# Workshop 12 Lab 1

In this activity, we are going to implement a few simple Vertex and Fragment Shader. Create a new Unity3D project and perform the following tasks.

1. Install Core RP and Universal RP

From Window->Package Manager->Packages: Unity Registry-> Install Core RP and Universal RP

Follow the instruction in the link to configure UPR into an existing Project

https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@10.8/manual/InstallURPIntoAProject.html

1. First Shader

Create a Shader folder.

Create a standard Surface Shader in Shader folder and name it as TestShader.

Create->Shader->Standard Surface Shader

Replace the content with the FirstShader code.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Create a new Material, name it as TestMaterial.

Create a new Sphere and Capsule object in the scene.

Attached the Material to the 2 objects. (Drag and drop the TestMaterial into Sphere and Capsule Component).

Change the Material shader.

Change the base colour in the shader.

|  |  |
| --- | --- |
| Graphical user interface, text, application, email  Description automatically generated | A picture containing shape  Description automatically generated |

1. Vertex Position Shader

Create a new Shader and name it VertexPositionShader.

Copy the FirstShader into VertexPositionShader.

Modify the following.

Change the name

A picture containing text

Description automatically generated

Change the vertex shader.

Graphical user interface, text, application, email

Description automatically generated

Change the Material shader.

Shape

Description automatically generated

Include \_Time in the vertex shader.

The effect can only be seen when the scene is executed.

Graphical user interface, text, application, email

Description automatically generated

1. Fragment Colour Shader

Create a new Shader and name FragColourShader.

Copy the FirstShader into FragColourShader.

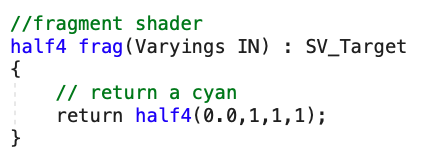
Modify the following.

Change the name.

A picture containing text

Description automatically generated

Always return a cyan in the fragment shader



Change the Material shader.

Chart

Description automatically generated

1. Fragment Blinking Shader

Create a new Shader and name FragBlinkingShader.

Copy the FirstShader into FragBlinkingShader.

Modify the following.

Change the name.

Include \_Frequency in Properties

Text, letter

Description automatically generated

Include \_Frequency in CBUFFER

Text

Description automatically generated

Modify the fragment shader

Text

Description automatically generated

Change the Material shader.

Run the scene and observe that the colour intensity is changing with time.

1. Texture Shader

Create a new Shader and name TextureShader.

Copy the FirstShader into TextureShader.

Modify the following.

Change the name.

A picture containing logo

Description automatically generated

Add \_Scale in Properties

Logo, company name

Description automatically generated

Add in positions in Varyings

Graphical user interface, text, application

Description automatically generated

Change to \_Scale in CBUFFER.

Text

Description automatically generated

Include positions in vertex shader.

Graphical user interface, text, application

Description automatically generated

Modify the fragment shader

Text

Description automatically generated

A picture containing shape

Description automatically generated

Make the scale dependent on sine of time for animation.

Text

Description automatically generated

End.