



SELECTIVE GLOW  
HEATHEN ENGINEERING

HEATHEN ENGINEERING LMT | Co. Kildare, Republic of Ireland

## Contents

### Description

### Getting Started

#### Settings

Use Fast Blur

Sample Division

Iterations

Blur Spread

Intensity

#### Adding Glow Materials

### Material Shaders

#### Groupings

Cutout

Lightmapped

Mobile

Normal

Particles

Reflective

Self-Illumin

Transparent

### For Shader Developers

Custom Composite Shader

Custom Render Glow Shader

Custom Material Shader

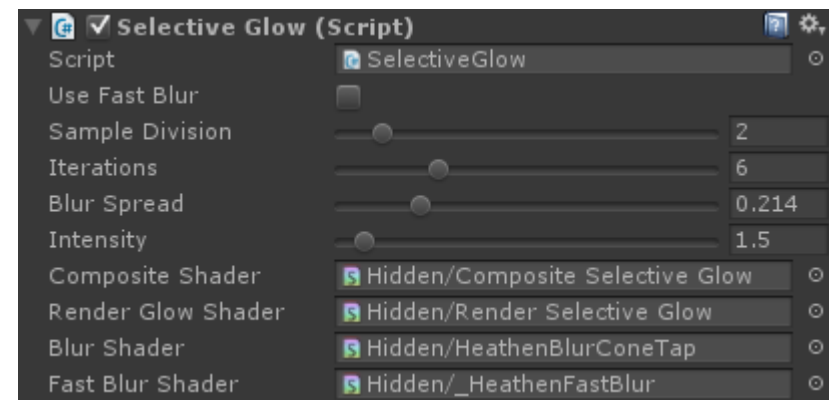
Troubleshooting

## Description

Heathen Engineering's Selective Glow is a screen effect and collection of material shaders which allow the developer to make any object glow regardless of rendering path or lighting model. The tool can be configured per material and at the camera level and can be combined with various other screen effects to intensify the dynamic look and feel of your scenes.

## Getting Started

First things first, add the 'Selective Glow' component to the camera you want to render the glow effects on.



You will notice that the Composite Shader, Render Glow Shader, Blur Shader and Fast Blur Shader variables are all populated for you. You can change these out to any shader you like however it will need to be a

compatible shader for that role. See the **For Shader Developers** section for more information on that.

## Settings

### Use Fast Blur

This tells the screen effect it should use the shader in the Fast Blur Shader slot when blurring the scene after rendering the glow. This has a big impact on how the 'glow' feels. Fast Blur Shader by default uses a Gaussian approximation and while very fast in comparison to the standard method and a rather easier on the GPU it tends to produce less clean glow effects especially over large areas. This is a good setting to use if you have a small screen or are sensitive to GPU performance such as mobile games but is also a solid choice when you have small areas glowing such as windows on a ship or decorations on an avatar and don't want to spend much in resourcing yet still get a strong blur.

For note with this off Blur Shader is used and by default uses a 4 cone tap, in short it runs 4 passes per iteration being the positive and negative of the X and Y axis generally producing a nicer blur effect but at a higher cost resource wise.

Note some platforms may miss behave when using the default Blur Shader while the Fast Blur Shader being simpler should produce stable results across a wide array of platforms.

### Sample Division

This tells the image effect how many times smaller the blur sample should be before blurring. In its most common use this is handy for decreasing impact of the blur operation regardless of the method chosen but it can also be used to achieve specific effects.

In simple terms it is scaling the image before blur down by the number of times indicated e.g. Sample Division of 2 means twice as small or divide it by 2.

### Iterations

This indicates how many times the effect should be blurred e.g. how many times to execute the selected blur option on the sample. This has an obvious effect on the efficiency but also appearance as each iteration is cumulative.

### Blur Spread

This indicates the offset of the texel size during the blur operation, the exact impact differs depending on the blur option used but in simple terms it causes the blur operation to sample nearer or further from the source pixel.

You can use this to make your blur spread further per single pass so you can get a wider glow effect with fewer iterations though it will have a lower quality than using a greater number of iterations.

### Intensity

This is a global strength value of the glow and is in addition to each materials specific inner and outer strengths. Use this to make the effect brighter, in short its multiplying the glow value.

## Adding Glow Materials

Once you have the screen effect applied you will to set something to glow. This is done by applying one of the Selective Glow material shaders. You can also make your existing materials 'Selective Glow' more on that in the **For Shader Developers** section.

You will find a wide selection of Selective Glow material shaders in your material shader list under **Heathen/Glow**; you can also find example materials already built under **Heathen Shaders/Materials**.

## Material Shaders

Heathen Engineering has provided you with a comprehensive list of the most common material shaders needed. Full source is available under the **Heathen Shaders/Shaders** folder. All material shaders use the same method to apply the glow effect e.g. the glow effect will render in the same way for all material shaders however Unity sorting may differ between each category of shader.

### Note

You might notice there is no Standard material shader at this time. This is because use of the standard material shader is generally accompanied by HDR cameras, bloom and other conditions which allow the stock Unity standard shader to present a glow effect. There are however specific cases where you may want to apply Selective Glow to a standard lighting model material shader. In this case please refer to the **For Shader Developers** section or contact Heathen Engineering Support by e-mail at [Support@HeathenEngineering.com](mailto:Support@HeathenEngineering.com) and we would be happy to assist you.

## Groupings

### Cutout

Also known as **AlphaTest** these shaders will render any pixel whose alpha channel is below a configured value as fully transparent otherwise it is rendered as full opaque e.g. cutting out the low alpha areas. The sorting of these materials on render is not as costly as 'Transparent' or semi-transparent as they are sometimes called but they also cannot render 'semi-transparent' materials.

These are particularly popular for mesh, screen and other common objects, examples can be found under the Materials folder pre-fixed with **SG Cutout**

### Lightmapped

This is a legacy material type not frequently used now but is fully maintained for backward compatibility with legacy Unity Shaders.

### Mobile

This is a collection of mobile optimized shaders, the glow effect behaves exactly the same between this and any other Selective Glow shader however the base material rendering is optimized for mobile platform in the same way as stock Unity Mobile shaders.

### Normal

These are simply opaque e.g. solid material shaders and make up the most commonly used shaders.

### Particles

This is a collection of Selective Glow enabled particle shaders based on Unity stock particle shaders.

### Reflective

This is a collection of Selective Glow enabled shaders based on Unity stock legacy reflective shaders.

### Self-Illumin

Based on the stock legacy Unity self-illuminated shaders

### Transparent

Collection of semi-transparent material shaders

## For Shader Developers

Heathen Engineering's Selective Glow provides you with the full source code of all the shaders and the screen effect. You can use these as a starting point for your own shaders or as production ready shaders. The following describes the steps necessary to create a custom Selective Glow shader as well as key requirements for the effect shaders.

### Custom Composite Shader

The composite shader referenced on the screen effect is responsible for combining the glow maps of all Selective Glow enabled materials on screen and the blur results its output will be a frame (image) used for the final result and is effectively  
[Original Rendering] + [Overlay un-Blurred GlowMaps] + [Overlay Blurred GlowMaps]

This is a very simple shader and won't likely need any adjustments, the screen effect expects to hand this shader 4 properties being

```
_MainTex ("", RECT) = "white" {}  
_BlurTex ("", RECT) = "white" {}  
_BlurRamp ("", 2D) = "gray" {}  
_Outter ("Intensity", Range(0.1,10))
```

1. MainTex; this is the original image, e.g. un-changed
2. BlurTex; this is the blurred image
3. BlurRamp; this is the unblurred glow map rendering
4. Outter; this is the screen effect 'intensity' value or strength

### Custom Render Glow Shader

This is a replacement shader, it renders material shaders according to the RenderType tag. That is it defines several unique SubShaders and uses the one whose tag matches that of the material shader being rendered. At minimal you will need to define a SubShader for the following tags

Example

```
Tags { "RenderType"="SelectiveGlow" "Queue"="Transparent" }
```

## Required RenderType SubShaders

- SelectiveGlow  
This tag value is used for all opaque materials which are Selective Glow enabled
- TransparentGlow  
This tag value is used for particle and other transparent materials which are Selective Glow enabled
- TriplanarGlow  
This tag value is used for the experimental Triplanar materials which are Selective Glow enabled
- Opaque  
This is used for non-selective glow non-transparent materials and should simply render black with 0 alpha

The properties required of this shader are as follows

```
_GlowMap ("Glow (A)", 2D) = "bump" {}  
_MainTex ("Base (RGB)", 2D) = "white" {}  
_Inner ("Inner Intensity", Range(0.1,10)) = 2.0  
_Outer ("Outer Intensity", Range(0.1,10)) = 2.0
```

- GlowMap  
This is the texture whose alpha defines glow strength and whose colour is used as the base glow colour of the pixel in question
- MainTex  
Not used but needed for fallback support
- Inner  
This defines the value by which the map is multiplied for the inner glow value
- Outer  
This defines the value by which the map is multiplied for the outer glow value

## Custom Material Shader

Any material shader can be made Selective Glow, all that is required is that a 'RenderType' tag be provided and set to SelectiveGlow or TransparentGlow depending on the desired behaviour of the glow with occlusion. When in doubt start with 'SelectiveGlow'.

While not required you most likely want to provide properties for the GlowMap, and the inner and outer strengths.

Tip: you may also want to set the Emission value of the material to some measure of the desired glow effect to support Unity's GI and reflection probes.

**Commented example located in `Heathen Shaders/Normal/Normal-Diffuse.shader`**



## Troubleshooting

If you have any issues feel free to contact Heathen Support by e-mail at [Support@HeathenEngineering.com](mailto:Support@HeathenEngineering.com) or on the web (live link available on main site) at [www.HeathenEngineering.com](http://www.HeathenEngineering.com)