Opsive Home Games Assets Forum About

# Glow Effect Documentation

Overview | Documentation | Support

Glow Effect is designed to be as flexible and efficient as possible. It is also optimized to be able run on mobile devices. This package contains four different methods of generating glow:

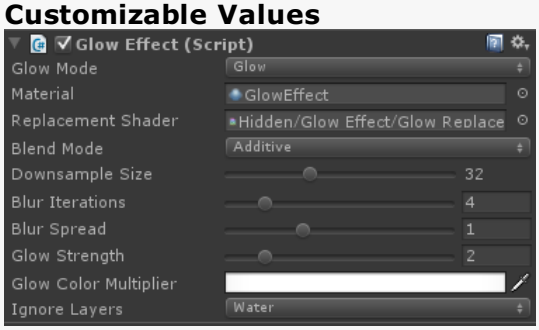
- **Standard Glow** – the standard way of generating glow by internally using a separate camera and replacement shaders. This method looks the best but it takes the most resources.
- **Simple Glow** – similar to Standard Glow except it uses less draw calls and less memory. This effect does everything in one pass instead of multiple passes to generate a blur for the glow.
- **Alpha Glow** – recommended method for mobile devices. This method uses the object's alpha channel in order to determine the amount of glow to apply. This method looks comparable to standard glow however it uses the object's alpha channel. This method does not require a separate camera.
- **Simple Alpha Glow** – similar to Alpha Glow except it further reduces the amount of draw calls and memory by generating the glow in one pass. This method looks the worst out of the four and should be used as the last resort.

## Setup

Adding the glow effect can be done in as little as three steps:

- Drag the script GlowEffect.cs onto your camera as a new component. Assign the glow material and the glow replace shader to the files included in the Glow Effect directory. Alternatively you can use the camera prefab included.
- Attach a material which has a glow shader to each object that you want to glow. If you are using the alpha channel to provide glow, alpha values of 1 means full glow and 0 means no glow.
- If you are targetting a mobile device, ensure you are using 32-bit depth buffer (set within player prefs)

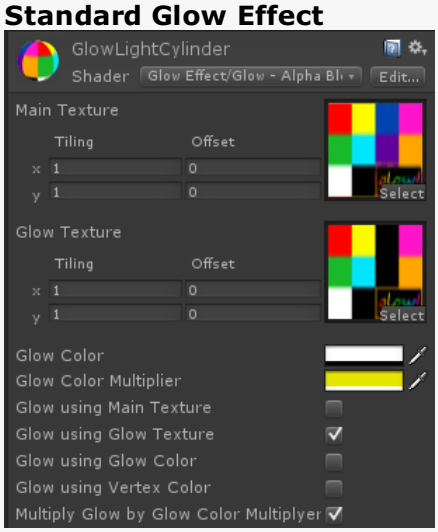
## Customizable Values




The following variables can be modified to customize the way your glow effect looks:

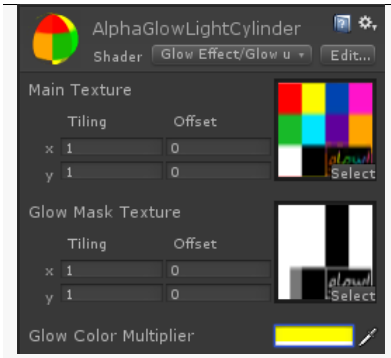
- glowMode - enumerator to specify the glow type. Choices are Glow, Alpha Glow, Simple Glow, or Simple Alpha Glow.
- blendMode - enumerator used to specify the blend type. Options are Additive, Multiply, and Screen
- downsampleSize - The size that the render texture is resized to when performing the glow. Must be a power of two
- blurIterations - The number of times the glow texture should be blurred. The more blur iterations the wider the glow. This value is only used if useSimpleGlow is false.
- blurSpread - The distance of the samples taken for the blurred glow. Too big of a value will cause noise in the blur. This value is only used if useSimpleGlow is false.
- glowStrength- Multiplies the glow color by this value.
- glowColorMultiplier - Multiplies the glow color by this color.

## Standard Glow Effect




The source of the glow can come from up to four different sources: the main texture, glow texture, glow color, or vertex color. In Unity 4.1 and beyond you can specify what source to use on each material. Prior to Unity 4.1 you are limited to specifying it for the scene. The amount of glow is based off of the glow texture. This glow texture is different between the standard glow effect and the alpha glow effect. In both cases, main texture is your standard main texture for the shader.

## Alpha Glow Effect




The alpha glow effect is very efficient because it does not need to a separate camera to apply the glow. The alpha glow effect uses the object's alpha channel to determine the amount to glow. A glow mask texture can be provided to prevent areas from glowing. A glow mask value of 1 (white) means the fragment should glow and a value of 0 (black) means it should not glow. A limitation of the alpha glow effect is that you cannot have an object in the transparent queue and not glow. If your design requires this, use the standard glow method. It will run just fine on most modern devices. *If the camera's background color is visible in the game and a glowing object is in the foreground then the alpha channel must be set to 1 of the camera's background color to get the full glow effect*.

## Built-In Shaders

BuiltInShaders.unitypackage/BuiltInShaders4.3+.unitypackage contains a modified version of the Unity built in shaders to add glow (with standard glow) or remove glow (with alpha glow). The 4.3+ versions includes the built in shaders starting with Unity 4.3.

## Creating your own Shaders

**Standard Glow Effect**

Add the following to your shader:

- The property: *_GlowTex ("Glow Texture", 2D) = "white" {}* - texture used to describe the glow amount/color. (if useGlowTexture is enabled)
- The property: *_GlowColor("Glow Color", Color) = (1, 1, 1, 1)* - color of the glow (if useGlowColor is enabled)
- The property: *_GlowColorMult ("Glow Color Multiplier", Color) = (1, 1, 1, 1)* - multiplies the glow texture by this color (if multiplyColor is enabled)
- The tag: *Tags { "RenderType" = "Glow" }* or *Tags { "RenderType" = "GlowTransparent" }* - specifies that this shader should glow.

See the Glow.shader and GlowAlphaBlend.shader in the Shaders folder for examples.

**Alpha Glow Effect**

Add the following to your shader:

- The property: *_GlowMask ("Glow Mask Texture", 2D) = "clear" {}* - texture mask used to determine the amount of glow. 0 means no glow, 1 means full glow.
- Return one channel from the _GlowMask to specify the amount of glow. For example, *return half4(tex2D(_MainTex, i.uv).rgb, tex2D(_GlowMask, i.uv).r);*

The following is an example of a surface shader that applies the alpha glow effect:

```
void surf (Input IN, inout SurfaceOutput o) {
        half4 c = tex2D (_MainTex, IN.uv_MainTex);
        half4 g = tex2D (_GlowMask, IN.uv_MainTex);
        o.Albedo = c.rgb;
        o.Alpha = g.a;
}
```

Examples of these shaders can be found in the Shaders folder as AlphaGlow.shader, AlphaGlowAlphaBlend.shader and SurfaceAlphaGlow.shader.

Shaders/UnlitNoGlow.shader provides an example of no glow occuring when using the alpha channel method. It returns 0 for the alpha channel.

If you have any questions please post on the forum or contact support@opsive.com.