

## Practical 7

**AIM:-** Using a Unity 3d Software and making a 2d UFO game.

### ★ Light Class:-

- Defines a set of lighting properties

- Properties :-

- 1) `Light.Type` :- Retrieves or Sets the type of the light source.
- 2) `Light.Diffuse` :- Retrieves or Gets the diffuse color emitted by the light.
- 3) `Light.Enabled` :- Retrieves or Sets a Boolean value that enables or disables a set of lighting parameters within a device.
- 4) `Light.Direction` :- Retrieves or Sets the direction in which light is pointing in world space, as specified by a `Vector3` structure.



### Vector3 :-

Describes and manipulates a vector in three-dimensional (3D) space.



### Steps to Perform Practical :-

1. Open Unity. Select 2D, give Project Name and click on create a new project.
2. Click on "Window Menu" at the top, then select asset store or else Press "Ctrl + 9".
3. Search for 2D UFO Tutorial and click on it.
4. Click on download and click on import.
5. Then Save your scene by "Ctrl+S" or from file menu and give the name to the scene as "Main" and save it into "Scenes" folder.
6. Drop the Sprite named "Background" to Hierarchy Window" from "Sprites" folder in project window.
7. Move your cursor over the image and then click "F" key to see full game object.

- 8) Drag and drop the "UFO" Sprite to the "Hierarchy Window" and then go to the Inspector Window to the right and type "Player" on the top.
- 9) Click the "Background" in "Hierarchy Window" and then select "Sorting Layer" in the "Inspector Window" at right and then Select "Background" from the dropdown.
- 10) Similarly Select "Player" and select its "Sorting Layer" as Player.
- 11) Change the Scale value to  $X=0.75, Y=0.75, Z=1$  in "Inspector Window".
- 12) Click on "MainCamera" in "Hierarchy Window" and then go to "Inspector Window" and edit the value of "Size" to 16.5.
- 13) Click on "Player" object and then go to "Component Menu" and then Select "Physics 2D" and then Select "Rigidbody 2D".
- 14) Now select the "Player" object and then go to "Inspector Window" and then click on "Add Component" Select "New Script", provide the name to the script as "PlayerController" then Click "Create and add".
- 15) Now move the Script file from "Project Window" to the "Scripts" folder.
- 16) Now double click on the Script file to open it for editing.
- 17) Remove the start() and update() method, and type the code.
- 18) Click on your "Player" object and then go to "Inspector Window" and then Set "Gravity Scale" to 0 and "Speed" to 10, and also click on "Add Component" then type circle. Select then "Circle Collider 2D" and then edit the "Radius" value to 2.15.
- 19) Click on "Background" then "Add Component", then type Box. Select BoxCollider2.
- 20) Then edit the Offset  $X = 14.3$  Size  $X = 3.3$   
Offset  $Y = 0$  Size  $Y = 31.64$

- 21) Similarly copy and paste this BoxCollider 3 times like this and adjust it to fit.
- 22) Click on "Main Camera" then click on "Add Component" then select "New Script" type the name as "CameraController", then click "Create and Add". Again move the Script to "Scripts" folder.
- 23) Now double click on that Script and type the code.
- 24) Disable the "Player" Object temporarily by unchecking it in "Inspector Window".
- 25) Drop the "Pickup" Sprite to the "Hierarchy Window", add set the Sorting layer to "Pickups" the "Circle Collider" and adjust the radius to 0.94
- 26) Add a new Script to the "Pickup" and give the name "Rotator" and place it in "Scripts" folder.
- 27) Now double click on the Script file to open it and type the code.
- 28) Drag your "Pickup" from "Hierarchy Window" to the "Prefabs folder" in the "Project" Window.
- 29) Rename it to "Pickups" and then drag your "Pickup" Object onto the "Pickups" Object.
- 30) Duplicate the "Pickup" Object by going to "Edit Menu" at the top and selecting "Duplicate" option.
- 31) Click on Play Button to on the top center to run the scene.

## Practical 8

Aim:- Using a Unity 3D software and creating Space Shooter.

### \* System.Drawing Namespace :-

The System.Drawing namespace provides access to GDI+ basic graphics functionality.

#### \* Font Class:-

- Defines a particular format for text, including font face, size and style attributes.
- This class cannot be inherited.
- Constructor:-

Font(FontFamily, Single, FontStyle):-

Initializes a new font using a specified size and style.

#### • Method:-

Font.Draw.Text:- Draws formatted text.

#### \* Sprite Class:-

Provides methods and properties that simplify the process of drawing Sprites using Microsoft Direct3D.

#### • Methods:-

1) Begin :- Prepares a device for drawing Sprites.

2) Draw2D :- Adds a sprite to the list of batched Sprites. Used for presentation in 2D space.



#### Steps to Perform Practical:-

1. Open Unity, Select 2D, give Project Name & Click on create a new project.

2. Download the asset (From the link mentioned in playlist) and then Click on open in Unity. After Downloading, Click on Import.

3. Right Click on Hierarchy Window  $\rightarrow$  2D  $\rightarrow$  Sprites  $\rightarrow$  Square. Rename it as Background. Reset the Transform.
4. Drag & Drop Background3 from Red Nebula folder.
5. Right Click on Hierarchy Window  $\rightarrow$  2D  $\rightarrow$  Sprites  $\rightarrow$  Square. Rename it as Player. Drag & Drop Destroyer1 from Ships and Stations folder.
6. Change the pixels per unit to 200 & Order in layer to 1 of Destroyer1 by clicking on it.
7. 2D Object  $\rightarrow$  Sprite  $\rightarrow$  Square  $\rightarrow$  Rename it as Missile. Change its pixel per unit to 50 & Order in layer to 1.
8. Create a folder named Script to put all your C# files inside it.
9. Also create Prefabs folder for the object which will be needed for Instantiating.
10. Drag & Drop Missile & Alien of your choice inside the prefabs.
11. Click on Player  $\rightarrow$  Add Component  $\rightarrow$  PlayerController  $\rightarrow$  Create & Add.
12. Also add Box Collider2D to the Player, from Add Component.
13. Give Player Tag for our Player from Inspector Panel.
14. Type the code inside the PlayerController.cs
15. Drag & Drop the PlayerController.cs inside the script section of Player's Inspector Panel & give speed as 10.
16. Right Click on Player  $\rightarrow$  Create Empty  $\rightarrow$  Rename it MissileSpawnPos  $\rightarrow$  Reset its Transform. Set position of Y to 0.5.
17. Drag & Drop MissileSpawnPos inside the MissileSpawnPos section of player.
18. Select Missile of Prefab folder  $\rightarrow$  Add Component  $\rightarrow$  Missile  $\rightarrow$  Create & Add & Put it inside Script folder & Type the code.

19. Right Click on Hierarchy Window  $\rightarrow$  2D  $\rightarrow$  Sprite  $\rightarrow$  Square  $\rightarrow$  Rename it to Enemy  $\rightarrow$  Reset its transform & give it's tag as Enemy from Inspector Panel & Order in layer to 1. Select Align of your choice inside the Sprite section. Give Pixels per Unit 76 & Order in layer 1. Rotate it to 180° for Z-axis.
20. Add BoxCollider 2D to Enemy.
21. Right Click on Hierarchy Window  $\rightarrow$  Create Empty  $\rightarrow$  Rename it as Game Manager.
22. Follow Through Add Component, Add GameManager Script & type the code, by adding it inside Scripts folder.
23. Follow Step 21 & 22 for adding Event Controller Script.
24. Add BoxCollider 2D to missile, also add RigidBody 2D to it. Keep it's gravity scale to 0.
25. Add MuzzleFlash for Sparkle effect during shooting of collision.
26. Click on Play Button on the top center to run it.

## Practical 9

**AIM:-** Create a simple rolling ball game that teaches you many of the principles of working with Unity.



**TextureLoader.FromFile (Device, String, Int32, Int32, Usage, Format, Pool, Filter, Int32) method:-**

- Creates a texture from the file.

- Parameters:-

1) device :-

A Device object that represents the device is associated with the texture.

2) SrcFile :-

String that specifies the file name.

3) width :-

Width of the texture in pixels. If this value is zero, the dimensions are taken from the file.

4) height :-

Height of the texture in pixels. If this value is zero, the dimensions are taken from the file.

5) miplevels :-

Number of mip levels requested. If this value is zero, a complete mipmap chain is created.

6) usage :-

Zero or Usage.RenderTarget, or Usage.Dynamic. Setting this flag to Usage.Render Target indicates that the surface will be used as a render target.

7) format :-

Member of the format enumerated type that describes the requested pixel format for the cube texture.

8) pool :-

Member of the pool enumerated type that describes the memory

CLASSTIME
Date :
Page:

class into which the cube texture should be placed.

9) filter:-

One or more filter flags that control how the image is filtered.

10) mipFilter:-

One or more filter flags that control how the mips are filtered.

11) colorkey:-

An Int32 value to replace with transparent black, or 0 to disable the Color key.



Steps to Perform Practical:-

- 1) Open Unity, Select 3D, give Project Name and Click on create a New Project.
- 2) Click on Main Camera  $\rightarrow$  Game Objects  $\rightarrow$  3D Objects  $\rightarrow$  Plane
- 3) Then again, Game Object  $\rightarrow$  3D Object  $\rightarrow$  Sphere.
- 4) Create a new folder inside asset & name it BallColour.
- 5) Select BallColour & go to Inspector Panel  $\rightarrow$  Main maps  $\rightarrow$  Albedo (Select a Colour) drag and drop BallColour to the Ball on the scene.
- 6) Similarly, create a new material inside asset and name it BackgroundColour and change its colour from Albedo.
- 7) Click on Sphere  $\rightarrow$  Inspector Panel  $\rightarrow$  Add Components  $\rightarrow$  New Script  $\rightarrow$  Player Controller  $\rightarrow$  Add New Components.
- 8) Double click on PlayerController and write the code.
- 9) Click on Sphere of Hierarchy Window  $\rightarrow$  Component  $\rightarrow$  Physics  $\rightarrow$  Rigidbody
- 10) Hierarchy  $\rightarrow$  Sphere  $\rightarrow$  Inspector Panel  $\rightarrow$  Rigidbody  $\rightarrow$  Speed  $\rightarrow$  10. Click on Save.
- 11) From Main Camera, Select Inspector Panel  $\rightarrow$  Add Component  $\rightarrow$  New Script  $\rightarrow$  CameraController  $\rightarrow$  Add Component. Click on C# file and write the code.
- 12) Again, click on Main Camera, go to Inspector Panel  $\rightarrow$  CameraController

→ player Drag & Drop Sphere from Hierarchy Window inside the Player.

- 13) Click on Game Object → Create Empty. Name it Wall.
- 14) Go to Game Object → 3D Object → Cube. Rename it Wall1.  
Similarly add Wall2, Wall3, Wall4.
- 15) Change the axis of all 4 Walls.
- 16) Go to Sample Scene → Game Objects → Create Empty. Rename it to Pickup.
- 17) Again go to Game Objects → 3D Object → Cube → Inspector Panel → Transform
 

Position	$x = 0.78$	$y = 0.78$	$z = 2.09$
Rotation	$x = 0^{\circ} 45$	$y = 45$	$z = 45$
Scale	$x = 0.5$	$y = 0.5$	$z = 0.5$
- 18) Right Click on Asset → Create → Material. Rename that Material as Pickup Color.
- 19) Change its colour from Albedo, then drag and drop it on Cube.
- 20) Rename the cube as 'Pickup1'.
- 21) Pickup1 → Inspector Panel → Tag → Add Tag → Pickup → Enter.
- 22) Again Pickup1 → Inspector Panel → Tag → Pickup.
- 23) Select Pickup1 and drag it inside 'Pickups'.
- 24) Pickup1 → Inspector Panel → Add Component → Rotator. Double click on Rotator & type the code.
- 25) Again Select Pickup1 → Components → Physics → Rigidbody.
- 26) Follow same procedure for Pickup 2, 3, 4, 5, 6 from Step 17 to 25.
- 27) Make Sure is Trigger is checked of all the pickups.
- 28) Go to Hierarchy Window → + → UI → Canvas. Click on Canvas → + → UI → Text. Rename it as "CountText".
- 29) Similarly add "WinText".
- 30) CountText → Inspect → Character → FontSize → 20, Also, color of your side. Select

- 31) Follow Similar Step for "WinText".
- 32) Uncheck the Gravity of all the Pickup from there respective Inspector Panels.
- 33) Select Sphere, Drag & Drop context of WinText, inside the Player Controller. (from canvas)
- 34) Click on Play Button on top center to run the Scene.

## Practical 10

### AIM:- Creating AR Content with Vuforia

Augmented Reality and Virtual Reality are reality technologies that either enhance or replace a real life environment with a simulated one.

- Augmented Reality (AR) augments your surroundings by adding digital elements to a live view, often by using the camera on a smartphone.
- Virtual Reality (VR) is a completely immersive that replaces a real-life environment with a simulated one.

#### ★ Steps to Perform Practical:-

- 1) Create a new project in unity and select 3D and name it AR using vuforia.
- 2) Go to the link "developer.vuforia.com" & Register. Enter the details asked & click on create a account.  
Verify your details by going to the link on email.  
Login to vuforia account.
- 3) Go to downloads → Sample.  
Click on download from Unity Store.  
A new tab opens of Unity Asset Store.
- 4) Go to Vuforia Engine SDK → Add to My Assets → Open in Unity.
- 5) A packet manager appears. Click on Import.
- 6) Now go to assets Store and select Vuforia Core / Sample → Add to My Asset → Open in Unity → Import.
- 7) To verify whether vuforia is imported successfully, right click on Hierarchy Window Panel. If you see vuforia engine in the dropdown list, it means that vuforia is imported successfully.
- 8) Delete the "Main Camera" from the Scene Right Click on Hierarchy Panel → Vuforia Engine → AR Camera.
- 9) In the Project Window - go to Asset/Resources and Double Click

on "VuforiaConfiguration.asset", to view its property in the Inspector Panel.

- 10) Click on Add License on the Inspector Panel. vuforia.com opens
- 11) Click on License Manager → Get Basics.
- 12) Keep the license name same as the Project name i.e AR using vuforia. Check the terms and conditions. Click on confirm.
- 13) Click on license & copy the license key and paste it in the app license key.
- 14) Go to vuforia → Target Manager → Add Database & Write your Database Name and Select Device → Create.
- 15) After creating database, open the database & click on Add Target.
- 16) Click on Image → Browse and Select an Image → Width: 0.5 → Name: Stones-Image → Add.
- 17) Repeat Step 15 & 16 until you get 4-5 star ratings.
- 18) Click on download database → Unity Editor → Download. Once it downloads you can get the file called hellovuforia.unity package. Double click on this file to open the unity, now you'll get a prompt, Click on Import.
- 19) Go to Asset /Resources & double click on vuforiaconfiguration.asset → Inspector Panel → Databases. There you would see hellovuforia listed. This means that you have successfully imported database in your Unity Project.
- 20) Right Click on Hierarchy Panel → Vuforia Engine → ImageTarget. Select the Image Target to view this property in the Inspector Panel.
- 21) Go to Image Target behaviour Script. Right Click on ImageTarget → 3D Object → Cube.  
Scale :- X:-0.2 Y:-0.2 Z:0.2

Name:- Sahil Mahendra Chavhan

Rollno:- 228621

classmate
Date :
Pogo :

- 22) Move "your box Slightly up to your Surface to make it appear levitating.
- 23) Click on Play button to run the Scene on the top center using your web Cam.