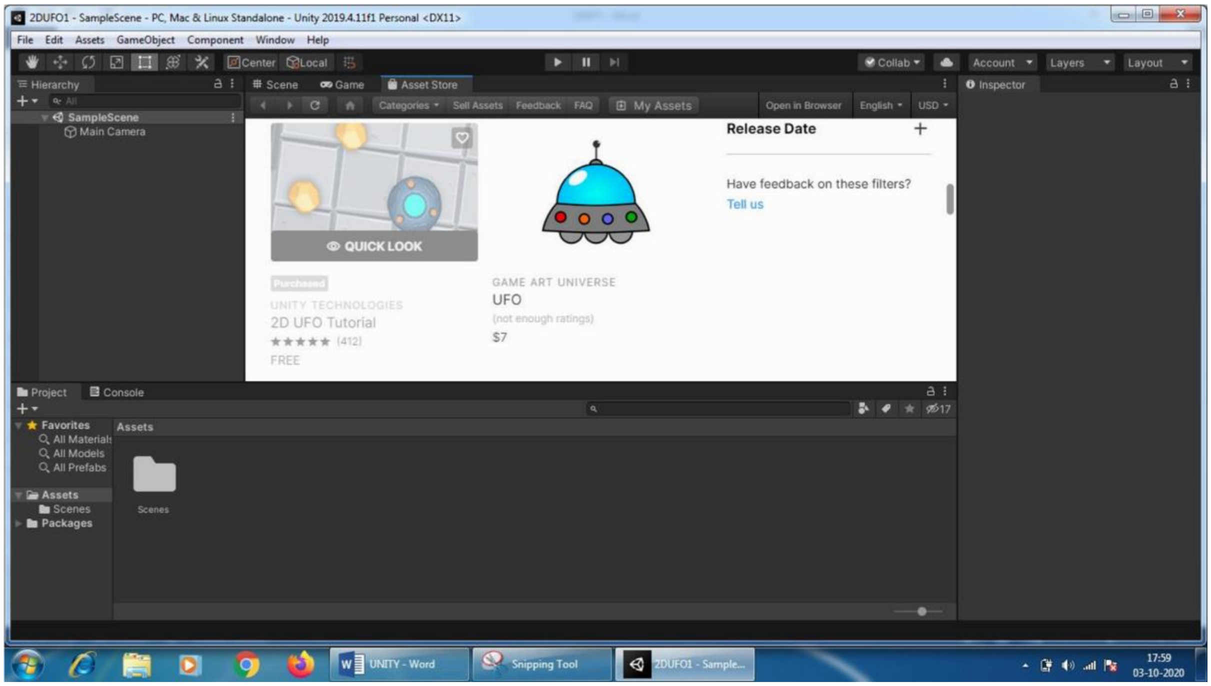
Create a new project with Unity 2019.4.11f1

Settings

Project Name * 2DUFO1

Location * D:\ASHISH_GP\Unity

**Step 3 :** Scroll down and click on 2D UFO



2DUFO1 - SampleScene - PC, Mac & Linux Standalone - Unity 2019.4.11f1 Personal <DX11>

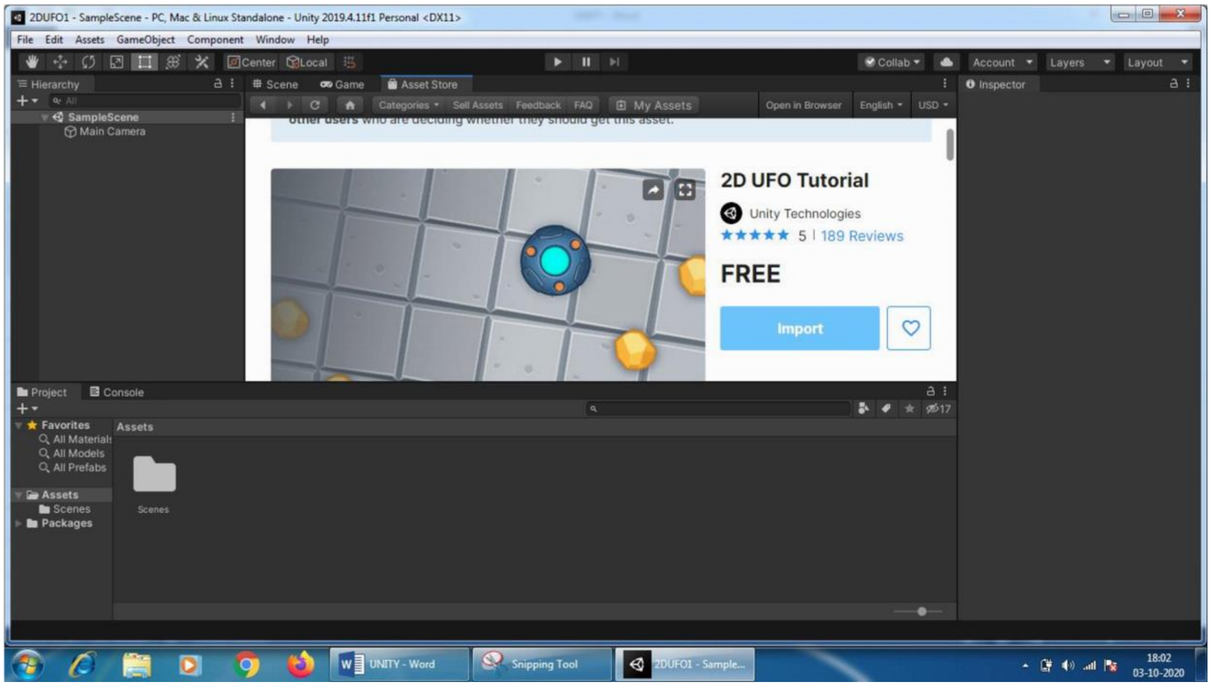
Purchased
UNITY TECHNOLOGIES
2D UFO Tutorial
★★★★★ (412)
FREE

GAME ART UNIVERSE
UFO
(not enough ratings)
$7

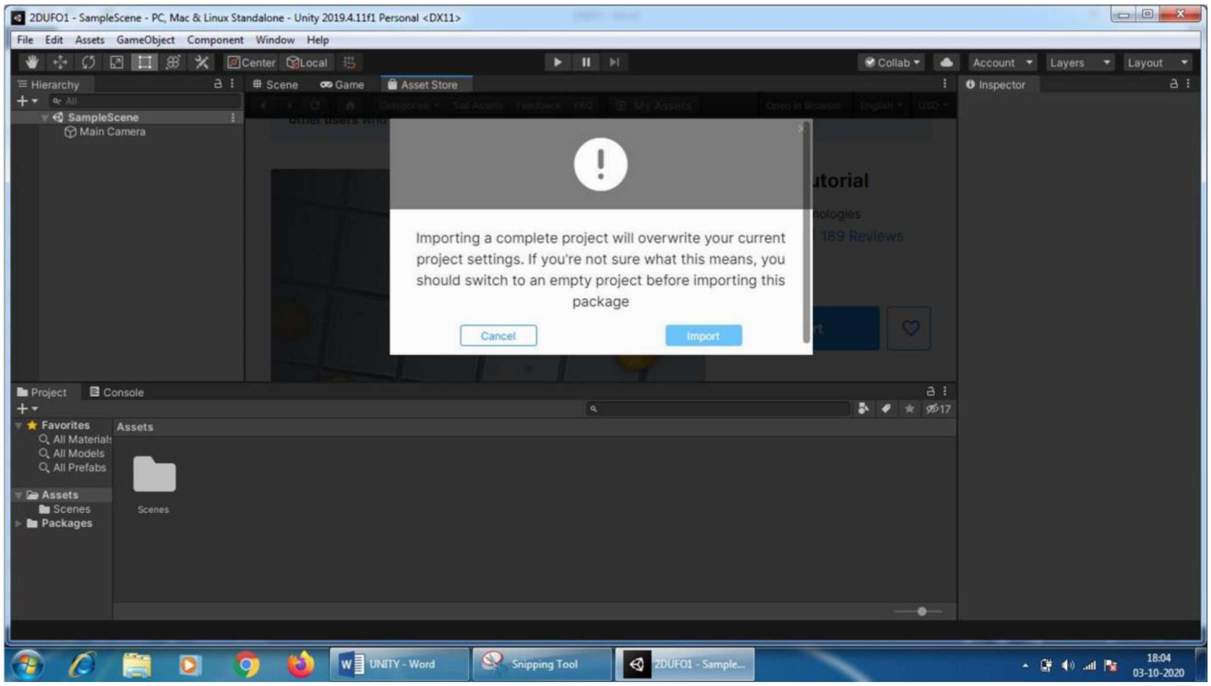
Release Date

Have feedback on these filters?
Tell us

**Step 4 :** Click on IMPORT button

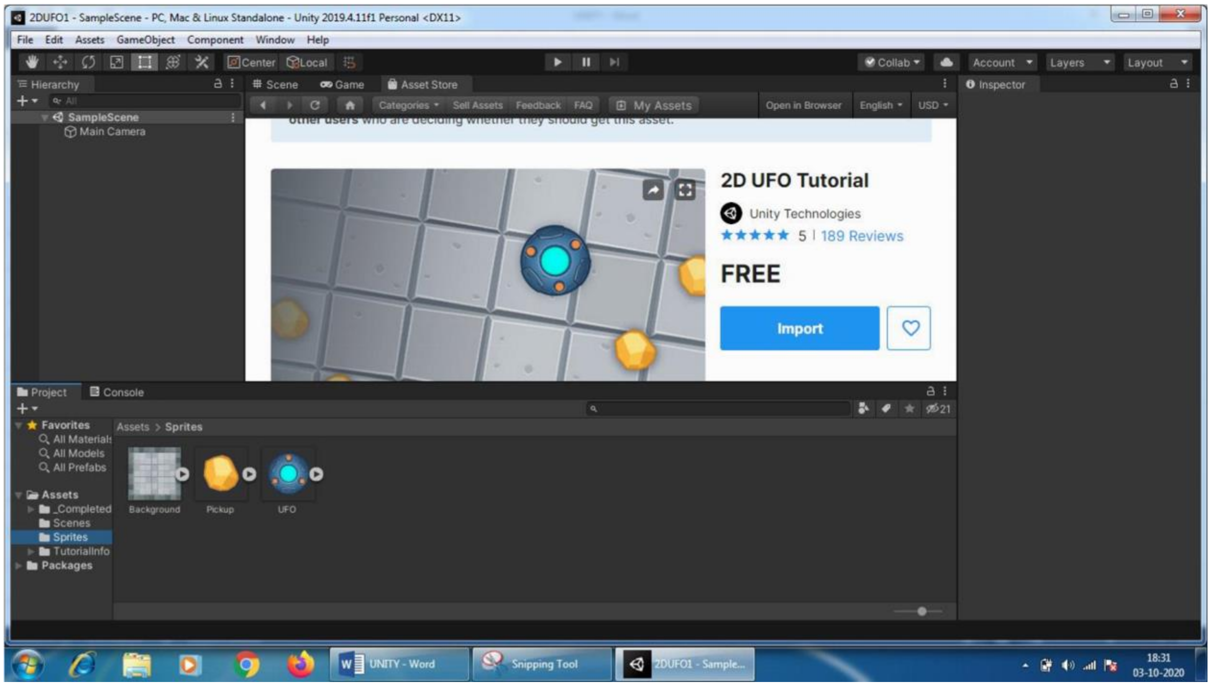


**Step 5 :** Click on Install / Upgrade button

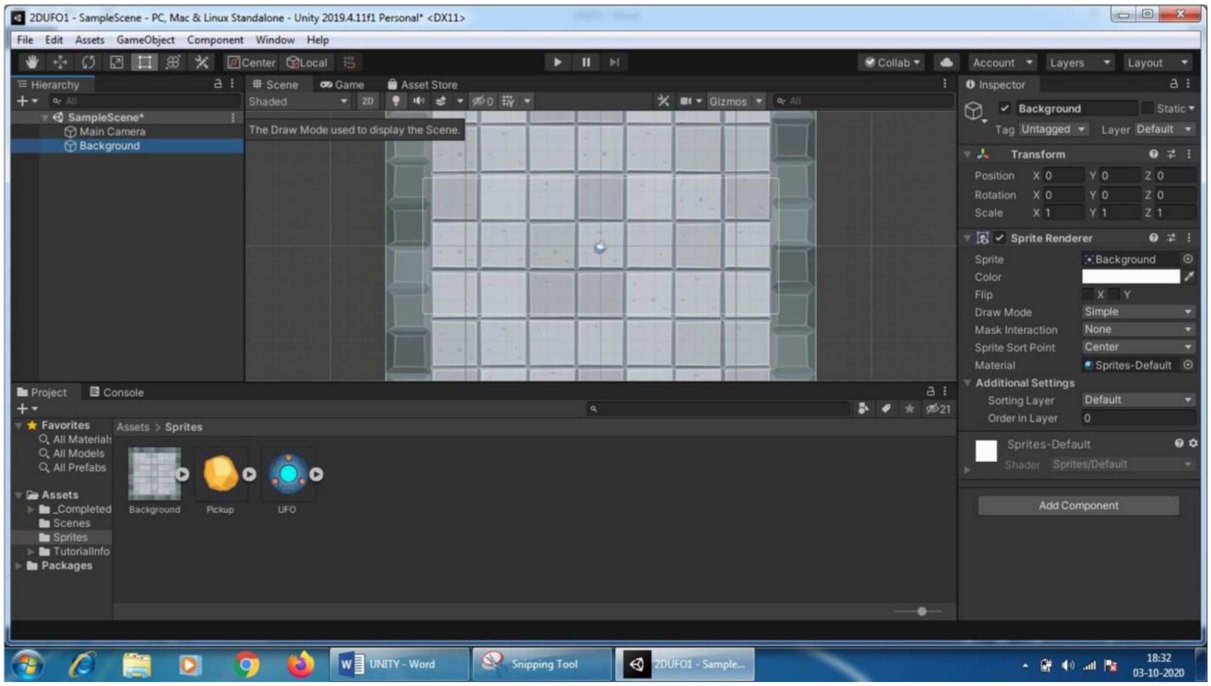


Importing a complete project will overwrite your current project settings. If you're not sure what this means, you should switch to an empty project before importing this package

**Step 6 :** Click on the SPRITES folder



**Step 7 :** Come to Scene tab and drag the BACKGROUND under MAIN CAMERA


2DUFO1 - SampleScene - PC, Mac & Linux Standalone - Unity 2019.4.11f1 Personal* <DX11>

The Draw Mode used to display the Scene.

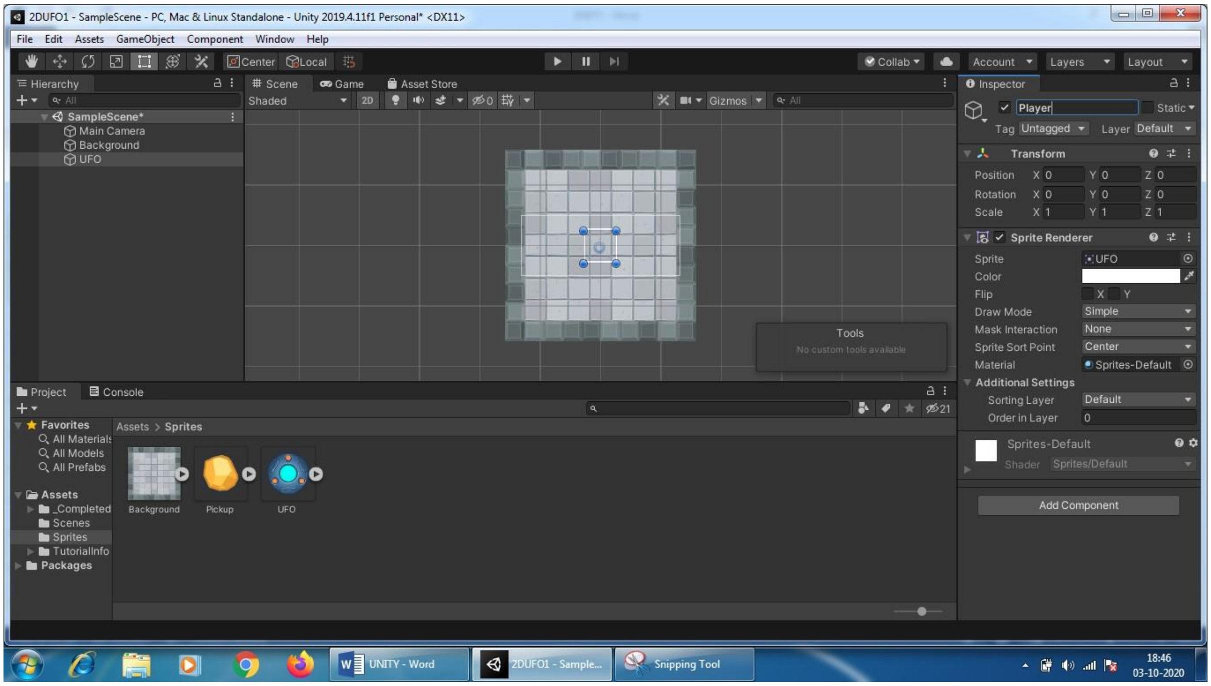
Background  Static
Tag Untagged  Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Sprite Renderer
Sprite: Background
Color
Flip: X Y
Draw Mode: Simple
Mask Interaction: None
Sprite Sort Point: Center
Material: Sprites-Default
Additional Settings
Sorting Layer: Default
Order in Layer: 0

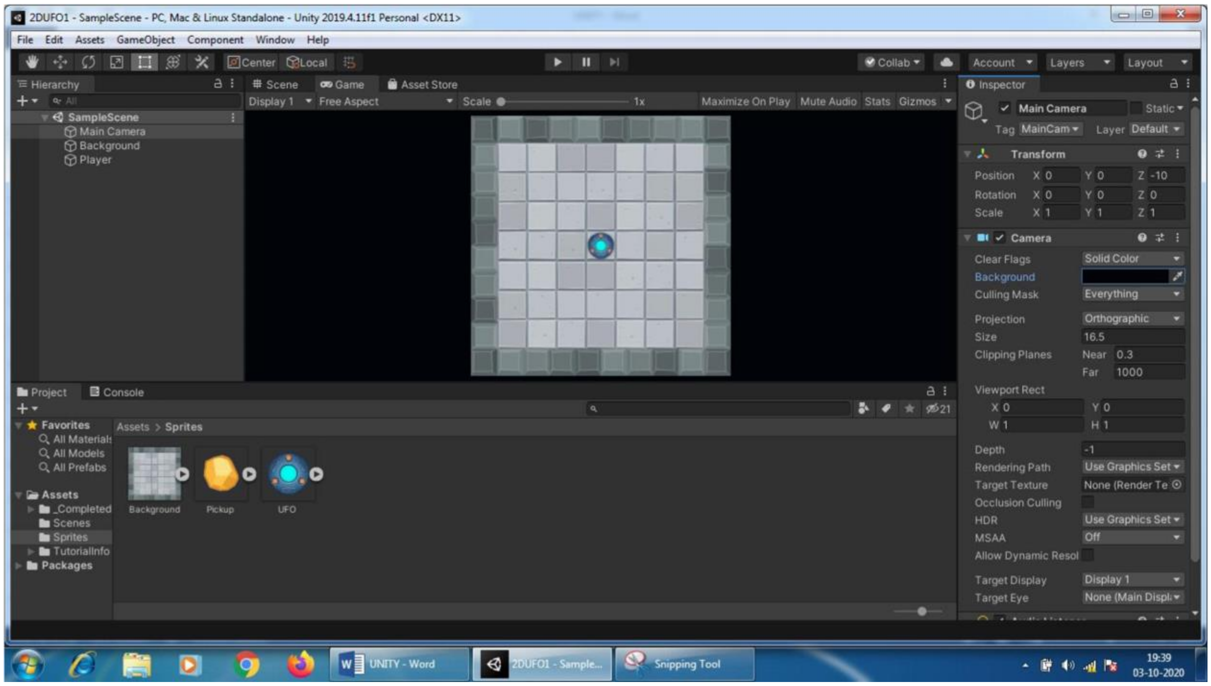
Sprites-Default
Shader: Sprites/Default

**Step 8 :** Select UFO and change the name to Player



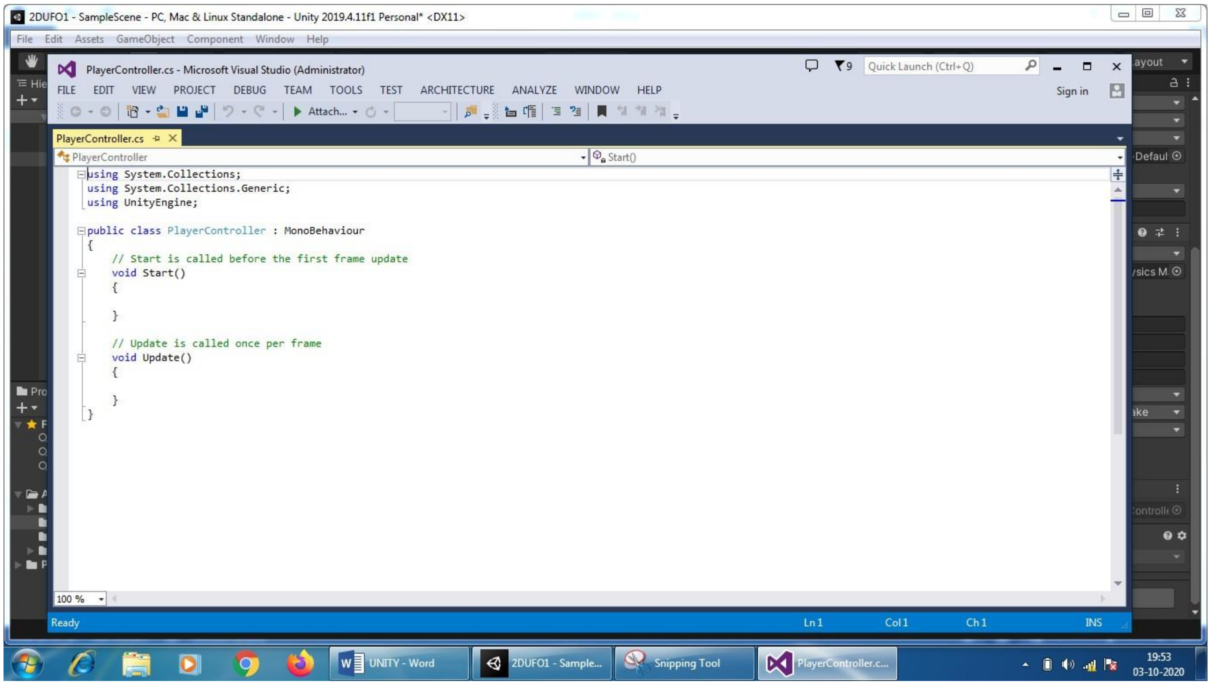
Inspector
Player Static
Tag Untagged Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Sprite Renderer
Sprite UFO
Color
Flip X Y
Draw Mode Simple
Mask Interaction None
Sprite Sort Point Center
Material Sprites-Default
Additional Settings
Sorting Layer Default
Order in Layer 0
Sprites-Default
Shader Sprites/Default
Add Component

**Step 9 :** Under Game tab, Click on Main Camera and set the size as 16.5 and background color.



Inspector
Main Camera Static
Tag MainCam Layer Default
Transform
Position X 0 Y 0 Z -10
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Camera
Clear Flags Solid Color
Background
Culling Mask Everything
Projection Orthographic
Size 16.5
Clipping Planes Near 0.3
Far 1000
Viewport Rect
X 0 Y 0
W 1 H 1
Depth -1
Rendering Path Use Graphics Set
Target Texture None (Render Te
Occlusion Culling
HDR Use Graphics Set
MSAA Off
Allow Dynamic Resol
Target Display Display 1
Target Eye None (Main Displ

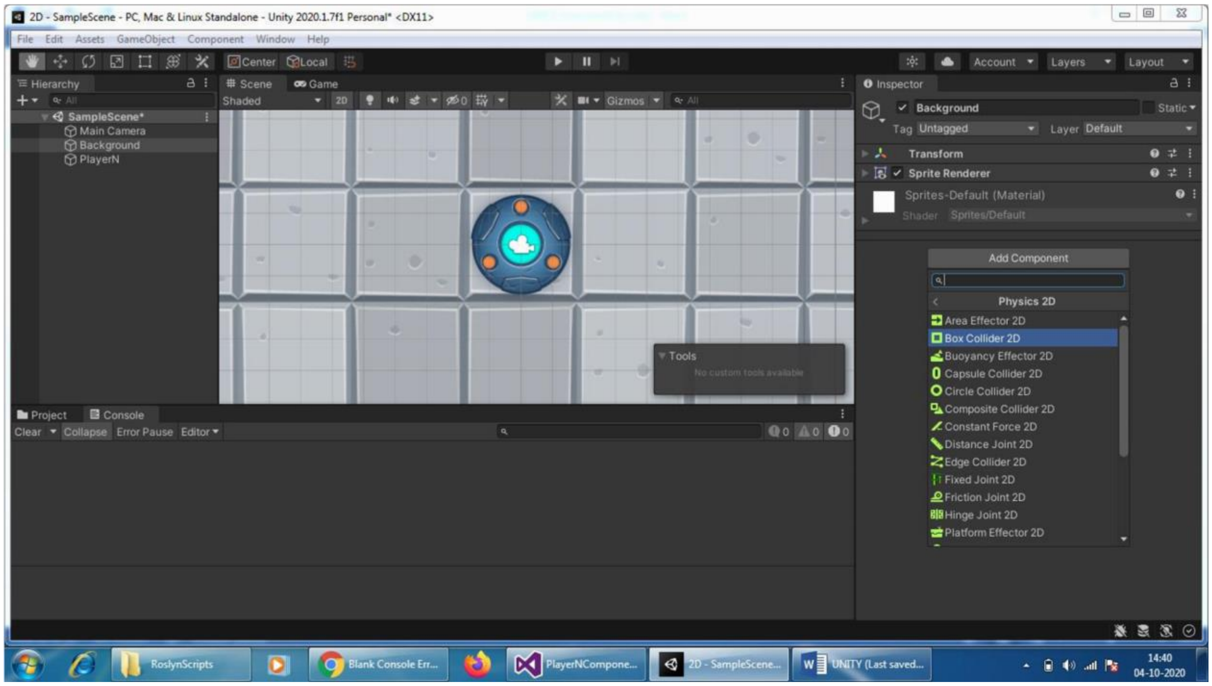
**Step 10 :** writing the following code



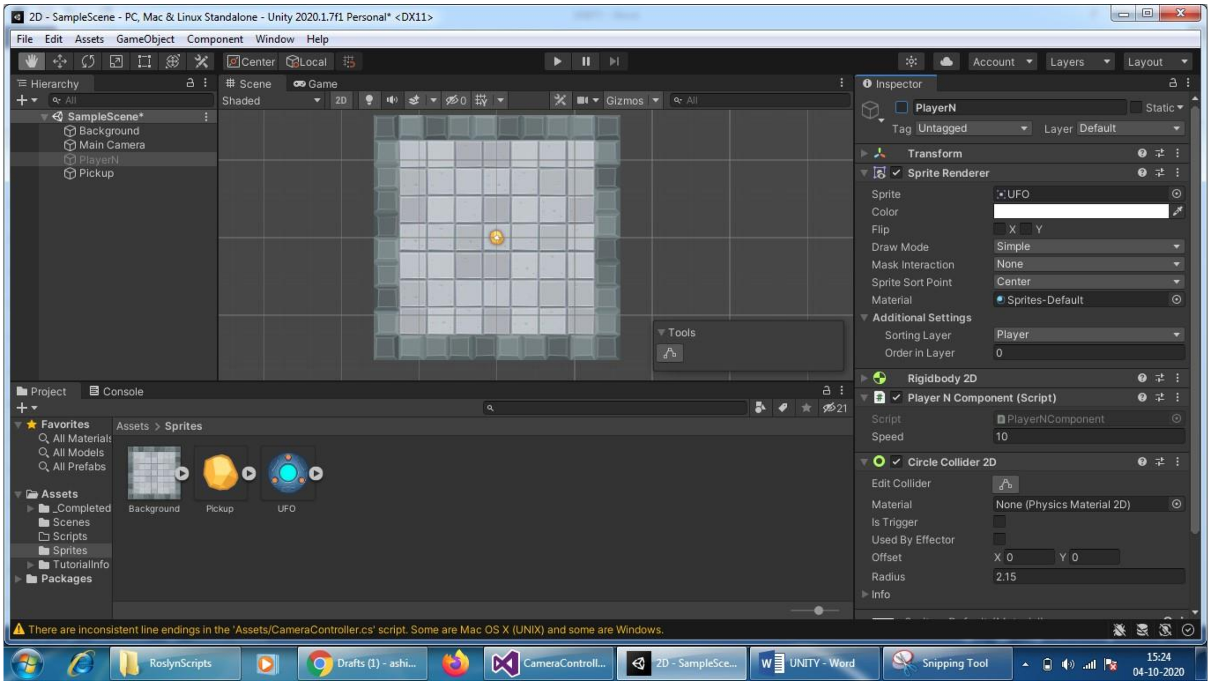
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

2DUFO1 - SampleScene - PC, Mac & Linux Standalone - Unity 2019.4.11f1 Personal* <DX11>
PlayerController.cs - Microsoft Visual Studio (Administrator)
FILE EDIT VIEW PROJECT DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Attach...
PlayerController.cs
PlayerController
Start()
Ready
Ln 1
Col 1
Ch 1
INS

**Step 11 :** Under Background - Click on Add Component - Physics 2D - Add Box Collider 2D



**Step 12 :** Since Pick up and Player are overlapping, we will TEMPORARY DESELECT the PLAYER object by clicking on it and UNCHECKING it.



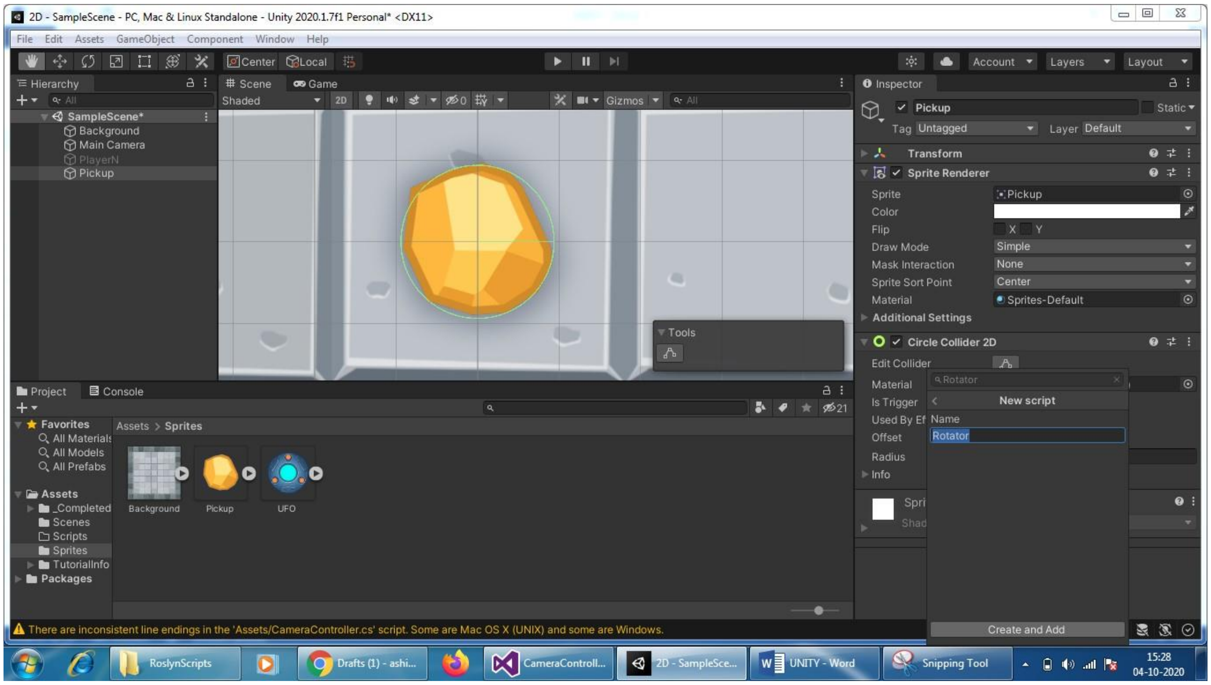
PlayerN Static

Tag Untagged Layer Default

Sprite: UFO
Flip: X Y
Draw Mode: Simple
Mask Interaction: None
Sprite Sort Point: Center
Material: Sprites-Default
Sorting Layer: Player
Order in Layer: 0
Script: PlayerNComponent
Speed: 10
Material: None (Physics Material 2D)
Is Trigger
Used By Effector
Offset: X 0 Y 0
Radius: 2.15

There are inconsistent line endings in the 'Assets/CameraController.cs' script. Some are Mac OS X (UNIX) and some are Windows.

**Step 13 :** Under Pick Up - Click on Add Component - Add New Script and Give name as Rotator and Edit the Script



Pickup Static

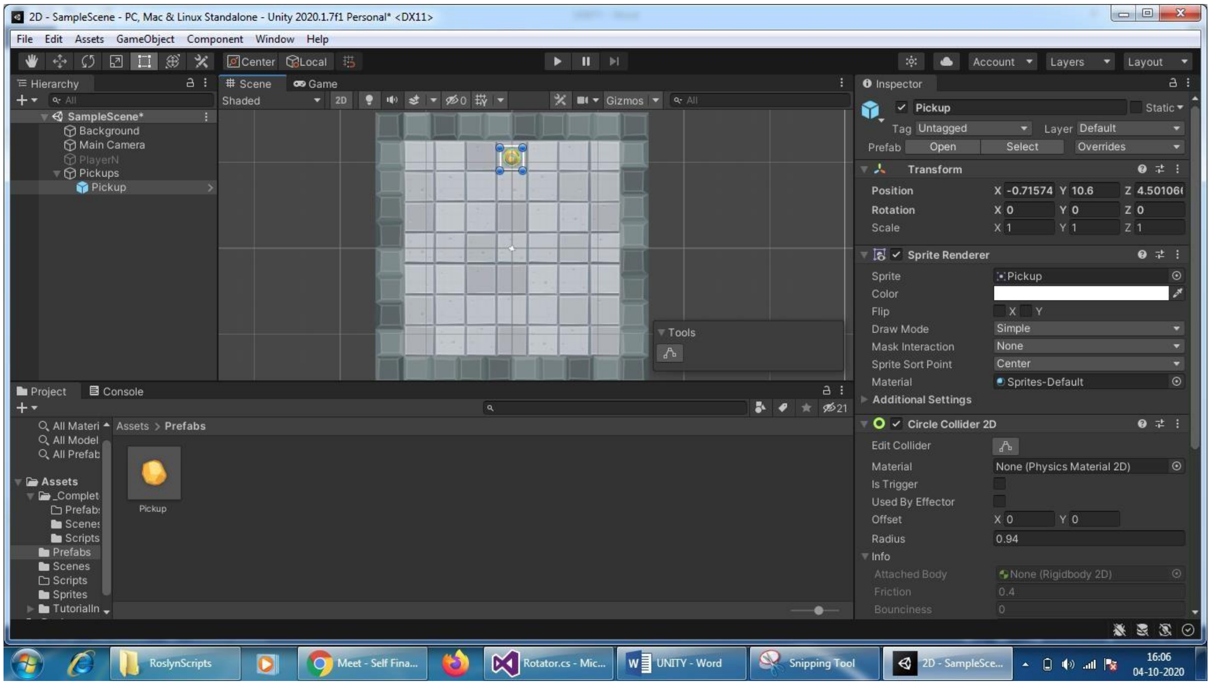
Tag Untagged Layer Default

Sprite: Pickup
Flip: X Y
Draw Mode: Simple
Mask Interaction: None
Sprite Sort Point: Center
Material: Sprites-Default

New script
Name: Rotator

There are inconsistent line endings in the 'Assets/CameraController.cs' script. Some are Mac OS X (UNIX) and some are Windows.

**Step 14 :** Click on Pickup □ Drag to the required position



Inspector

Pickup — Static

Tag Untagged — Layer Default

Prefab Open Select Overrides

Transform

| | | | |
|---|---|---|---|
| Position | X -0.71574 | Y 10.6 | Z 4.50106( |
| Rotation | X 0 | Y 0 | Z 0 |
| Scale | X 1 | Y 1 | Z 1 |

Sprite Renderer

| | |
|---|---|
| Sprite | Pickup |
| Color | |
| Flip | X Y |
| Draw Mode | Simple |
| Mask Interaction | None |
| Sprite Sort Point | Center |
| Material | Sprites-Default |

Additional Settings

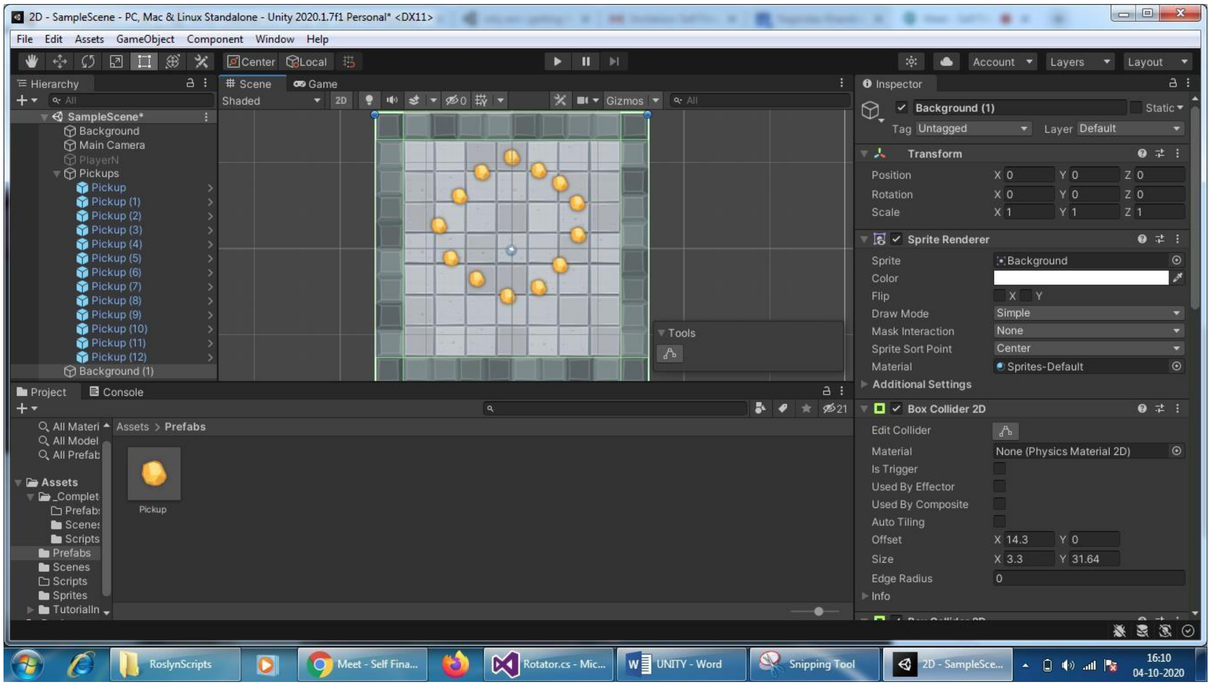
Circle Collider 2D

| | |
|---|---|
| Edit Collider | |
| Material | None (Physics Material 2D) |
| Is Trigger | |
| Used By Effector | |
| Offset | X 0 Y 0 |
| Radius | 0.94 |

Info

| | |
|---|---|
| Attached Body | None (Rigidbody 2D) |
| Friction | 0.4 |
| Bounciness | 0 |

**Step 15 :** Click on Edit □ Duplicate and drag and set, as many required for GAME



Inspector

Background (1) — Static

Tag Untagged — Layer Default

Transform

| | | | |
|---|---|---|---|
| Position | X 0 | Y 0 | Z 0 |
| Rotation | X 0 | Y 0 | Z 0 |
| Scale | X 1 | Y 1 | Z 1 |

Sprite Renderer

| | |
|---|---|
| Sprite | Background |
| Color | |
| Flip | X Y |
| Draw Mode | Simple |
| Mask Interaction | None |
| Sprite Sort Point | Center |
| Material | Sprites-Default |

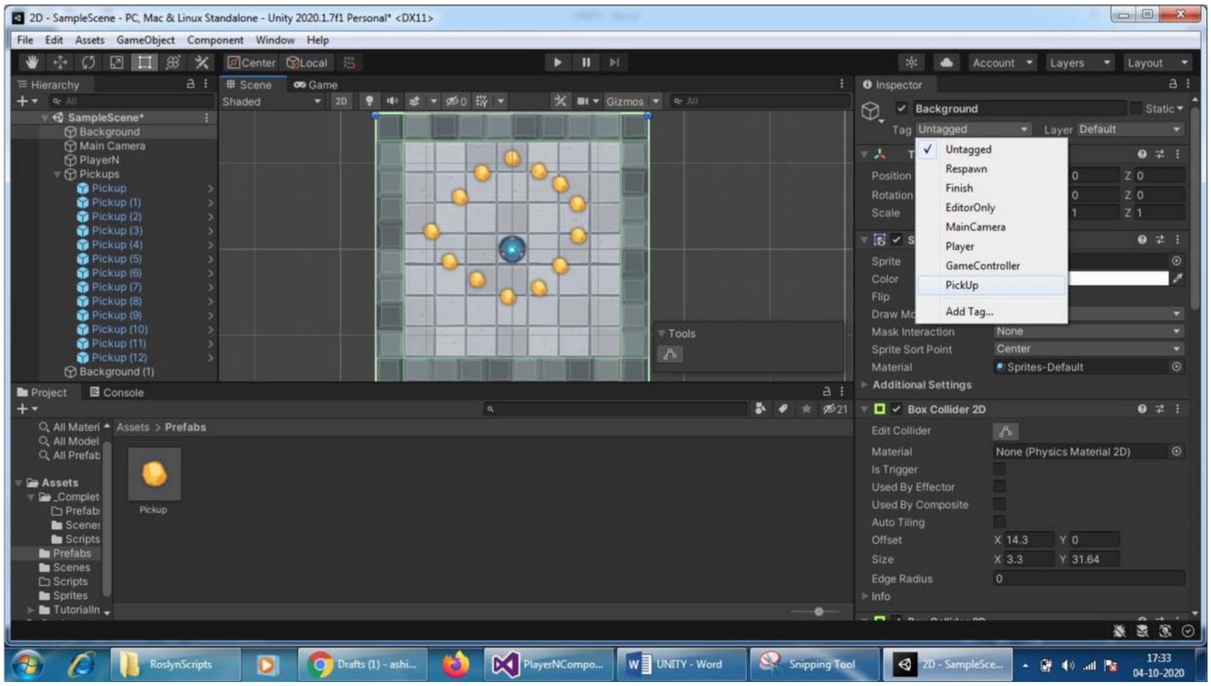
Additional Settings

Box Collider 2D

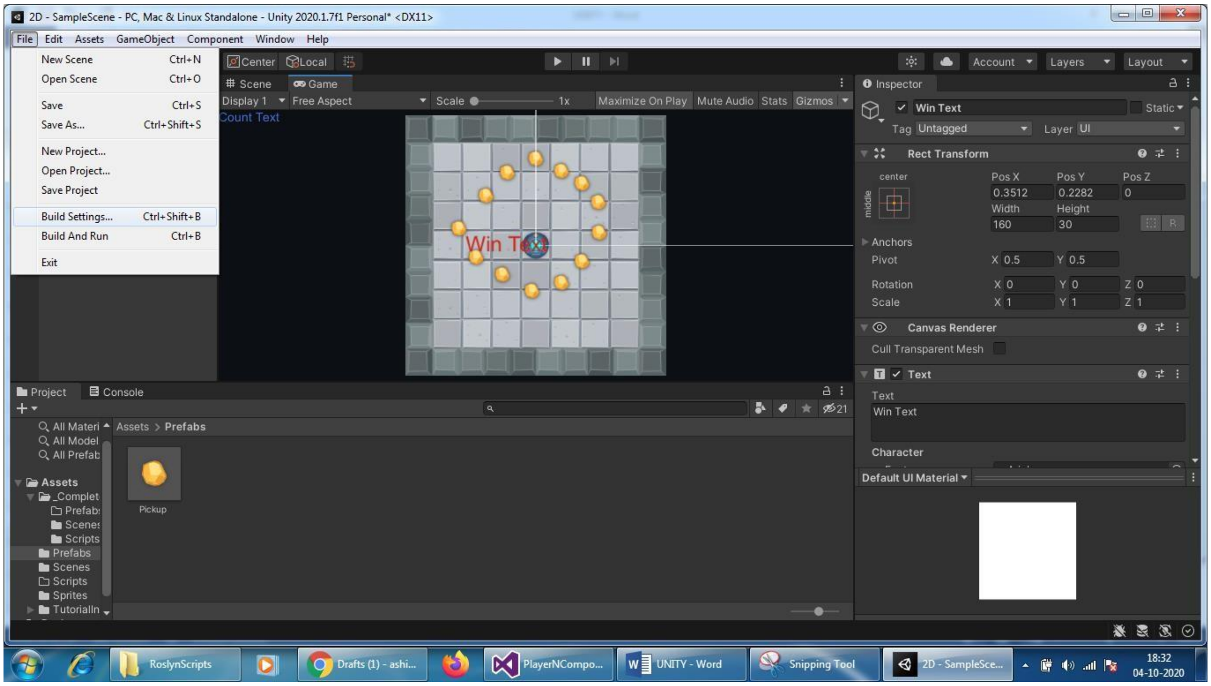
| | |
|---|---|
| Edit Collider | |
| Material | None (Physics Material 2D) |
| Is Trigger | |
| Used By Effector | |
| Used By Composite | |
| Auto Tiling | |
| Offset | X 14.3 Y 0 |
| Size | X 3.3 Y 31.64 |
| Edge Radius | 0 |

Info

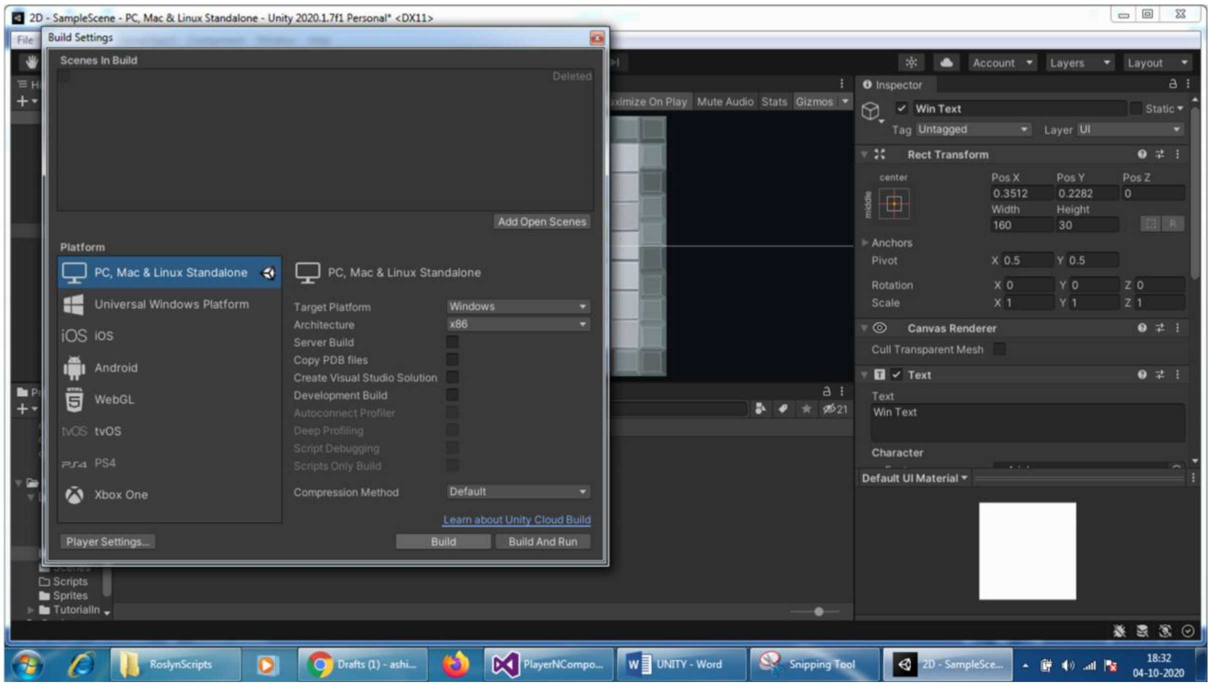
**Step 16 :** Click on PREFABS folder - click on PickUp - select Tag as PickUp



**Step 17 :** Click on File - Build Settings



**Step 18 :** default selection, Click Build


2D - SampleScene - PC, Mac & Linux Standalone - Unity 2020.1.7f1 Personal* <DX11>

Build Settings

Platform: PC, Mac & Linux Standalone

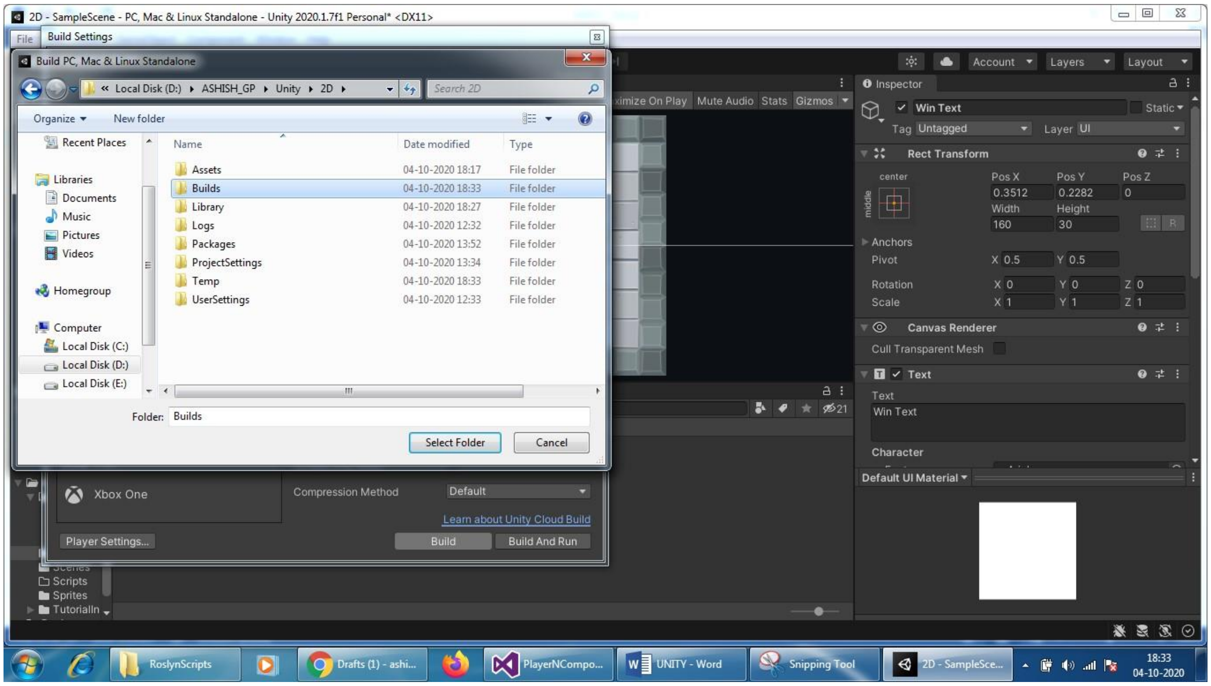
| | |
|---|---|
| Target Platform | Windows |
| Architecture | x86 |
| Compression Method | Default |

Inspector: Win Text, Tag Untagged, Layer UI

| | | | |
|---|---|---|---|
| Rect Transform | Pos X 0.3512 | Pos Y 0.2282 | Pos Z 0 |
| | Width 160 | Height 30 | |
| Pivot | X 0.5 | Y 0.5 | |
| Rotation | X 0 | Y 0 | Z 0 |
| Scale | X 1 | Y 1 | Z 1 |

Text: Win Text

**Step 19 :** In your same Same folder of PROJECT, Create a new folder named Builds and save this build in that


2D - SampleScene - PC, Mac & Linux Standalone - Unity 2020.1.7f1 Personal* <DX11>

Build PC, Mac & Linux Standalone

« Local Disk (D:) ▸ ASHISH_GP ▸ Unity ▸ 2D ▸

| Name | Date modified | Type |
|---|---|---|
| Assets | 04-10-2020 18:17 | File folder |
| Builds | 04-10-2020 18:33 | File folder |
| Library | 04-10-2020 18:27 | File folder |
| Logs | 04-10-2020 12:32 | File folder |
| Packages | 04-10-2020 13:52 | File folder |
| ProjectSettings | 04-10-2020 13:34 | File folder |
| Temp | 04-10-2020 18:33 | File folder |
| UserSettings | 04-10-2020 12:33 | File folder |

Folder: Builds

**Step 20 : 2 d ufo game done**