WHAT IS "SNAPSHOT SHADERS 2 FOR URP"?

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. Plus, you'll see some all-new effects like **Dot Matrix** and **Thermal**.

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** 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.
- A**ttach 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

Snapshot 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.

EFFECTS LIST

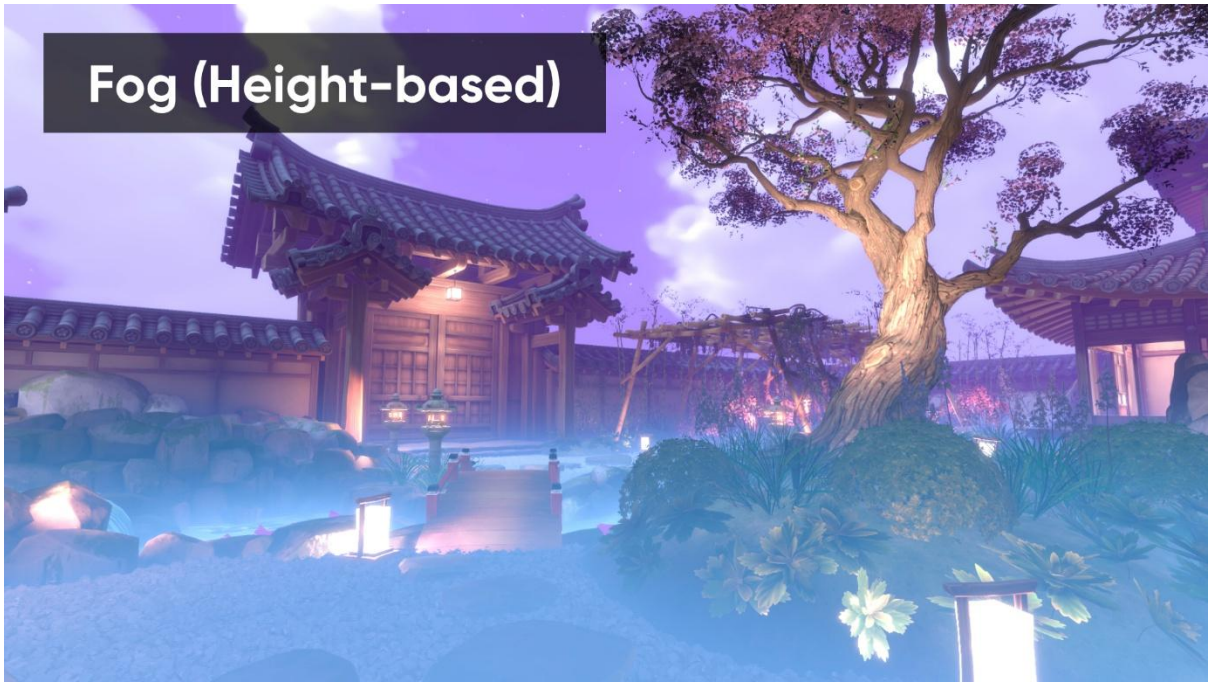
As of the current version, Snapshot Shaders 2 contains the following effects:

- **Fog** – add height-based and distance-based fog
- **Silhouette** – emphasize foreground and background elements
- **Painting** – make everything look like a Bob Ross oil painting
- **Retro** – relive the glory of old game consoles
- **Blur** – apply Gaussian, box, or radial blurs, or draw bright light streaks
- **Outline** – overlay outlines around scene elements
- **Underwater** – make the screen flow and overlay caustics
- **Thermal** – separate the screen into cold and warm colors
- **Glitch** – offset parts of the screen erratically in three different ways
- **Colorize** – tint the screen a different color
- **Afterimage** – blur a few of the previous frames with the current one for a lag effect
- **Color Separation** – move the red, green, and blur color channels apart
- **Colorblind** – simulate various color vision deficiencies
- **Synthwave** – overlay bright, colorful gridlines onto the world
- **Dot Matrix** – turn the screen into an array of lit bulbs
- **Distortion** – curve the screen like an old television
- **Scanline** – draw scrolling lines over the image
- **Dither** – overlay a dither pattern based on brightness
- **Pixelate** – reduce the resolution of the screen
- **Filmic** – add film bars or randomly-generated noise grain
- **Kaleidoscope** – mirror parts of the screen radially
- **Vortex** – twirl and twist the screen contents into a center point
- **Greyscale** – remove all color information and just show luminance
- **Sepia Tone** – harken back to the days of early photography

- **Sharpen** – increase the contrast between adjacent colors
- **Invert** – render red as cyan, yellow as blue, and green as purple
- **Debug Mask** – draw the contents of a mask texture to the screen
- **Global Mask** – set up a mask texture to share between effects

FOG

Overlays two kinds of fog onto the scene: distance-based fog which gets thicker as you get further from the camera, and height-based fog which operates on world height.



DISTANCE-BASED FOG

- **Distance Strength** - Visual strength of the distance-based fog.
- **Start Distance** - The distance from the camera at which distance-fog starts appearing.
- **Start Distance Color** - Color of the distance-fog at the *Start Distance*.
- **Full Strength Distance** - The distance from the camera at which distance-fog is at maximum strength.
- **Full Strength Distance Color** - Color of the distance-fog at the *Full Strength Distance*.
- **Distance Falloff** - Controls the color curve between the *Start* and *Full Strength Distances*.

HEIGHT-BASED FOG

- **Height Strength** - Visual strength of the height-based fog.
- **Start Height** - The height in world space at which height-fog starts appearing.
- **Start Height Color** - Color of the height-fog at the *Start Height*.
- **Full Strength Height** - The height in world space at which height-fog is at maximum strength. If this value is lower than start height, the fog gets stronger as height drops, and vice versa.
- **Full Strength Height Color** - Color of the height-fog at the *Full Strength Height*.

- **Use Fog At Skybox** - Should the fog be visible at the skybox?
- **Skybox Height** - A modified height value to use at the skybox. For distances between *Full Strength Distance* and the skybox distance, the shader will interpolate between *Full Strength Height* and *Skybox Height*. This ensures a smooth gradient transition between both fog values at the skybox, so I recommend making this value far lower than *Full Strength Height*.
- **Height Falloff** - Controls the color curve between the *Start* and *Full Strength Heights*.

SILHOUETTE

Draws each pixel of the screen differently based on its distance from the camera by blending between two colors or using a color ramp.

- **Enabled** - When ticked, the effect will render.
- **Use Texture Ramp** - When ticked, the effect will provide a Unity color gradient which you can edit, and will create a texture and gradient sub-asset on your volume profile. Removing the **Silhouette** effect from your volume profile will destroy those sub-assets.
- **Texture Ramp** - The texture being used for the ramp (this setting is not editable).
- **Near Color** - When the texture ramp is not used, this is the color of objects at the camera's near clip plane.
- **Far Color** - When the texture ramp is not used, this is the color of obejcts at the camera's far clip plane.
- **Power Ramp** - A power curve applied to the image before mapping colors. Values below 1 will bias the image towards the far clip plane color, and values above 1 will bias the image towards the near clip plane color.

PAINTING

Applies a painting-style filter onto the screen according to the chosen painting medium:

- Oil - Uses a Kuwahara filter to remove texture detail while preserving object edges.

- **Drawing Mode** - Choose which painting algorithm to use, with the following choices: *Oil*.
- **Kernel Size** - How many nearby pixels to consider when using the *Oil* painting filter.

RETRO

A range of retro graphics in the style of old videogame consoles:

- **Game Boy** - Quantizes the image based on luminance and applies four configurable colors to those luminance bands.
- **SNES (Posterize)** - Quantizes individual color channels with individual bit depths (the SNES used different values in different circumstances, up to 8).

- **Enabled** - When ticked, the effect will render.
- **Drawing Mode** - Which algorithm to use to draw the screen. Choose between *Game Boy* and *SNES (Posterize)*.
- **Power Ramp** - Applies a power scaler to each color channel prior to performing any color transformations. Values higher than one will darken the image, while values lower than 1 will brighten it.

GAME BOY

- **Darkest Color** - Color to apply to the darkest luminance band.
- **Dark Color** - Color to apply to the second darkest luminance band.
- **Light Color** - Color to apply to the second lightest luminance band.
- **Lightest Color** - Color to apply to the lightest luminance band.

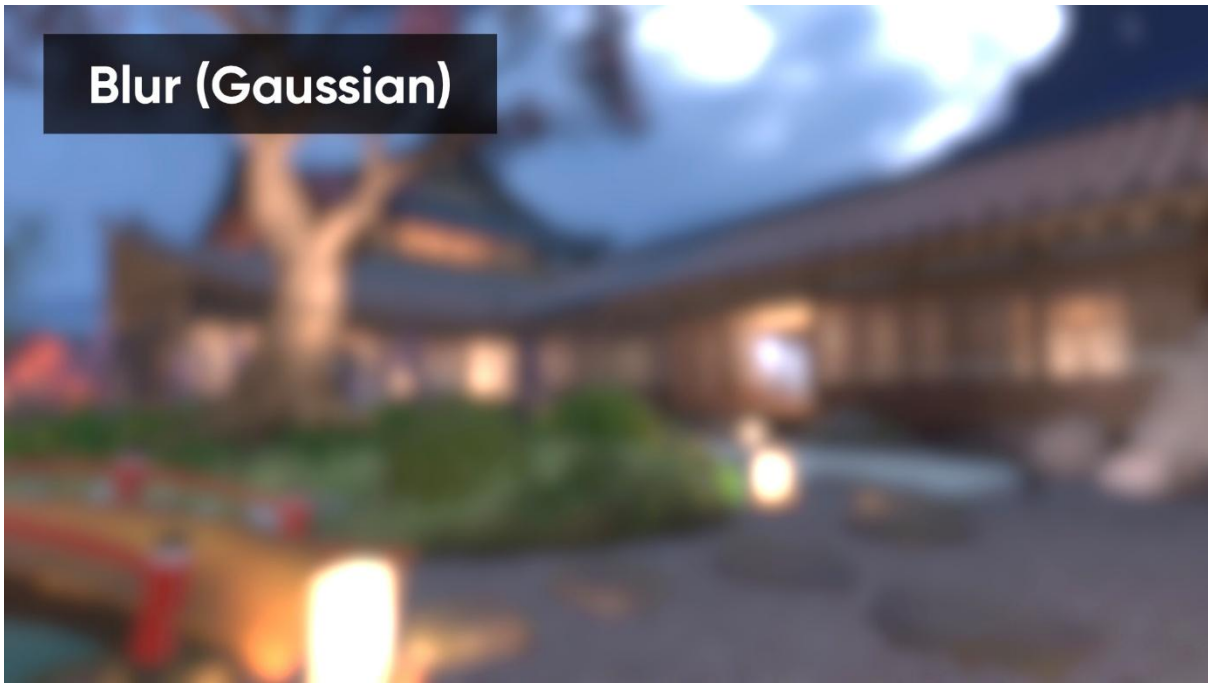
SNES (POSTERIZE)

- **Red Levels** - How many quantization levels to apply to the red color channel.
- **Green Levels** - How many quantization levels to apply to the green color channel.
- **Blue Levels** - How many quantization levels to apply to the blue color channel.

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.

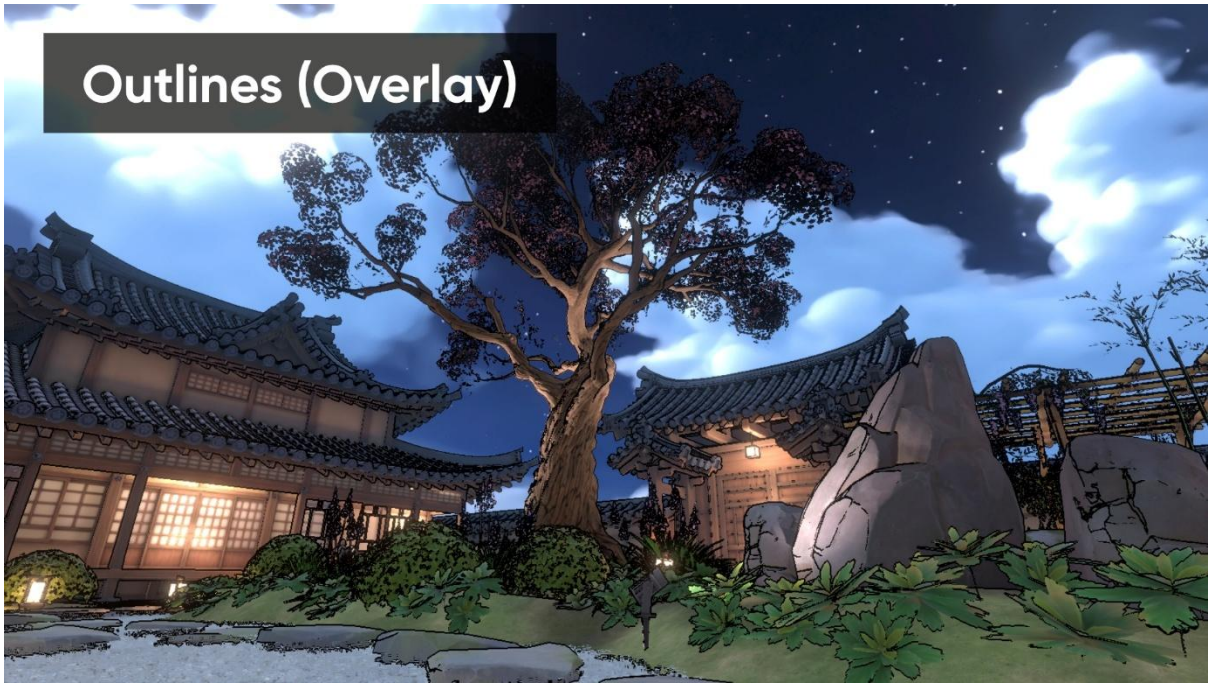


Note that some parameters only appear under some blur modes.

- **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.

OUTLINE

Overlays an outline onto the screen based on color, normal, and depth differences between nearby pixels.



- **Enabled** – Is the effect enabled?
- **Outline Algorithm** – Which outline should the shader to use to detect outlines?
    - *DepthNormalsColor* – uses differences in color, depth, and normal vectors across the original image.
- **Outline Color** – Color to use for the outlines. The alpha component acts as a global multiplier for outline strength.

- **Color Threshold** – Adjacent colors with differences over this threshold will be edge-detected.
- **Color Strength** – How strongly color-based edge detection factors into the overall edge strength calculation.
- **Depth Threshold** – Adjacent depth values with differences over this threshold will be edge-detected.
- **Depth Strength** – How strongly depth-based edge detection factors into the overall edge strength calculation.
- **Normal Threshold** – Adjacent normal vectors with direction differences over this threshold will be edge-detected.
- **Normal Strength** – How strongly normal-based edge detection factors into the overall edge strength calculation.
- **Skybox Depth Cutoff** – Pixels with depth exceeding this threshold will not be edge-detected.
- **Drawing Mode** – How should the shader draw the edges?
    - *Outline* modes use the Outline Color, while *Neon* modes use the original pixel colors and may boost their brightness and saturation.
    - *Overlay* modes layer the outline onto the original image, while *Only* modes display only the outlines alone.
- **Background Color** – Color to use for the background in *OutlinesOnly* or *NeonOnly* drawing modes.
- **Neon Saturation Floor** – When using neon colors, boost all pixels to use at least this saturation value.
- **Neon Lightness Floor** – When using neon colors, boost all pixels to use at least this lightness value.

UNDERWATER

Offsets the UVs of the screen according to a flow map texture, giving the scene an undulating appearance. This effect can also overlay caustics onto the scene based on a caustics texture, which is sampled multiple times and scrolled in different directions.

UNDERWATER SETTINGS

- **Wave Flow Map** - A texture representing the movement of screen UVs. The red and green channels of the image encode the strength and direction of movement of each part of the screen.
- **Wave Strength** - How strongly the *Wave Flow Map* should distort the screen UVs.
- **Wave Flow Tiling** - How many times the *Wave Flow Map* should be tiled over the screen.
- **Wave Flow Speed** - How quickly the *Wave Flow Map* scrolls over the screen.

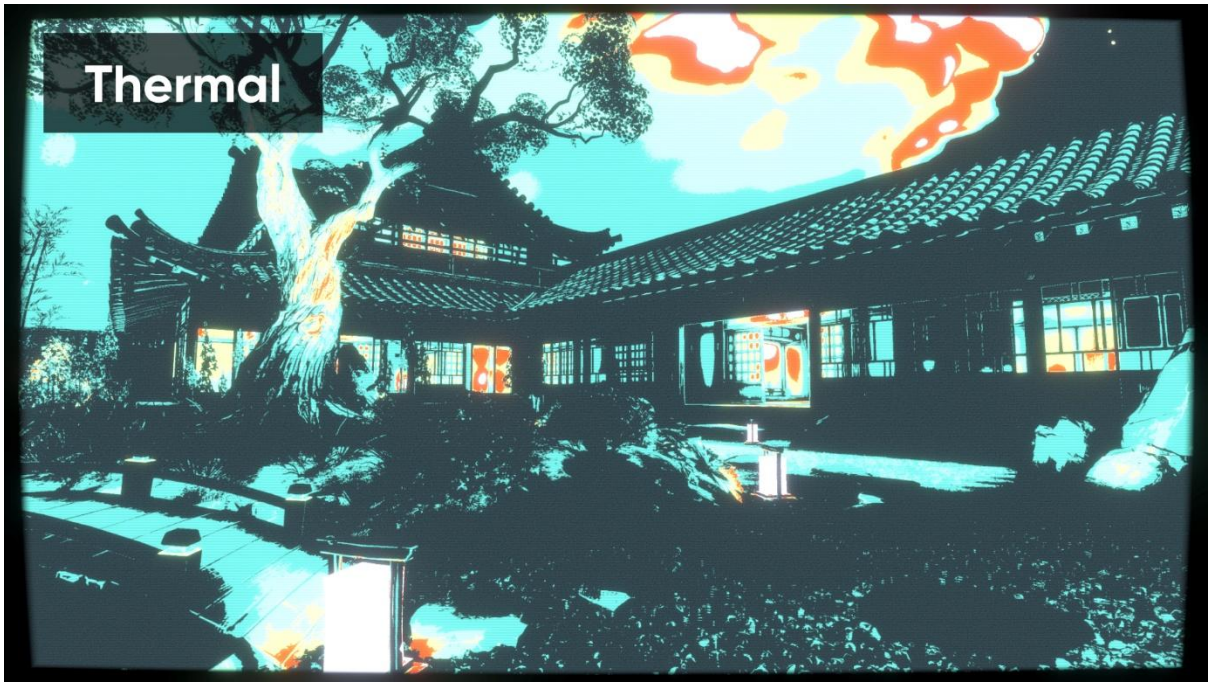
CAUSTICS SETTINGS

- **Caustics Mode** - Caustics are complex light reflections and refractions caused by curved surfaces, such as water ripples. You can choose:
    - *Off* - Don't display caustics.
    - *Triplanar* - Sample the caustics along the XY, YX, and XZ planes and blend the three results based on surface normals. This uses triple the number of texture samples compared to the *Light Aligned* option.
    - *Light Aligned* - Align the caustics sample to the direction of the main light.
- **Caustics Texture** - Texture to use for the caustics.
- **Caustics Tint** - Tint applied to the caustics. Use the alpha channel to control the overall strength.

- **Causting Tiling 1 & 2** - How frequently to tile the caustic texture samples. I recommend making these values different, but still relatively close to one another.
- **Caustics Scroll Velocity 1 & 2** - How quickly the caustics scroll over scene objects. I recommend using opposite values on each axis (e.g. if the x-component for one velocity is positive, make the other negative).
- **Caustics Color Separation** - How far the individual color channels of the caustics texture get separated. Warning: this will triple the number of texture samples you are already computing.
- **Caustics Start Fade** - The distance from the camera where the caustics begin fading out.
- **Caustics Fade Falloff** - The distance over which the caustics fade from full to zero strength.
- **Caustics Height Thresholds** – The height at which caustics start disappearing (at x-component), with a smooth falloff to zero strength (at y-component).

THERMAL

Calculates color bands in the image based on the luminance of each pixel and tints each band a different color, ideally starting from cold colors and transitioning to warm colors. This effect has a special interaction with the mask texture, which can be used to render the effect normally but artificially tag specific objects as ‘extra bright’.



THERMAL SETTINGS

- **Enabled** - Should the effect run?
- **Thermal Mask Mode** - Choose to only render masked objects, or render everything but masked objects are brighter than usual.
- **Mask Blur** - When a mask is enabled, how much should it be blurred? This can ensure a nice falloff between highlighted and background objects.
- **Mask Blur Step Size** - The blur algorithm can skip over some pixels, resulting in a sparse kernel for efficiency.
- **Add Brightness To Mask** - How much brightness should be added to masked objects?

COLORS AND THRESHOLDS

- **Color 0 - 5** - Color to use for each color band of the image. There are six available color bands.
- **Threshold 0 - 4** - Luminance thresholds to use between each pair of color bands. There are five thresholds between the six color bands.

GLITCH

Distorts the UVs of the image in several ways, such as small offsets along scanlines, large offsets produced by a scrolling shutter effect, and blocky chunks of the image sampling from the wrong place.

OFFSET GLITCHES

- **Offset Strength** - How strongly to offset parts of the image based on the **Offset Texture**, as a proportion of the total screen width.
- **Offset Texture** - A texture encoding how far to offset each line of the screen. This should be a vertical texture, where each pixel represents the distance each line of the image gets shifted. Black pixels move the image fully left, white pixels fully right, and grey pixels keep the image where it is.
- **Use Point Filter** - If ticked, the **Offset Texture** will be sampled with nearest-neighbor sampling. Otherwise, it will use bilinear filtering.
- **Offset Speed** - How quickly the shader scrolls down the **Offset Texture**.
- **Offset Tiling** - How many times the **Offset Texture is tiled across the screen vertically.

SLICE BAND GLITCHES

The slice band is a thick scrolling band of configurable width and speed which travels over the screen periodically and offsets the entire band of pixels. It randomly comes on and off.

- **Use Slice Band Glitches** - Choose whether the slice band is visible.
- **Slice Band Strength** - How strongly to offset pixels, as a proportion of screen width. Values below 0 will move slices to the left, and values above 0 move them to the right.
- **Slice Band Speed** - How quickly the band travels across the screen.
- **Slice Band Jitter** - How much the band wobbles up and down randomly as it moves.

- **Slice Band Width** - How thick the band is, as a proportion of screen height.
- **Slice Band Min Duration** - The minimum amount of time that a given band is visible for.
- **Slice Band Max Duration** - The maximum amount of time that a given band is visible for.
- **Slice Band Frequency** - The number of times per second that a slice band begins (this happens at regular intervals).

BLOCK ARTIFACT GLITCHES

Block artifacts are areas of the screen, divided into blocks based on UVs, which randomly sample from an incorrect part of the texture.

- **Use Block Artifacts** - Choose whether block artifacts are used.
- **Block Artifact Tiling** - How many blocks the screen is divided into along each axis.
- **Block Artifact Chance** - Roughly what proportion of the blocks will be randomly offset at a given time.
- **Block Artifact Strength Min** - How strongly the block is offset along the x-axis. Values below 1 move the blocks to the right, and values above 1 move them to the left (this directionality may be counter-intuitive). Offsets are random for each block - this is the minimum possible offset.
- **Block Artifact Strength Max** - This is the maximum possible offset.
- **Block Artifact Speed** - How quickly blocks cycle through random values. Quicker speeds mean each block is offset for less time.

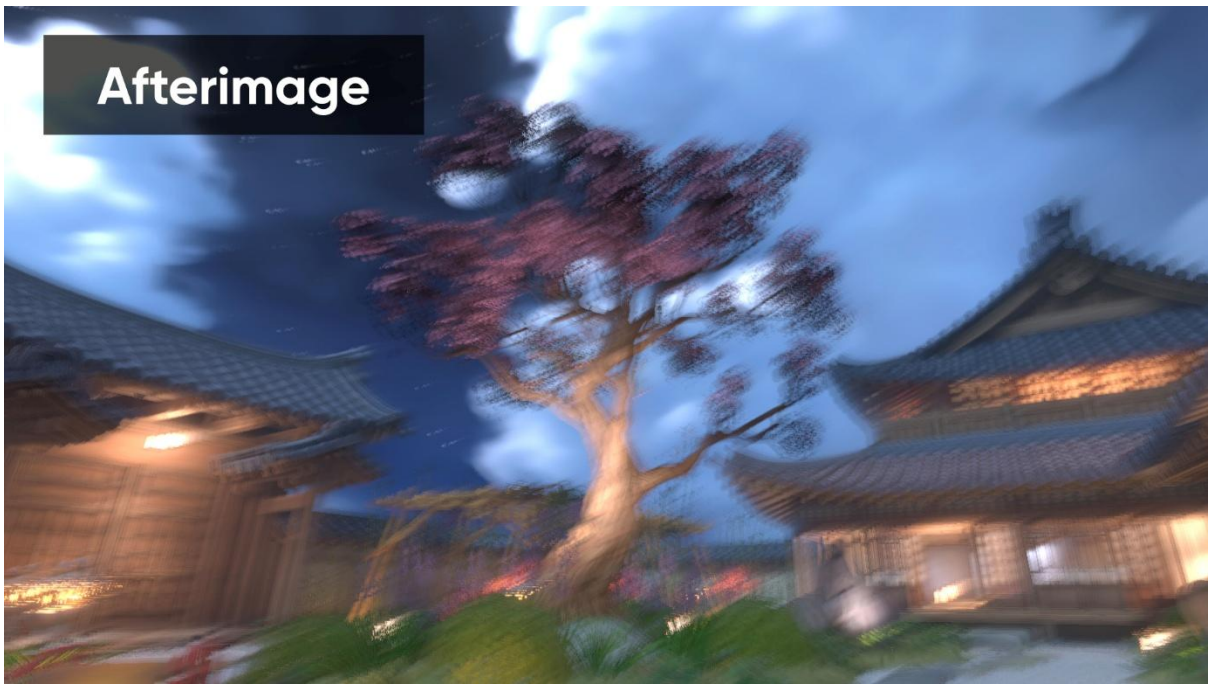
COLORIZE

Applies a color filter onto the screen contents.

- **Tint Color** - Which color to tint the screen. The alpha channel of this color acts like a strength setting.

AFTERIMAGE

Mixes the current frame with the contents of the previous frame with a configurable mixing proportion. Higher proportions make the screen feel ‘laggier’ as a higher number of previous frames stick around on the screen.



- **Afterimage Mode** – How should the afterimage be drawn?
  - *Off* = no afterimage.
  - *Masked* = only masked objects.
  - *Everywhere* = the entire screen uses an afterimage, even if a mask is set.
- **Persistence** – What proportion of the next drawn frame should be made up of the previous frame? Larger values cause a 'laggier' screen.

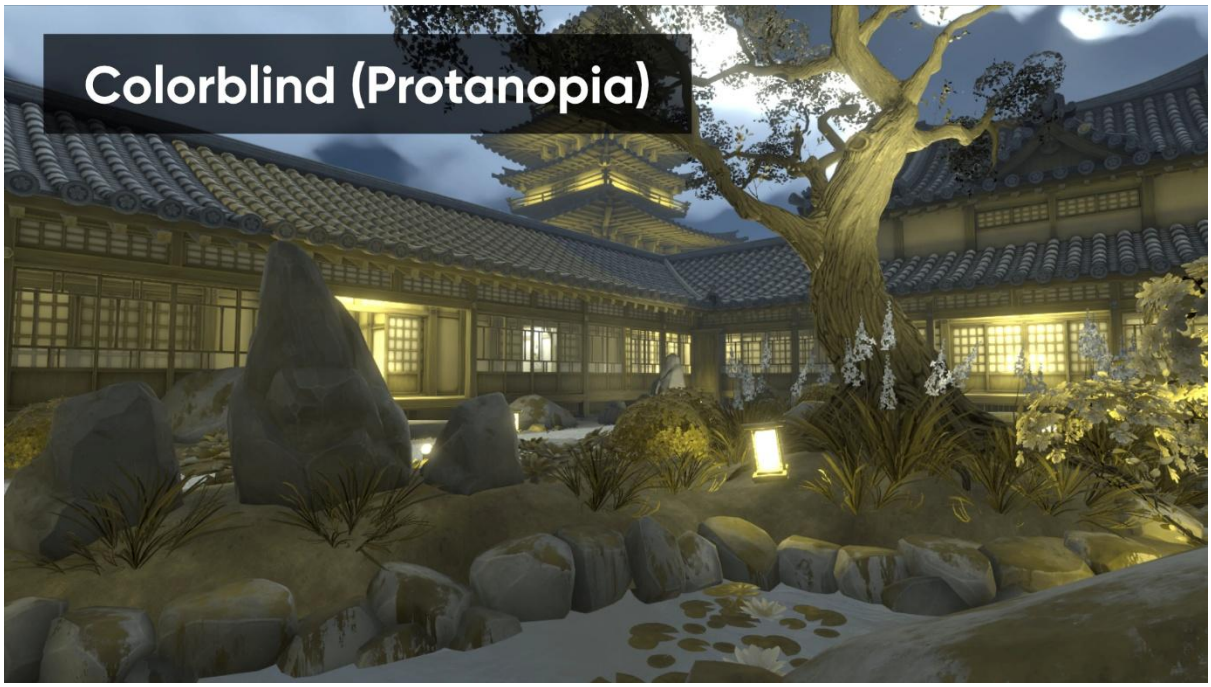
COLOR SEPARATION

The individual red, green, and blue color channels can be moved away from each other uniformly, sampled, and overlaid back onto each other.

- **Separation Offset** – Direction in which the blue channel moves in UV space. The red channel moves in the opposite direction.

COLORBLIND

Simulates different types of color vision deficiency as a quick test to see if your game is easy to visually parse for colorblind people. Note that this is not a medical diagnostic tool – please consult specialist tools for a better idea of how readable your game is!

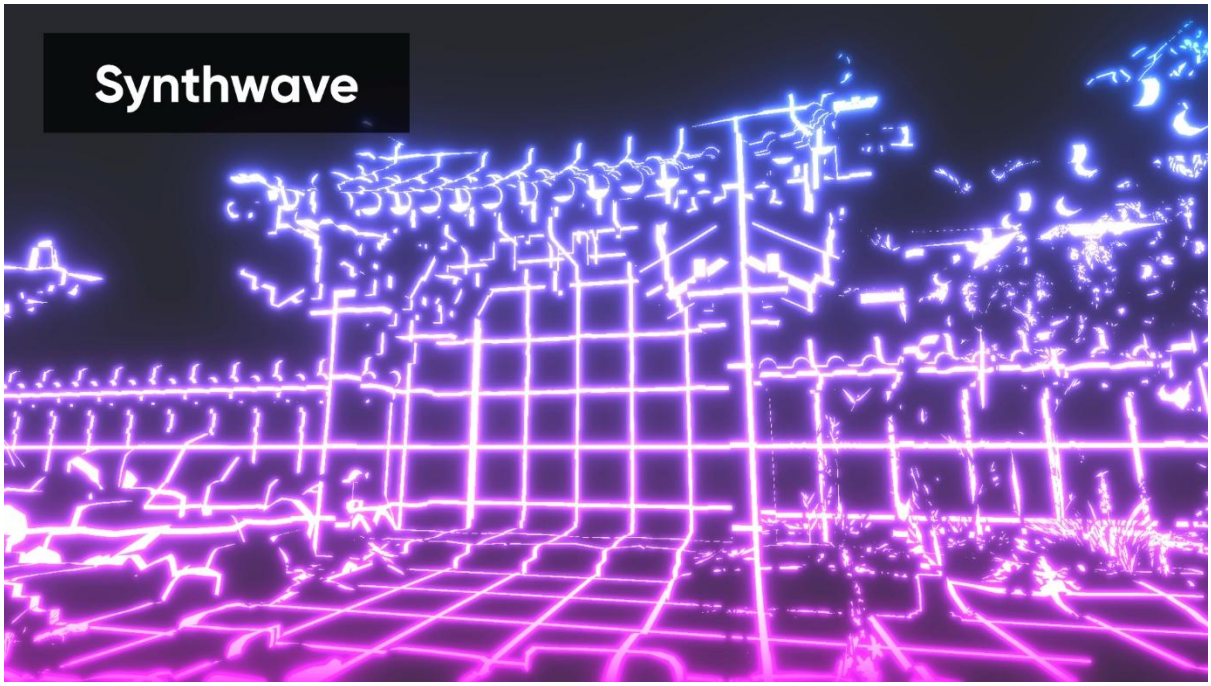


- **Colorblind Mode** – Which type of colour-blindness to simulate. The following are supported:
  - Protanopia
  - Protanomaly
  - Deuteranopia
  - Deuteranomaly
  - Tritanopia
  - Tritanomaly
  - Achromatopsia
  - Achromatomaly
- **Strength** – How strongly the filter is applied to the screen.

The values used in this effect are based on matrix values found on **this webpage**.

SYNTHWAVE

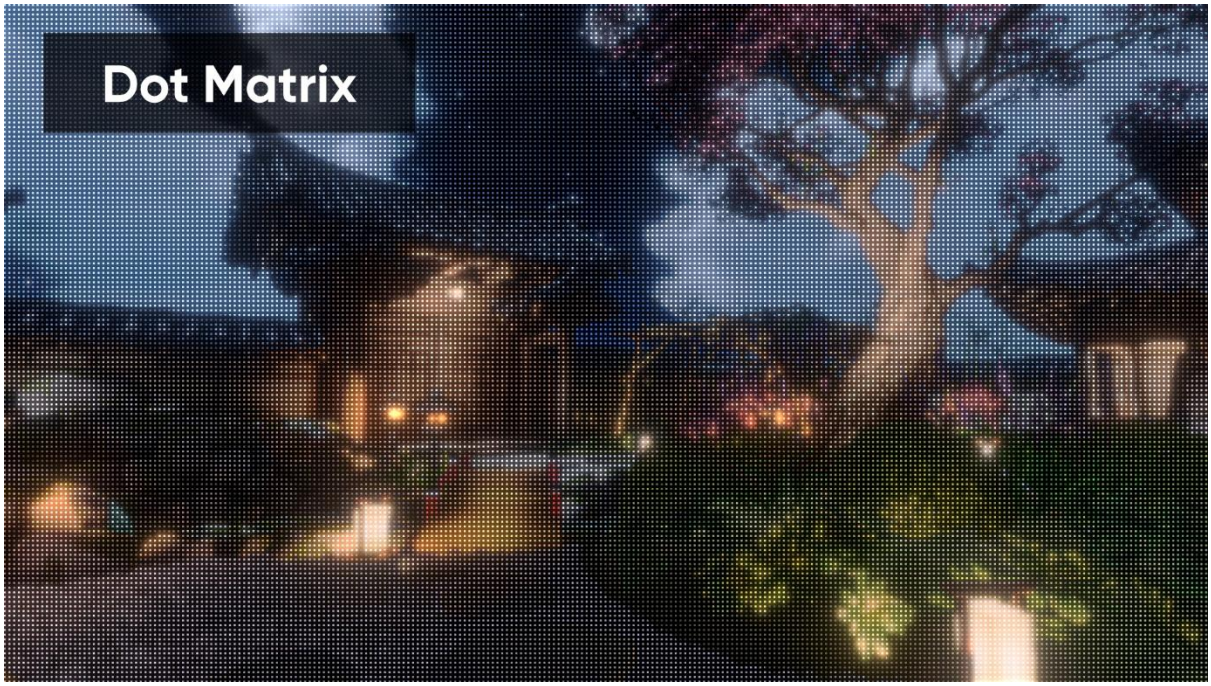
Overlays a synthwave gridline pattern onto the scene, with the option to replace the original scene color with a block background color.



- **Use Scene Color** - Use the scene color instead of *Background Color*?
- **Background Color** - Color of the background if *Use Scene Color* is turned off.
- **Line Color 1** - Color of the synthwave lines at the bottom of the screen. HDR colors will glow if a Bloom effect is present.
- **Line Color 2** - Color of the synthwave lines at the top of the screen. HDR colors will glow if a Bloom effect is present.
- **Line Color Mix** - Controls the mix between the two line colors. Lower values favour the top color (2). Higher values favor the bottom color (1). A value of 1 is a neutral mix (although it may not appear to be perceptually).
- **Line Width** - Thickness of the lines in world space units.
- **Line Softness** - Falloff between synthwave lines and *Background Color* in world space units.
- **Gap Width** - Space between lines along each axis in world space units.
- **Line Offset** - Offset from (0, 0, 0) along each axis in world space units.
- **Start Fadeout Distance** - Distance from the camera where the synthwave lines start to fade out.
- **End Fadeout Distance** - Distance from the camera where the synthwave lines become completely invisible.
- **Axis Mask** - Synthwave lines are shown only along these axes. Choose from X, Y, Z, XY, XZ, YZ, or XYZ.

DOT MATRIX

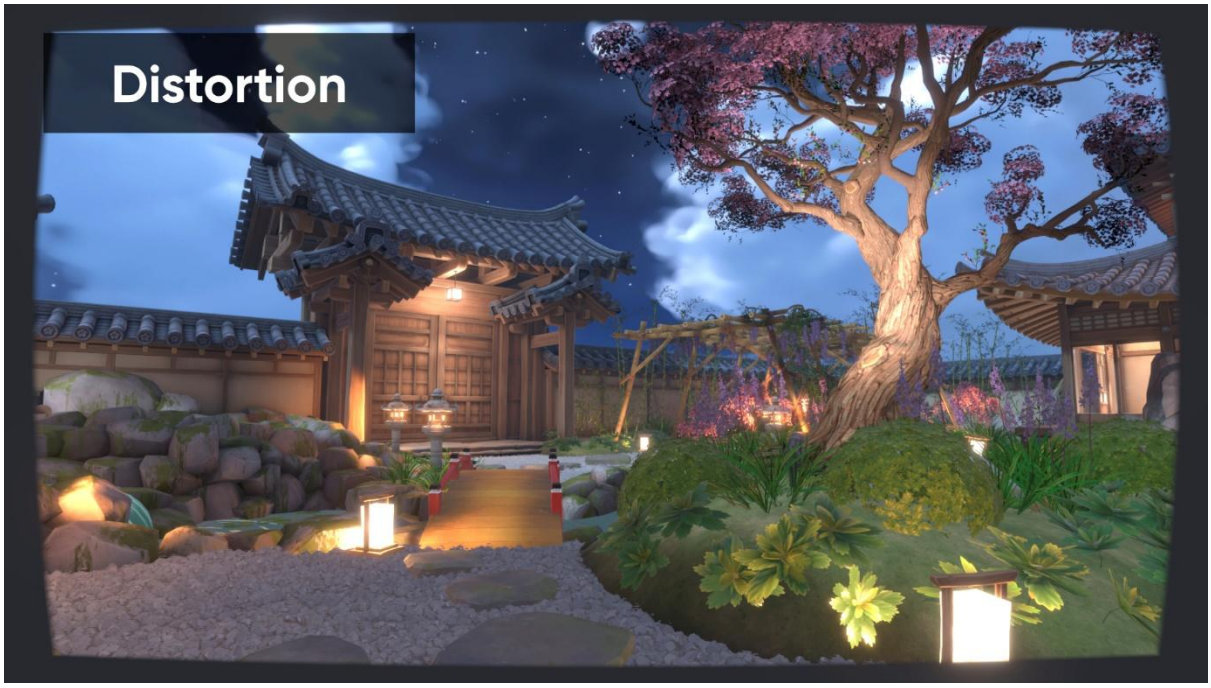
Turns the screen into an array of small square dots.



- **Gap Width** - How wide should the pixel gaps between dots be?
- **Dot Size** - How many pixels wide should each dot be?
- **Background Color** - The background color seen in the gaps between dots.

DISTORTION

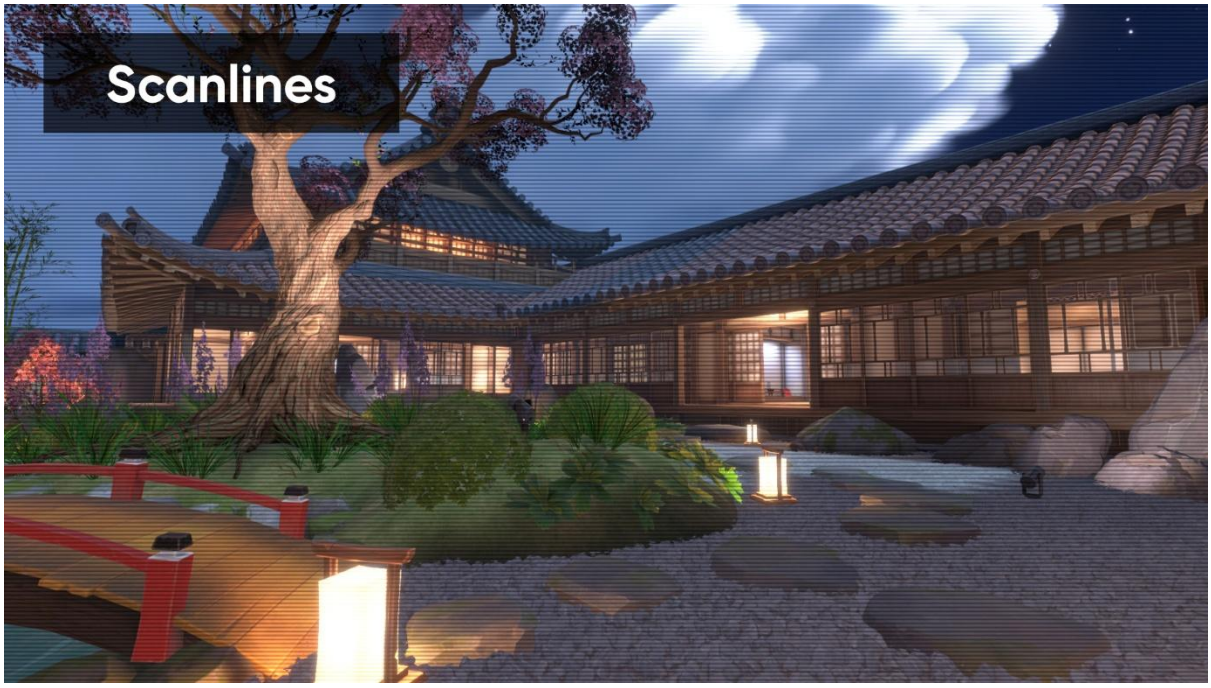
Imitates the effect seen when wrapping a 2D image across a barrel with a bulge in the centre. The corners of the screen are pulled in towards the centre more than the edges.



- **Strength** - How strongly to distort the screen. Values above 0 cause CRT screen-like distortion. Values below 0 bulge the screen outwards.
- **Smoothing** - Amount of smoothing applied to the edges of the distorted screen.
- **Background Color** - Color to use for the outer edge of the screen.

SCANLINE

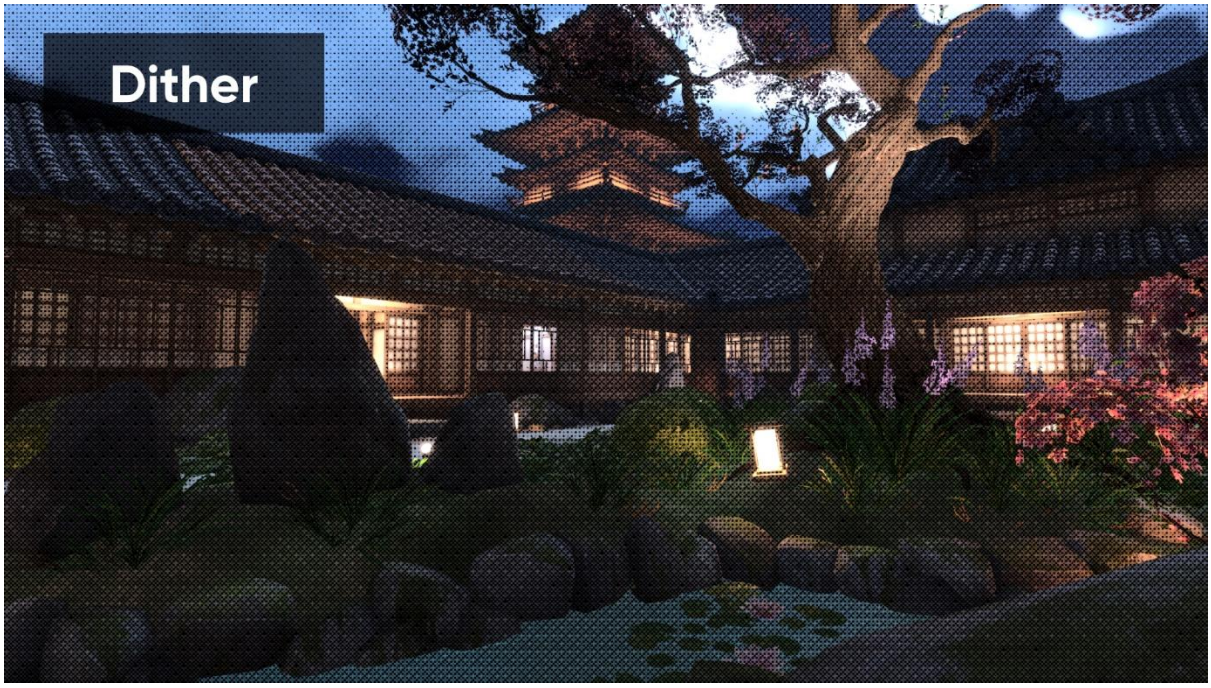
Overlays a small texture onto the screen and multiplies its colors by the existing screen colors. This texture can be scrolled downwards over time.



- **Scanline Texture** - Small texture containing the scanline pattern which will be tiled over the screen.
- **Scanline Strength** - How strongly the scanline texture should be multiplied with the original camera texture.
- **Scanline Size** - How many pixels the scanline texture takes up (i.e. setting this to 4 will tile the texture over 4x4 pixels).
- **Scanline Scroll Speed** - How quickly the scanlines scroll over the screen vertically.
- **Use Point Filtering** - Should the scanlines use point filtering for a crisper pattern?

DITHER

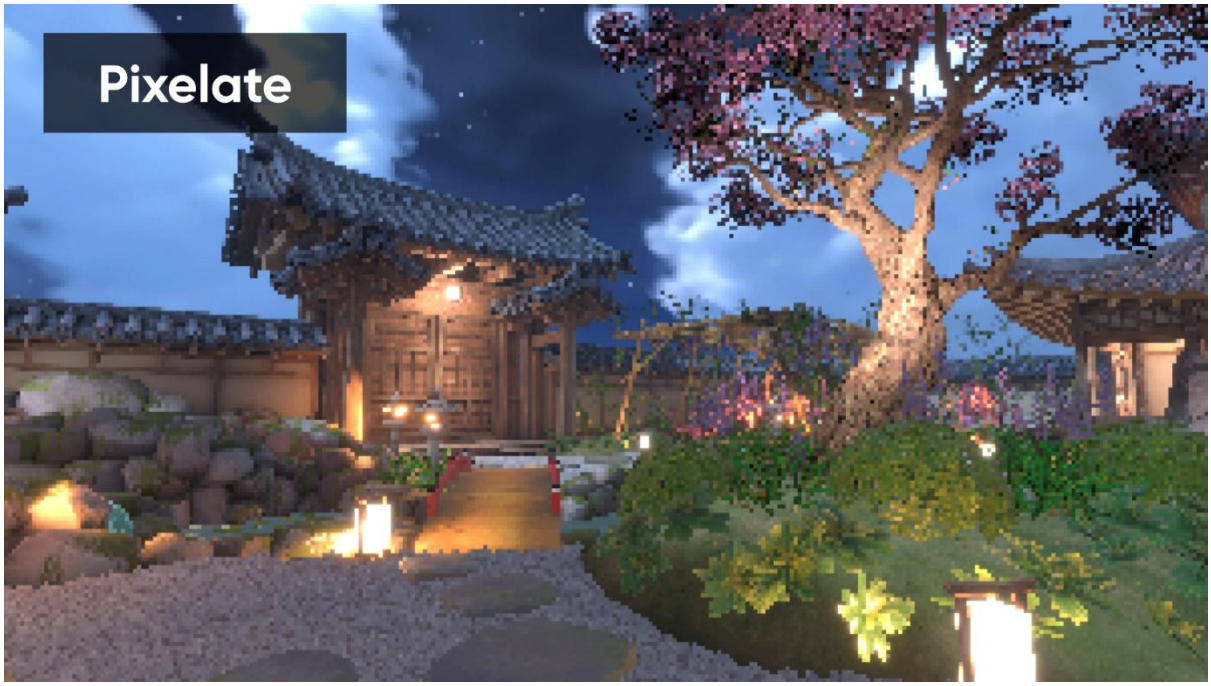
Overlays a dither pattern onto the screen based on the luminance (brightness) of each pixel. The dither pattern provides a pleasing gradient between dark and light areas of the screen.



- **Dither Mode** – Choose whether the threshold is applied to the luminance of the screen color, or to the individual RGB color channels of the screen color.
- **Noise Texture** - Texture pattern to use as dither thresholds (I recommend using a Bayer matrix or blue noise pattern, both of which are included in the asset pack).
- **Noise Size** - Size of the dither pattern tiled across the screen.
- **Luminance Threshold** - An additional offset applied to each pixel's luminance before thresholding.
- **Dark Color** - Color of the areas whose luminance fall under the threshold.
- **Use Scene Color** - When enabled, the Light Color is replaced by the original screen color.
- **Light Color** - Color of the areas whose luminance exceed the threshold.

PIXELATE

Reduces the screen resolution, with an option to use bilinear or nearest-neighbor filtering.



- **Downsample Size** - The factor by which to downsample the image. If this is set to 8, then a 'pixel' in the new image will take up 8 pixels on the screen.
- **Use Point Filtering** - When enabled, the shader uses crisp nearest-neighbor filtering. Otherwise, the shader will use bilinear filtering, resulting in a blurry image.
- **Expand Mask** - If you are using a mask, you can expand it slightly to ensure the entire mask contents fall within the blur. Enabling this setting will cause some minor elements outside of the mask boundaries to become pixelated, and is slightly slower, but usually looks nicer. If this setting is turned off, you will find parts of an image where some edges of the original object are still visible and not pixelated, due to unavoidable pixelation artefacts.

FILMIC

Add filmic effects like black bars based on aspect ratio and random fuzzy gradient noise for film grain.

- **Use Film Bars** - Toggle whether to use black bars at the top, bottom, and sides of the screen. This part of the effect ignores any masks applied to the effect.
- **Aspect Ratio** - If film bars are enabled, this controls the end aspect ratio of the screen. If this is wider than the original image, then bars are drawn on the top and bottom of the screen. If it is not as wide, bars are drawn on the sides of the screen.
- **Film Bar Color** - Which color to draw the bars if the aspect ratio does not match. This is usually black.
- **Noise Strength** - How strongly the film grain should be overlaid onto the screen.
- **Noise Speed** - How quickly the film grain scrolls to the next random value.
- **Noise Size** - How large each 'grain' of noise should appear on screen.
- **Noise Interpolation** - Which interpolation algorithm to use for smoothing the noise grain values. *Hermite* is faster, while *Quintic* looks very slightly nicer (but you may not notice with fast-scrolling grain).

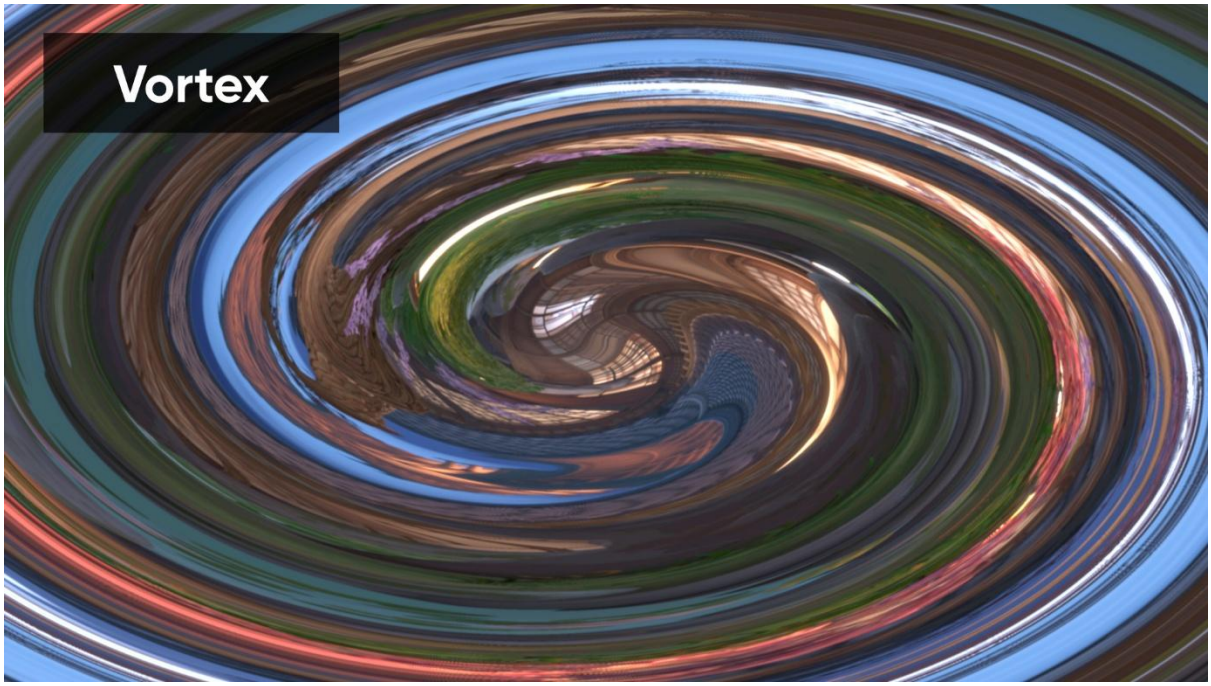
KALEIDOSCOPE

Reflects part of the scene radially along several mirror lines crossing through the centre of the image.

- **Segments** - The number of radial segments to use.

VORTEX

Swirls the image around a center point in a vortex pattern.



- **Strength** - How strongly to whirl pixels around the vortex. Higher values mean pixels make more rotations around the screen.
- **Center** - The center point of all vortex rotations, as a normalized screen coordinate.
- **Offset** - How far to offset the image prior to performing the vortex effect.
- **Rotation** - How much to rotate the entire screen uniformly post-vortex, as a proportion between no rotation and one full rotation.

GREYSCALE

Outputs the luminance of each pixel color.

- **Strength** - How strongly to blend between the original color and the greyscale version.

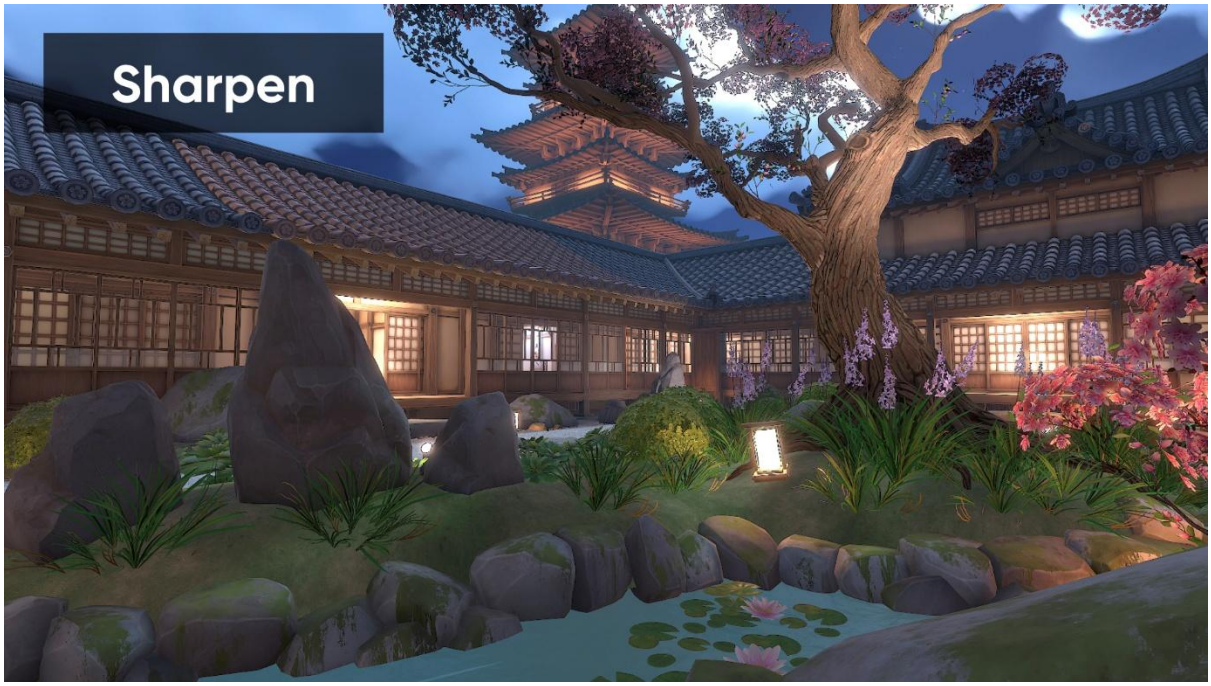
SEPIA TONE

Outputs each color as a yellow-tinted version, styled like an old photograph.

- **Strength** - How strongly to blend between the original color and the sepia-toned version.

SHARPEN

Sharpens the image by reversing the effect of small-kernel blur effects.



- **Strength** - How strongly to increase the sharpness of the image.

INVERT

This effect inverts the screen's colors in an HDR-friendly manner. This means that color data will be inverted, but the shader will try and conserve luminance values.

- **Strength** - How strongly to blend between a normal and inverted image. A value of 0.5 will result in a grey image, although some areas may turn out bright white if HDR is used and this filter happens before any tonemapping.

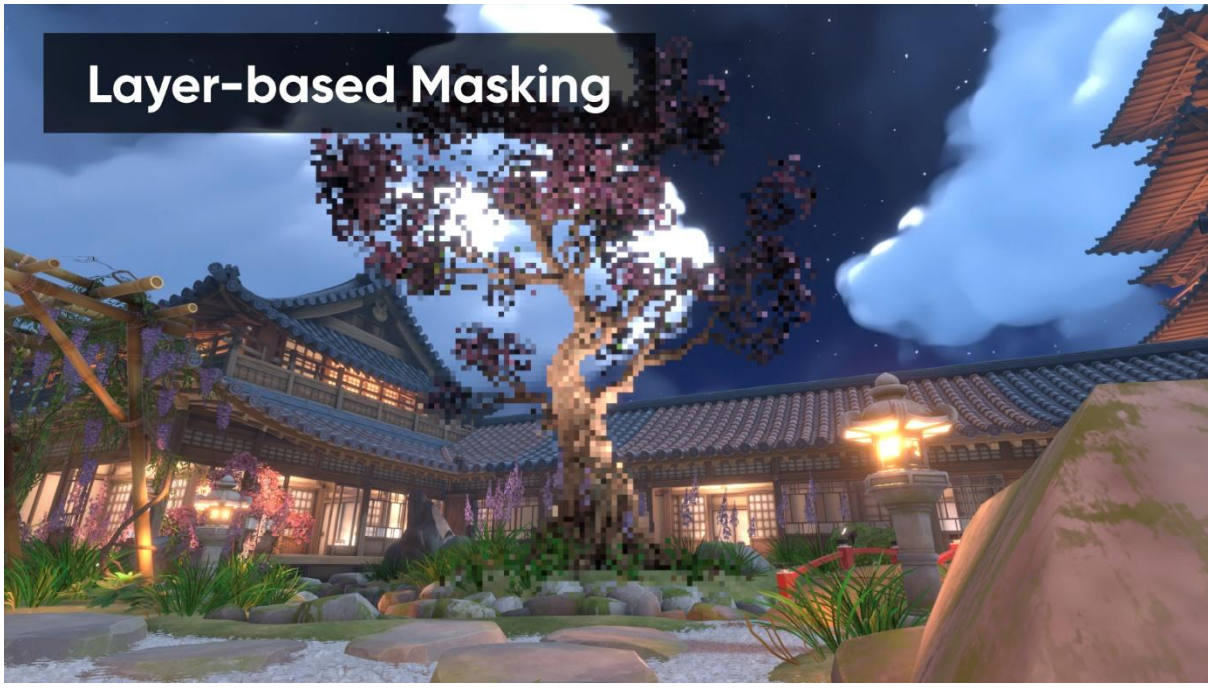
DEBUG MASK

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.

SPECIAL THANKS

Many thanks to:

- ambientCG for some of the CC0 licensed textures used in the demo