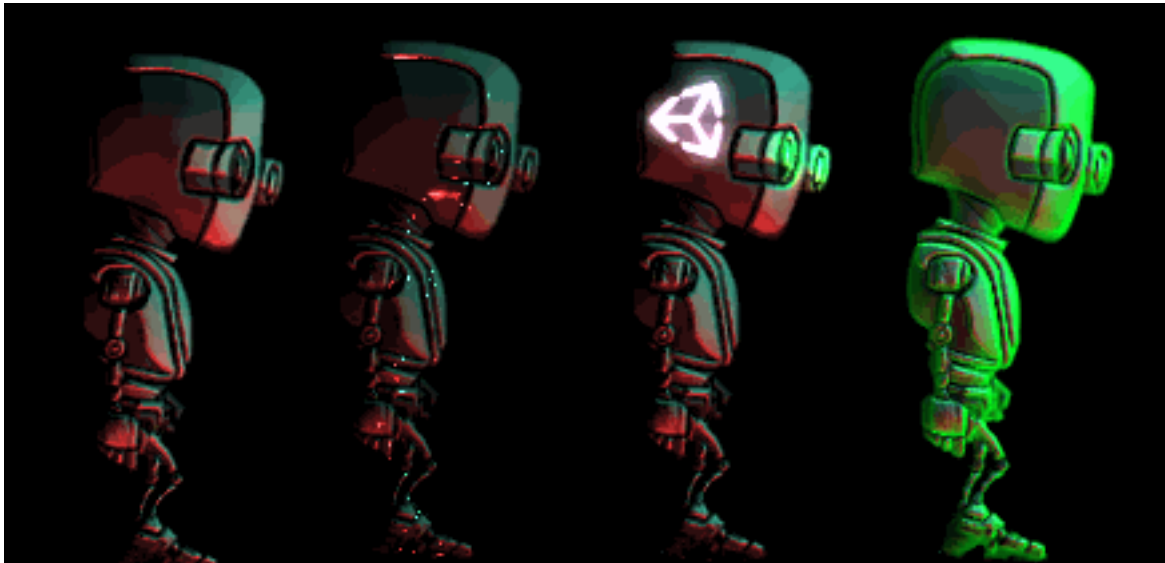

Unity Sprite Uber Shader

An Uber Shader for Unity specialised in rendering Alpha Blended objects like sprites. It has a ton of features and a nifty Shader GUI meaning its very easy to use. It's also modular so features you don't use don't impact performance at all. Because it supports all its feaures in a one pass Vertex lit mode it can be used on skeleon based animation or sprites with submeshes (eg Spine animations) whilst preserving soft alpha edges, this includeds per pixel effects like normal mapping and specular.



Above shows single pass normal mapping, specular, emission and rim lighting on Unity's example robot sprite.

Note: Material requires a mesh with Tangents.

Lighting Mode

VertexLit

Blend Mode

StandardAlpha

Renderer Queue

0


Culling


Off


Pixel Snap


☐



Main Maps

 ☐ Albedo

 ☐ Normal Map

 ☐ Diffuse Ramp

 ☐ Alt Albedo

0

Tiling

X 1 Y 1


Offset

X 0 Y 0

Specular

Enable Specular

☒

 ☐ Metallic Gloss Map


Smoothness



1

Emission

Enable Emission

☒

 ☐ Emission

1

Emission Power

1

Depth

Write to Depth

☒

Depth Alpha Cutoff

0

Normals

Use Fixed Normal

☒

Fixed Normal Space

ViewSpace

Fixed Normal Direction

X 0 Y 0 Z 1

Adjust Backface Tangents

☐

Shadows

Shadow Alpha Cutoff

0.1

Ambient Spherical Harmonics

Enable Spherical Harmonic

☒

Fog

Enable Fog

☐

Color Adjustment

Enable Color Adjustment

☐

Rim Lighting

Enable Rim Lighting

☐

Features

Lighting

The shader supports both multi-pass Pixel Lighting and single-pass Vertex Lighting as well as simple unlit.

Normal/Bump Mapping

The shaders support Normal maps even when using single pass Vertex Lighting. For Normals maps to work you need a mesh with Tangents.

Blend Modes

The shaders support easily changing the blend mode between Opaque, Standard Alpha, Pre Multiplied Alpha, Additive, Soft Additive, Multiply and Multiply x2 blending modes.

Depth Writing

The shaders allow you to optionally write to depth using a Cutoff alpha value. Included are also several shaders that can be used by a camera to render a Depth or DepthNormals texture with soft edged Depth rendered for objects using these shaders that don't write to depth. Meaning you can have Post Effects like Depth of field or Ambient Obscurance working with soft edged sprites.

Specular

The shaders optionally support physically based BRDF specular in both Pixel Lit and single-pass Vertex Lit lighting modes. This is based off the metallic specular in Unity's Standard Shader. (It uses a Metallic Gloss Map and a Smoothness Value).

Emission

The shaders optionally support an Emission map in both Pixel Lit and single-pass Vertex Lit lighting modes. This again mimics Unity's Standard Shader.

Fixed Normals

The shaders optionally support using a Fixed Normal instead of Mesh Normals. This can be usefull for rendering objects that don't have normals (like a TextMesh for example). Also sprites can use a view-space Fixed Normal to make it look less flat than using its own normal when it rotates around the Y axis. (it stop's it looking like a sheet of paper because it tricks the lighting into thinking its still facing the same way). When using a fixed normal the shaders can also automatically flip the tangents of a mesh if its rendering the wrong side of it, meaning normal maps can work for both front and back faces.

Shadows

Shadows are supported in all lighting modes (including unlit) using an alpha cutoff value.

Fog

Fog is optionally supported and works correctly with all blend modes.

Gradient based Ambient lighting

The shaders optionally support using Spherical Harmonics for ambient lighting. In Vertex Lit mode, the Spherical Harmonics is approximated from the ground, equator and sky ambient colors.

Color Adjustment

The shaders allow optional adjustment of hue / saturation and brightness as well as applying a solid color overlay effect this can be used for things like damage effects etc.

Rim Lighting

The shaders allow optional camera-space rim lighting in both lighting modes.

How To Use

Copy the SpriteShaders folder to anywhere inside your Unity Assets folder. On your objects material click the drop down for shader and select either Sprite (Pixel Lit), Sprite (Vertex Lit) or Sprite (Unlit).

Material Property Names

Below are some of the Material Property Names/IDs that are commonly animated from code. You can use these in conjunction with Unity's MaterialPropertyBlock API (MaterialPropertyBlock, Renderer.SetPropertyBlock) to override property value per renderer. You can also use these with Shader.PropertyToID.

Main Maps

Albedo/Main Texture: (Texture2D) `_MainTex` (Color) `_Color`

Normal Map: (Texture2D) `_BumpMap`, (float) `_BumpScale`

Diffuse Ramp: (Texture2D) `_DiffuseRamp`

Secondary Albedo/Blend Texture: (Texture2D) `_BlendTexture`, (float) `_BlendAmount`

Specular

Metallic Gloss Map: (Texture2D) `_MetallicGlossMap`, (float) `_GlossMapScale`

Smoothness: (float) `_Glossiness`

Emission

Emission: (Texture2D) `_EmissionMap`, (Color) `_EmissionColor`

Emission Power: (float) `_EmissionPower`

Color Adjustment

Overlay Color: (Color) `_OverlayColor`

Hue: (float) `_Hue`

Saturation: (float) `_Saturation`

Brightness: (float) `_Brightness`

Rim Lighting

Rim Color: (Color) `_RimColor`

Rim Power: (float) `_RimPower`

You can see a more complete list of the material properties by selecting the relevant Shader asset in your Project View and looking at its Inspector. Note that the list of properties may not show up in the

inspector if your project has a script that overrides Shader asset inspectors. Some third-party Shader editors such as ShaderForge do this.

Known Issues

When using Unity's Sprite Renderer class, its tangents are incorrect when you use the flip X or flip Y flags. This results in incorrect lighting when using normal maps. If you want to use these shaders with a normal map on a Sprite Renderer then either set a negative scale on the objects transform or rotate it instead (with back face culling turned off).