<https://docs.unity3d.com/Manual/ShadersOverview.html>

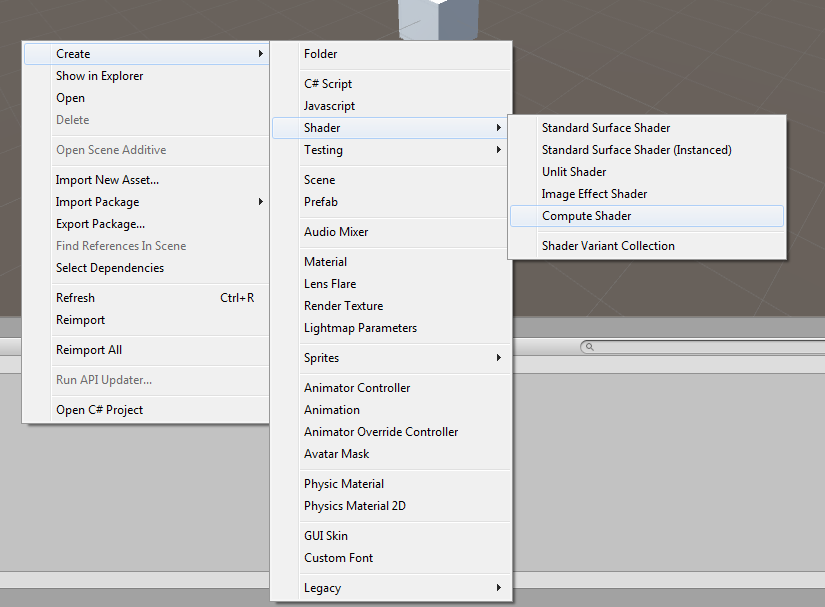
!You can’t use shader directly you have to attach it to some Material!

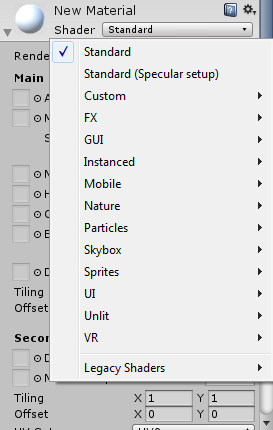
Shaders in Unity can be written in one of three different ways:

### Surface Shaders - written in **Cg/HLSL**

### Vertex and Fragment Shaders - written in **Cg/HLSL**

### Fixed Function Shaders - written in a language called **ShaderL**



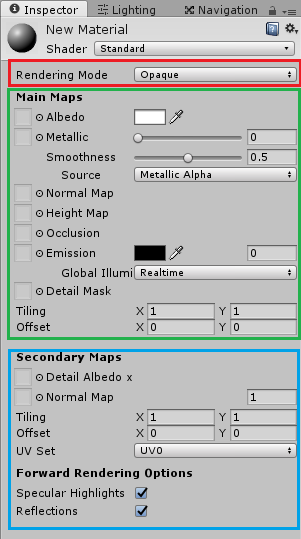
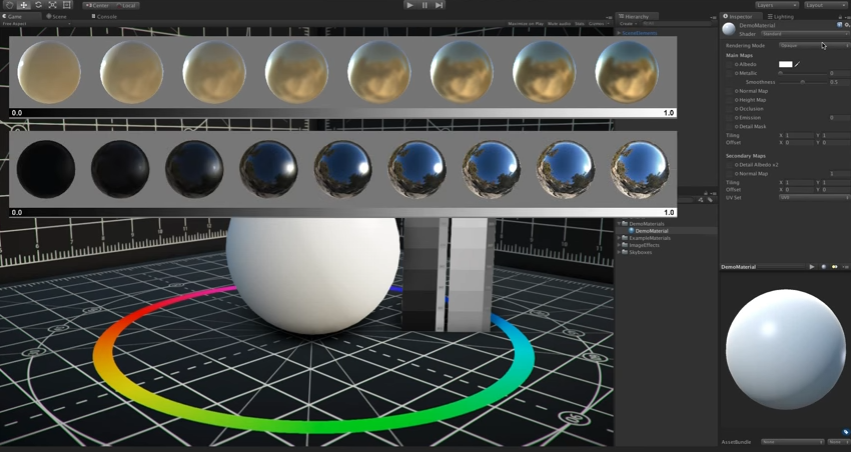


Create 3D object cube -> Mesh Renderer -> Default Material -> Standard Shader

The Material responsible to combine the Texture and to calculate the color with all the lights in the scene.

**Standard Shader - (template for vertex + fragment shader)**

By default all newly created material are created with the **Standard Shader** (Metallic) you can change to Standard Specular setup.  
Standard shader is very optimized - all properties that are not in use are discarded in the build.



There are 3 section to the standard shader:

* Rendering Mode
* Main Maps
  + Each property can be colored with a texture map
* Secondary Maps

**Image Effect Shader**

Pure Fragment Shader - a full screen effect (Not attached to any object)

We can create Image Effect shader which is very close to empty shader template and we will attach it to the camera, that will provide us a full screen effect =>

it’s gonna change the way the entire scene is rendered

Youtube:

[Unity Rendering Pipeline](https://www.youtube.com/watch?v=cJbXlONSpTE)

[Unity Tutorial: A Practical Intro to Shaders - Part 1](https://www.youtube.com/watch?v=C0uJ4sZelio)

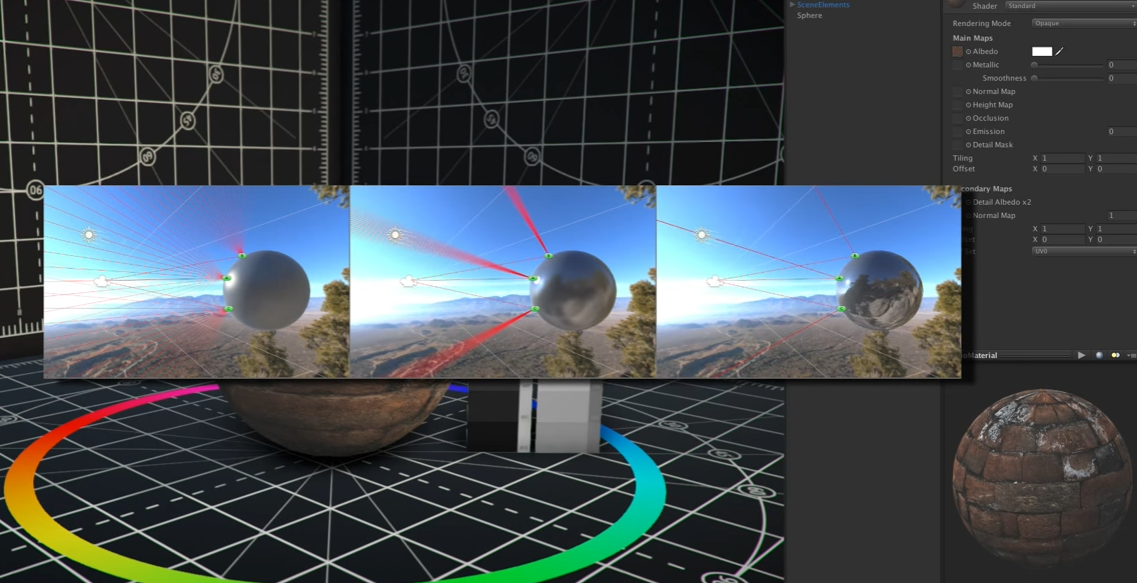
[Can i use of HLSL or GLSL shaders in unity?](http://answers.unity3d.com/questions/7925/can-i-use-of-hlsl-or-glsl-shaders-in-unity.html)

<https://docs.unity3d.com/Manual/SL-SurfaceShaderLightingExamples.html>

<https://docs.unity3d.com/Manual/SL-SurfaceShaderLighting.html>

<https://docs.unity3d.com/Manual/SL-GLSLShaderPrograms.html>

Smoothness



**Notes**

Unlit shader - closest to empty shader

Image Effect shader - closest to empty shader but a little bit closer to image (good for camera effect)

There is some Unity UI default shaders - for image on Canvas… and more

Unity path to the shader a time parameter - how much time is passed from the moment the program is started running: **\_Time[1] //float 4 accept number 0-3**

The “Property” section add some visual parameters to the GUI

But you can’t use it directly from your prog,

you’ll get an error because the declaration used only for Unity itself.

So you’ll need to declarate it again in the #CGPROGRAM section and if the property and program variable have the same names Unity link them automatically.

All shader work is done between “CGPROGRAM” =>>> “ENDCG”

Next we will define 2 functions “vert” and “frag”

#pragma vertex vert

#pragma fragment frag

Ideally the vert function will run once for each vertex

And the frag function will always run once per pixel.

<https://docs.unity3d.com/Manual/SL-VertexFragmentShaderExamples.html>

<http://wiki.unity3d.com/index.php/Shader_Code>

[Get light info in shader](https://forum.unity3d.com/threads/light-info-in-shader.2379/)

**UNITY\_LIGHTMODEL\_AMBIENT**

struct v2f

{

float2 uv : TEXCOORD0;

fixed4 diff : COLOR0; // diffuse lighting color

float4 vertex : SV\_POSITION;

};

fixed4 frag (v2f i) : SV\_Target

{

// sample texture

fixed4 col = tex2D(\_MainTex, i.uv);

// multiply by lighting

col \*= **i.diff**;

return col;

}