



# Blur Shaders 2 for URP

## Four layer-maskable blur effects

### WHAT IS "BLUR SHADERS 2 FOR URP"?

---

*Blur Shaders 2* is a subset of blur-related effects from *Snapshot Shaders 2*. Parts of the pack, including filepaths, may reference the larger pack.

*Snapshot Shaders 2* is a collection of powerful new post processing effects, with the ability to mask every effect on a per-layer basis. It's the evolution of the original *Snapshot Shaders Pro*, with overhauled filters from the original pack making their way to the sequel with extra functionality and more planned in the future.

The latest version of this asset was created with [ **Unity 6000.0.23f1, URP 17.0.3** ]. This pack only supports the **Universal Render Pipeline**.

### SUPPORT

---

Sometimes things break! If you've encountered an error and this README doesn't have the answers (or if you have feature requests), then pop me an email at [danielilett+snapshotshaders@gmail.com](mailto:danielilett+snapshotshaders@gmail.com) and I'll try to sort you out. Please:

- **Clearly** describe the problem you are having and what steps I can take to reproduce the error.
- Include the **Unity version you are using**.
- Also include the **package version** you are using (you can find this in the Package Manager).
- Include the **render path** (i.e. Forward, Forward+, Deferred, Deferred+) – this option is on your URP settings asset.
- **Attach images or short videos** where necessary to describe your problem.

Following these steps will help me fix your issue as quickly as possible!

## SETUP

---

**Folder Structure:** Upon installing the pack, all the assets will be contained in the “*Daniel Ilett/Snapshot Shaders 2*” folder. All demo scenes are included in the “*Demo/Scenes*” folder. The main **DemoScene** displays every effect, and additional scenes are provided for every effect which showcase a handful of use cases for the effects.

## SETTING UP EFFECTS

---

*Blur Shaders 2* relies on URP’s Renderer Features and Volume system to render each effect. The [Unity documentation](#) will outline the basics of URP if you’re not familiar with how to create custom renderers.

Please follow these steps to ensure your effects run properly and avoid errors:

- Find your **URP Renderer Asset** and add the effect(s) you wish to use in the **Renderer Features** section at the bottom of the Inspector.
  - This asset is located in the *Assets/Settings* folder if you created a new project using the URP template in the Unity Hub and haven’t changed or added your own URP Renderer asset.
  - The default asset is named something like “PC\_RPAsset” by default, although this name may change in future Unity versions.
  - *Snapshot Shaders 2* also provides an asset in the “*Daniel Ilett/Snapshot Shaders 2*” folder, and a pipeline asset which uses the renderer in the root “*Daniel Ilett*” folder.
  - This step is **crucial!** If your effects aren’t included in this list, then they will **not** render, even if you add them to a volume profile in the next step.
- Create a volume profile asset via *Create -> Volume Profile* and add the effects you want to use to this profile. Add a volume to your scene via *GameObject -> Volume* and attach the volume profile.
  - You can find a range of pre-built profiles under “*Snapshot Shaders 2/Demo/Profiles*”.
  - If you add an effect to a profile which isn’t included in the renderer’s *Renderer Features* list, then an error message will be displayed in the Inspector with a button to automatically add it for you.
- Add a volume to your scene via *GameObject -> Volume -> {any option}*.
- Tweak the settings on your volume profile as desired.
  - **All effects start in an inactive state** when you first add a volume component to your profile, so don’t panic when you don’t see a difference the moment you add one. Just change a few of the parameters.
  - Some effects require a texture to function correctly. At least one example texture is supplied for each effect in the “*Snapshot Shaders 2/Textures*” folder.
  - When you select a volume in your scene and start tweaking the settings in its Inspector window, **you are modifying the source shared volume profile**. If you share that profile between multiple volumes, the changes propagate to all those volumes!

If you don't want that to happen, consider using separate profiles for each of those volumes.

## ADDITIONAL WARNINGS

---

This asset pack is designed from the ground up for **Unity 6 and above**. With that in mind, I have decided to support **Render Graph** *without the compatibility mode for non-RG paths*. This compatibility mode will be deprecated and removed in the future.

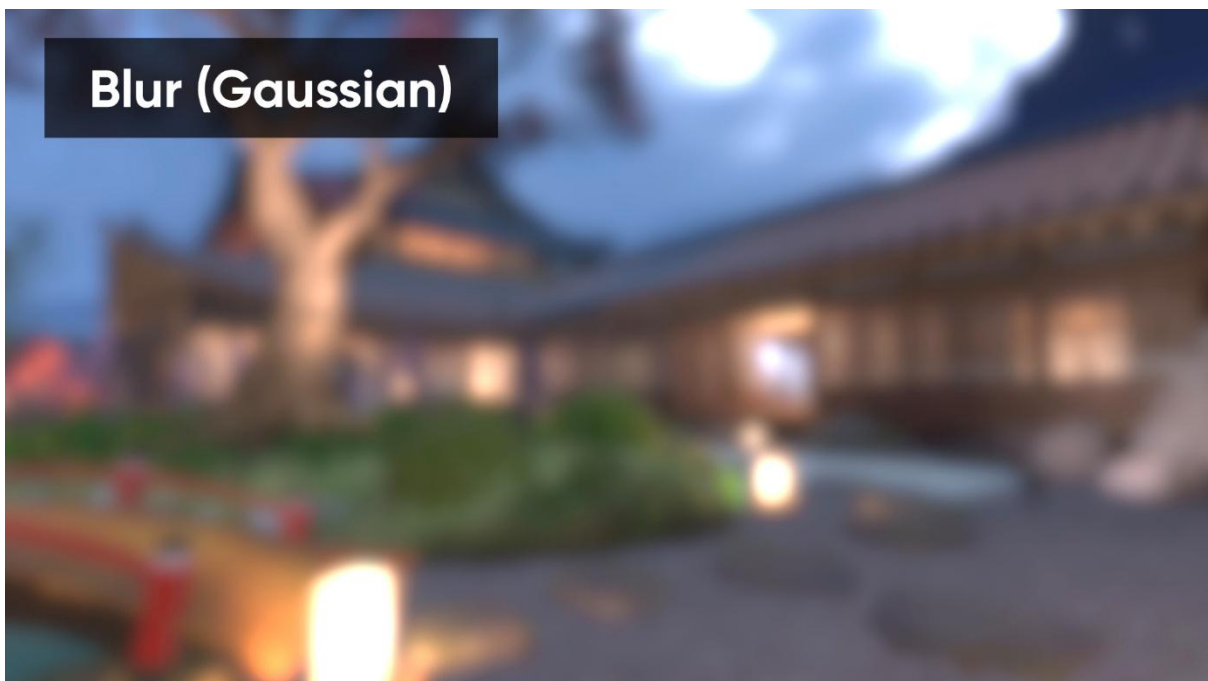
These shaders are designed for **linear color space**, so you may encounter issues in gamma space. To swap between color spaces, go to *Project Settings->Player->Other Settings* and find the **Color Space** dropdown option.

## BLUR

---

A multi-featured blur effect with several modes:

- **Gaussian** - Blur each pixel according to a Gaussian distribution, where pixels further away from the center pixel have a lower contribution to the center pixel output.
- **Box** - Blur each pixel by taking an unweighted average of the nearby pixel colors.
- **Radial** - Blur each pixel by taking a Gaussian-weighted average of a line of pixels radiating from the center of the image.
- **Light Streaks** - Blur pixels horizontally after taking a threshold of image brightness. Works especially well with HDR images, where some objects use high-intensity emissive colors or the scene contains extremely bright lights.



Note that some parameters only appear under some blur modes.

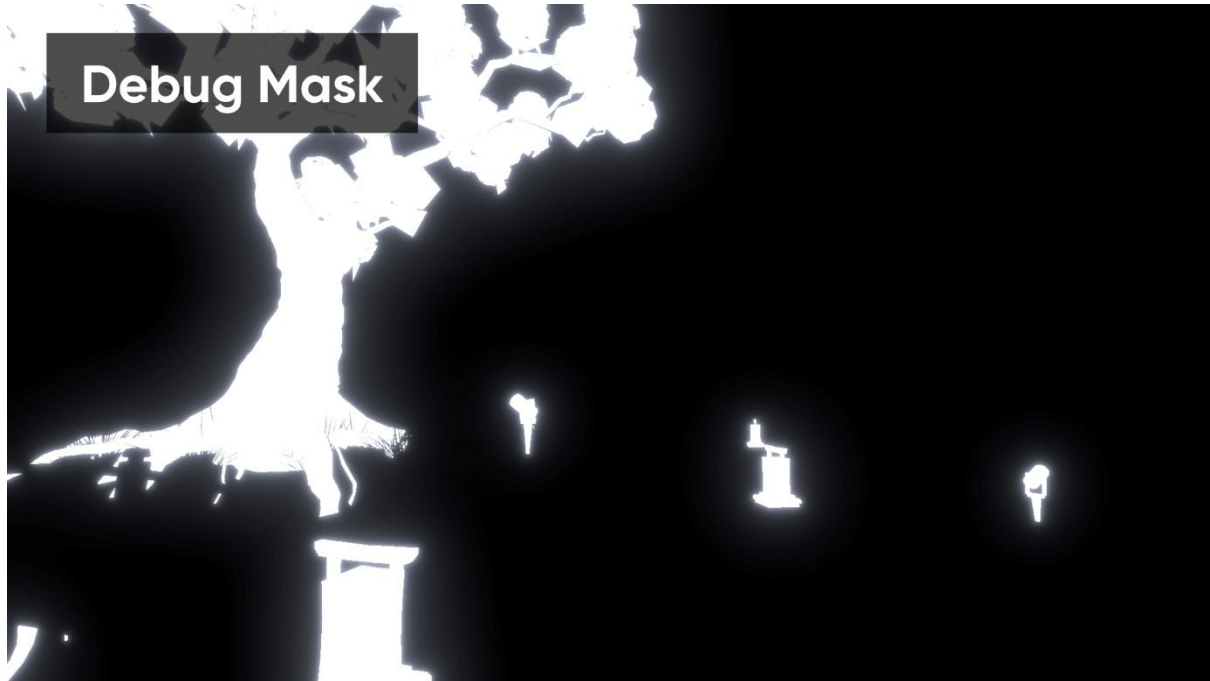
- **Blur Mode** - Which blurring algorithm to use. Choose between *Gaussian*, *Box*, *Radial*, and *Light Streaks*.

- **Strength** - How strongly the image is blurred. This impacts the size of the blurring kernel, so each effect is less efficient when strength is increased.
- **Blur Step Size** - For *Gaussian*, *Box*, and *Light Streaks*, this adds gaps in the blurring kernel, resulting in a “sparse” blur where far-away pixels may be considered without sacrificing performance. For *Radial*, this impacts the gap between each texture sample.
- **Luminance Threshold** - For *Light Streaks*, this is the brightness level which a pixel must exceed to produce streaks. For a non-HDR image, a completely white pixel has brightness 1.

## DEBUG MASK

---

A debug feature which lets you see the output of a local or global mask texture. This is useful for finding problems related to layer-based masking.



- **Enabled** - Should the debug view be rendered?

## LAYER MASKING

---

*Snapshot Shaders 2* comes with the ability to mask out every effect in the pack to specific layers based on a highly configurable filter.



You can mask objects based on a local or global mask. Local masks generate a texture for use by one effect only, whereas global masks are available for every effect in a single volume profile. Every effect comes with mask-specific settings, most of which only appear if you have chosen to use a mask.

### MASK SETTINGS

---

- **Mask Mode** - Which type of mask the effect should use. You can choose either *None*, *Global*, or *Local*.
- **Layer Mask** - Choose which object layers should be detected and drawn to the mask. You can choose any combination of layers, including *Nothing* or *Everything*. Note: you can modify the layer of any object in the top-right of its Inspector window, just above the Transform component.
- **Rendering Layer Mask** - Similar to the *Layer Mask*, but using Rendering Layers instead. Note: you can modify an object's Rendering Layer by accessing its Renderer component and finding this parameter near the bottom of its section in the Inspector.
- **Light Modes** - Choose which kind of objects to include in the mask texture, based on the sort of material being used to render it. You may add several Light Modes to the list. The most common settings you will use are *UniversalForward (Lit)* and *SRPDefaultUnlit (Unlit)*, but you can use the following:
  - *UniversalForward (Lit)* - This tag is used by shaders which support lighting.
  - *UniversalForwardOnly (some custom shaders)* - This tag is used by shaders which need to use Forward rendering, even if the Deferred rendering path is active.

- *SRPDefaultUnlit (Unlit)* - This tag is used by shader passes that don't specify any tag. It is commonly used to render objects which don't need to use lighting.
- *UniversalGBuffer* - This tag is used by shaders which are compatible with Deferred rendering.
- *Universal2D* - This tag is used by shaders which use the 2D Renderer.
- *ShadowCaster* - This tag is used by shaders which support casting shadows.
- *DepthOnly* - This tag is used by shaders which support rendering depth information into the depth texture.
- *DepthNormals* - This tag is used by shaders which support rendering normals information into the normal texture.
- *DepthNormalsOnly* - Similar to *UniversalForwardOnly*, this pass renders normals information into the normal texture using Forward rendering, even if the Deferred rendering path is active.
- *Meta* - This tag is used by the lightmap baking pass. Important: it is stripped from shaders when you build the game.
- **Render Queue** - Choose whether to include *Opaque*, *Transparent*, or *All* objects in the mask texture.
- **Draw Skybox To Mask** - Choose whether to include the skybox in the mask texture (unless an opaque object has been drawn in front of it).
- **Invert Mask** - Should the effect invert the contents of the mask texture (black becomes white, and vice versa)?

## LOCAL MASKS

---

Using a local mask is as easy as choosing *Local* mode in the **Mask Mode** and picking whichever settings you want. This mask will apply only to the effect which sets it.

## GLOBAL MASKS

---

Using a global mask requires you to add a volume component called **Global Mask** to your volume profile using the *Add Override* button on the profile. The *Render Pass Event* of the **Global Mask** must be before or the same as any effect that wishes to use it. Then, you need to select *Global* mode in the **Mask Mode** for each effect which needs to use it. This is more efficient than using individual local masks with the same settings, as you only need to generate one mask texture which can be shared between effects, but you may only use one Global Mask on each volume profile.

Global masks are most useful in the full pack, although even in *Blur Shaders 2*, you can use the global mask texture inside other effects that you create.

## SPECIAL THANKS

---

Many thanks to:

- [ambientCG](#) for some of the CC0 licensed textures used in the demo