WHAT IS "RETRO SHADERS PRO FOR URP"?

Retro Shaders Pro for URP is a collection of shader effects which emulate the look and feel of retro games. This style is commonly referred to as the "PSX style" due to its similarity to the look and feel of the PS1. As of version 1.2, this pack also includes N64-style settings.

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+retroshaders@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) – 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 “Retro Shaders Pro” folder. All demo scenes are included under “Retro Shaders Pro/Demo”, containing examples of meshes and textures configured to work with the Retro shaders.

Most of the shaders are included inside “Retro Shaders Pro/Shaders”, including the terrain, decals, and skybox shaders. The post processing assets are included in the “Retro Shaders Pro/Scripts” and “Retro Shaders Pro/Resources” folders.

Auto-Installer: An auto-installer tool is included with this pack which will help you install extra Shader Graph versions of some shaders for you to modify. The window should open upon installing the pack, but you can open this installer at any time via *Tools -> Retro Shaders Pro -> Open Installer Window*.

Warning: When the “Retro Base Lit” graph is imported into your project, you may see compiler errors related to being unable to import the graph. Please go into *Preferences -> Shader Graph* (before Unity 6) or *Project Settings -> Shader Graph* (Unity 6 and after) to increase the shader variant limit – this might need to be at least 512. This is due to the usage of shader keywords for the graph’s lighting capabilities.

MESHES

The Retro Lit shader is intended to be used with regular meshes in your scene. To use it, create a regular Material and select *Retro Shaders Pro/Retro Lit* from the drop-down menu, then attach the material to your object.

This pack also features the Retro Base Lit and Retro Base Unlit shaders, which are built in Shader Graph and are intended to provide a more approachable basis for you to create your own effects which use similar functionality as the Retro Lit and Retro Unlit shaders.

You may also use the CRT (Mesh) shader to apply the CRT-style effect to any regular mesh in your scene.

TERRAIN

The Retro Terrain Lit shader may be used for your Unity terrains. To use this shader, create a Material and select the “Retro Shaders Pro/Terrain/Lit” shader from the menu. Then, select your terrain and assign this material in the Terrain Settings tab on the Terrain component (the option is inside the Basic Terrain foldout). Then, you can paint textures onto your terrain as usual.

SKYBOX

The *Retro Skybox* effect supports some of the retro features of the Retro Lit shader, except you apply it to the skybox field in *Lighting -> Environment -> Skybox Material*. For the sky color, you can choose between a color gradient and sampling a cubemap texture.

This shader also gives you the option to overlay a procedural cloud pattern across the sky which scrolls across the X-Z plane.

DECALS

The Retro Decals shader can be used in conjunction with the URP Decal Projector. This shader supports some of the features of the Retro Lit shader (namely, resolution and color depth limiting), but sadly, it doesn't seem possible to support affine texture mapping or vertex snapping currently.

POST PROCESSING

This shader pack uses **Universal Render Pipeline**'s **ScriptableRenderFeature** functionality for the custom CRT post processing 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 enable an effect in your scene:

- Find your **URP Renderer Asset** and add the effect(s) you wish to use in the **Renderer Features** section at the bottom.
    - This is most commonly found in the Assets/Settings folder if you created a new project using the URP template from the Unity Hub.
    - This asset will be named something like "UniversalRP-HighQuality" (Unity versions 2022.3 and prior) or "PC_RPAsset" (Unity 6) by default.
    - *Retro Shaders Pro* also includes a ready-made asset named "Retro_Renderer" in its root folder, which has the CRT effect pre-added.
- Create a volume profile asset via **Create -> Volume Profile** and add the CRT effect (and any other effects you want to use) to the profile.
- Add a volume to your scene via *GameObject -> Volume* and attach the volume profile.
- Tweak the settings on your volume profile as desired. The CRT effect may require textures to achieve the visuals you desire.

The latest version of this asset was created using Unity 2022.3.0f1 and URP 14.0.7.

UNITY 6 RENDER GRAPH COMPATIBILITY MODE

Post processing effects in Unity have moved to the new **Render Graph API**. This migration started in Unity 6.0, and the non-RG workflow was removed in Unity 6.3. *Retro Shaders Pro* supports both RG-based and non-RG-based projects.

Unity 6.0, 6.1, and 6.2 let you use compatibility mode for post processing effects. In these versions, you can disable Render Graph via *Project Settings -> Graphics -> Pipeline Specific Settings -> URP*. You will find a checkbox to disable Render Graph near the bottom of the window. However, this shouldn't be necessary unless your project uses external effects which rely on the non-RG workflow.

ADDITIONAL WARNINGS

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.

The shaders in *Retro Shaders Pro* support the Forward, Forward+, Deferred, and Deferred+ rendering paths.

ASSETS INCLUDED

The following assets are included in the asset pack:

RETRO LIT

A PSX-style shader that can be attached to any mesh to instantly give it a classic PS1 aesthetic. It supports as many light sources as you want, with the option to choose between several lighting models (unlit, diffuse only, specular lighting, texel-aligned lighting, and so on).

This shader gives you the option to use affine texture mapping for sampling textures (as this was a limitation of the original PS1 hardware), so objects may appear warped when viewing them at extreme angles, or when viewing large triangles.

Note that the shader file uses shader_feature extensively. **Unity will only include the shader variants which exist on materials at build time**, as opposed to multi_compile, which includes all possible variants. This will speed up build times drastically, but you may not be able to dynamically switch between some options at runtime.

SURFACE OPTIONS

- **Surface Type** – Toggle between *Opaque* and *Transparent* rendering.
- **Render Face** – Choose whether to render *Front*, *Back*, or *Both* faces.
- **Alpha Clip** – Toggle whether the shader should apply the *Alpha Clip Threshold*.
- **Alpha Clip Threshold** – Pixels with final alpha values below this threshold will be culled (if *Alpha Clip* is enabled).

COLOR & TEXTURE PROPERTIES

- **Base Color** – The albedo color of the object.
- **Base Texture** – An albedo texture to apply more color detail than *Base Color* alone.
- **Color Depth** – Each color channel is constrained to this many possible values. Low values may darken your image because a floor function is applied.
- **Color Depth Offset** – Applies a slight offset to the colors to avoid the darkening issue.
- **Resolution Limit** – Sets an upper bound on the resolution of *Base Texture* (will be rounded down to the next power of two).
- **Affine Texture Mapping** – Toggle between affine and perspective-correct mapping.
- **Filter Mode** – Choose how to filter the Base Texture while sampling:
    - *Bilinear*: Blend between the nearest four pixels, which appears smooth.
    - *Point*: Use nearest neighbor sampling, which appears blocky.
    - *N64*: Use the limited 3-point sampling method from the Nintendo 64.
- **Wrap Mode** – Choose how to wrap the Base Texture when sampling:
    - *Clamp*: When UVs exceed the 0-1 range, the color clamps to the nearest edge of the texture.
    - *Repeat*: When UVs exceed the 0-1 range, the texture is sampled using the fractional part of the UV coordinates.
- **Dither Mode** – Choose how the shader should use dithering to blend colors which fall between color bit values.

    - *Screen*: Use screen-space coordinates for dithering. This mode is driven in part by the pixel size parameter of the CRT post process effect.
    - *Texture*: Use texture coordinates for dithering.
    - *Off*: Don't use any dithering.
- **Use Vertex Colors** – Choose whether to multiply the base color using vertex colors.

VERTEX SNAPPING PROPERTIES

- **Snapping Mode** – Choose how to snap vertices to a limited number of points in space.
    - *Object*: Snap vertices relative to the model coordinates.
    - *World*: Snap vertices relative to the scene coordinates.
    - *View*: Snap vertices relative to the camera coordinates.
    - *Off*: Don't do any snapping.
- **Snaps Per Meter** – The vertices of the mesh will snap to this number of snap points per meter along each axis.

LIGHTING & SHADOW PROPERTIES

- **Normal Texture** – A texture which changes the direction of the normal vector while calculating lighting.
- **Normal Strength** – Changes the strength of the normal offsets produced by the normal texture.
- **Lighting Mode** – Choose what type of lighting model to use.
    - *Lit*: Use per-pixel lighting as standard.
    - *Texel Lit*: Snap lighting and shadows to the closest texel on the object's texture.
    - *Vertex Lit*: Use per-vertex lighting and interpolate light values for pixels.
    - *Unlit*: Don't use lighting calculations (everything is always fully lit).
- **Use Flat Shading** – Should the shader flatten the mesh triangles while shading?
- **Receive Shadows** – Should shadows from other objects affect this object?
- **Ambient Light Override** – Toggle to choose Unity ambient light or the Strength property:
- **Ambient Light Strength** – Sets a lower bound for how dark the shadowed areas of a mesh may appear.
- **Use Specular Lighting** – Choose whether to apply a specular highlight to the object.
- **Glossiness** – A power value to apply to the specular lighting. The higher this value is, the smaller the highlight appears on the surface of the object.
- **Use Reflection Cubemap** – Choose whether to use a cubemap texture to approximate indirect light reflections surrounding the object.
- **Reflection Cubemap** – A cubemap texture which encodes the reflected light around the object.
- **Cubemap Rotation** – How much to rotate the cubemap around the y-axis, in degrees.

NOTE: SCREEN-SPACE DITHERING SIZE

When screen-space mode is chosen, the pixel size parameter of the dithering is driven by the CRT Renderer Feature if you are using one. If no such CRT effect is active, the pixel size will be 1:1 with the screen resolution.

NOTE: VERTEX LIT, UNLIT, AND BILINEAR3 VARIANTS

Versions of this asset pack prior to Version 1.5 split much of the functionality of the Retro Lit shader into different shader files (e.g. vertex lighting was supported by the Retro Vertex Lit shader, N64-style filtering was supported by the Retro Bilinear3 Lit shader, and so on). When you update your pack, you may find that these shaders no longer function properly. If this happens, please swap your materials to use the *Retro Shaders Pro/Retro Lit* shader.

RETRO OUTLINE

A simple hull outline effect which supports many of the same retro-style features of the other effects in the pack. Hull outlines were a common way to achieve a toon look in some PS2 and late PS1 games.

- **Base Color** – Color of the entire outline (no lighting is applied).
- **Thickness** – How far the mesh vertices are extended along their vertex normals in world space units.
- **Snaps Per Meter** – The vertices of the mesh will snap to this number of snap points per meter along each axis.

RETRO TERRAIN LIT

This shader applies the PSX-style effect to Unity's built-in terrains. See above for full property descriptions.

- **Use Stochastic Texturing -** Uses random UV offsets in chunks to break up texture tiling at the expense of using more texture samples overall.
- **Color Depth**
- **Color Depth Offset**
- **Resolution Limit**
- **Filter Mode**
- **Dither Mode**
- **Use Vertex Colors**
- **Snapping Mode**
- **Snaps Per Meter**
- **Lighting Mode**
- **Receive Shadows**
- **Ambient Light Override**
- **Ambient Light Strength**
- **Use Specular Lighting**
- **Glossiness**
- **Use Reflection Cubemap**
- **Reflection Cubemap**
- **Cubemap Rotation**

RETRO SKYBOX

This skybox shader reads from a cubemap texture and applies it to the scene.

RESOLUTION & ORIENTATION PROPERTIES

- **Resolution Limit –** Sets an upper bound on the resolution of **Base Cubemap** and the cloud texture, if either are being used.
- **Rotation** – Adjust the orientation of the skybox around the y-axis.

SKY BACKGROUND COLOR PROPERTIES

- **Background Mode** – Choose whether to use a color gradient or a cubemap for the base sky color.
- **Ground Color** – In *Gradient* background mode, the color of the bottom part of the sky below the horizon.
- **Sky Color** – In *Gradient* background mode, the color at the very top of the sky.
- **Color Mix Power** – Lets you configure which of **Ground Color** or **Sky Color** are more strongly mixed in the sky gradient.
- **Base Color –** In Cubemap background mode, a global multiplier for the sky color.
- **Base Cubemap** – Instead of a regular texture, the skybox samples a cubemap texture for its albedo.
- **Color Depth** – Each color channel is constrained to this many possible values. Low values may darken your image because a floor function is applied.
- **Color Depth Offset** – Applies a slight offset to the colors to avoid the darkening issue.
- **Point Filtering** – Choose whether to force point (nearest-neighbor) filtering for the cubemap.
- **Dithering Mode** – Choose whether to dither in screen space, texture space, or not at all.

PROCEDURAL CLOUD PROPERTIES

- **Use Clouds** – Turn procedurally generated clouds on or off.
- **Cloud Height Threshold** – Controls how far the clouds extend. The first value determines a cutoff point for 0% opacity, and the second value determines at what point the clouds use 100% opacity.
- **Cloud Density Threshold** – Controls the amount of cloud that appears. The first value thresholds the generated noise values. The second value controls where the clouds reach 100% opacity.
- **Cloud Color** – Tint applied to the clouds.
- **Cloud Sizes** – Values used for the noise generator while creating the cloud shapes.
- **Cloud Velocity** – How fast the clouds scroll across the sky.
- **Combine Mode** – Choose which operation to use to combine the two cloud patterns: add, subtract, multiply, or divide.

NOTE: RETRO SKYBOX (PROCEDURAL)

Versions of this asset pack prior to Version 1.5 came with a separate shader for procedural clouds, which could be edited with Shader Graph. It is still included in the extra Retro Base graph package, but all functionality has been rolled into the above Retro Skybox shader.

RETRO DECALS

This decal shader should be attached to the URP Decal Projector to seamlessly project textures onto other objects in the scene. The other shaders in this pack support decal rendering. See above for full property descriptions.

- **Base Color**
- **Base Texture**
- **Normal Map –** An additional texture which can be used to modify the lighting applied to this decal.
- **Normal Blend –** Controls how strongly the decal normals are blended with the existing object normals.
- **Resolution Limit**
- **Color Depth**
- **Color Depth Offset**
- **Enable Dithering**

CRT

The CRT post processing effect covers the entire screen to make it appear like an old-school CRT TV, complete with distortion effects and RGB subpixels.

BASIC SETTINGS

- **Show In Scene View** – Tick to apply the CRT effect while in Scene View.
- **Enabled** – Tick to apply the CRT effect.
- **Render Pass Event** – Choose whether to run the CRT effect before or after URP's internal post processing loop (which includes effects like Bloom).

RESOLUTION & FIDELITY

- **Pixel Size** – An integer value representing how pixelated the image becomes.
- **Scale in Screen Space** – Choose whether to scale the size of other parameters relative to a reference resolution. Enable this when you want pixelation/RGB properties to appear the "same size" on different resolution screens.
- **Reference Resolution (Vertical)** – Set this to the vertical screen resolution you are designing the game around. When Scale Parameters is enabled, each parameter remains unchanged when the screen is at this resolution.
- **Force Point Filtering** – Use nearest-neighbor filtering when upscaling the pixelated image back to full screen size. Disable to use bilinear filtering instead.
- **Interlaced Rendering** – If enabled, the effect will only render every other row of pixels this frame, and then it will render the remaining half of the rows the next frame, and so on.

BARREL DISTORTION

- **Distortion Strength** – Controls how strongly the edges of the screen warp inwards to form the shape of a CRT glass screen.

- **Distortion Smoothing** – How much blending should happen at the edge of the distorted 'screen' shape.
- **Background Color** – Color of the areas outside the distorted CRT screen shape.

RGB SUBPIXELS & SCANLINES

- **RGB Subpixel Texture** – Texture to use for the RGB subpixel effect. All pixels on the screen are multiplied by this texture such that the red, green, and blue screen colors appear separate to each other.
    - Some examples of textures to use for this is contained in "Retro Shaders Pro/Resources/Textures/". Try one of the **RGBTextureX.png** files.
- **RGB Subpixel Strength** – How strongly the RGB subpixel effect is applied.
- **Scanline Texture** – Texture to use for the scanline effect. All pixels on the screen are multiplied by this texture such that scanlines appear scrolling over the image.
    - An example of a texture to use for this is contained in "Retro Shaders Pro/Resources/Textures/ScanlineTexture.png".
- **Scanline Strength** – How strongly the scanline effect is applied.
- **Scanline/RGB Size** – Larger values make the scanlines (and RGB subpixels) appear larger on-screen.
- **Scanline Scroll Speed** – How quickly the scanline texture scrolls over the screen.

VHS ARTIFACTS

- **Random Wear** – Adds small, noisy UV distortions horizontally to simulate the fuzzy look common with old VHS tapes.
- **Aberration Strength** – How strongly chromatic aberration (color channel separation) is applied at the screen edges.
- **Use VHS Tracking** – Should the shader apply the following VHS tracking artifacts?
- **Tracking Texture** – A control texture for tracking artifacts.
    - This should be an x-by-1 image.
    - The red channel controls the strength of the tracking UV offsets.
    - The green channel controls the presence of tracking lines overlaid onto the screen.
    - An example of a texture to use for this is contained in "Retro Shaders Pro/Resources/Textures/TrackingRamp.png".
- **Tracking Size** – How 'zoomed in' the tracking ramp is when scrolling over the screen.
- **Tracking Strength** – How strongly the tracking ramp red channel offsets the UVs of the screen horizontally.
- **Tracking Speed** – How quickly the tracking ramp scrolls across the screen. Use negative values to scroll upwards.
- **Tracking Jitter** – A random offset applied to the scrolling to make it appear jittery.
- **Tracking Color Damage** – Cycle the chrominance of the image slightly to look like the tape is damaged. The screen is converted to YIQ color space, and the offset is applied to the I and Q channels.
- **Tracking Lines Threshold** – A threshold for tracking lines to appear on screen. Higher values mean fewer lines.

- **Tracking Lines Color** – The color of the tracking lines, where the alpha channel acts as a global multiplier on the tracking line strength.

COLOR ADJUSTMENTS

- **Tint Color** – A global tint applied to the entire CRT screen effect.
- **Brightness** – Global multiplier for the image colors before some effects are applied. A value of 1 preserves the image as-is.
- **Contrast** – Forces differences in colors to become more obvious. A value of 1 preserves the image as-is.
- **Color Ramp Mode** – Should the shader apply a color mapping to approximate the color palette of a retro console or computer? This setting comes with several presets.
- **Color Ramp Texture** – The ramp texture used to map screen input colors to new shader output colors.
- **Red Levels** – The number of possible values that the red channel can take on when *Custom RGB Sliders* mode is used.
- **Green Levels** – The number of possible values that the green channel can take on when *Custom RGB Sliders* mode is used.
- **Blue Levels** – The number of possible values that the blue channel can take on when *Custom RGB Sliders* mode is used.
- **Use Dithering** – Should the shader dither colors that fall between color values? Note that individual materials may already be using dithering for their colors, which may look strange when combined with this setting. Can be set when *Custom RGB Sliders* mode is used.

NOTE: COLOR MAPPING RAMPS

The color ramp functionality of the CRT effect only *approximates* the color palettes found in retro consoles and computers. For example, the SNES had a 15-bit RGB palette, but it could only display 256 colors simultaneously; this color mapper only implements the *palette restrictions*, not the simultaneous color restrictions, and some consoles only use a crude approximation.

The included presets implement the following:

- *None* – Don't apply any mapping.
- *Game and Watch* – A 1-bit monochrome effect.
- *Game Boy* – Restricts the color to a 2-bit greyscale, tinted green.
- *Game Boy Advance* – Uses 5 bits per channel (15 total bits, 32,768 total colors).
- *Nintendo DS* – Uses 6 bits per channel (18 total bits, 262,144 total colors).
- *Greyscale* – Converts the image to greyscale based on pixel luminance.
- *NES* – Crude approximation, 3 available values for each channel (27 total colors).
- *SNES* – Same as GBA: uses 5 bits per channel (15 total bits, 32,768 total colors).
- *MSX2* – Uses 3 bits for red and green, 2 bits for blue (8 total bits, 256 total colors).
- *IBM PS/2* – Uses 5 bits for red and blue, 6 bits for green (16 total bits, 65,536 total colors).
- *Amstrad CPC* – Uses 3 levels for each channel (27 total colors).
- *Teletext* – Uses 1 bit per channel (3 total bits, 8 total eye-destroying colors).

- *ZX Spectrum* – Uses one bit per channel, plus additional 'intensity' bit (4 total bits, 16 total colors).
- *Sega Master System* – Uses 2 bits per channel (6 total bits, 64 total colors).
- *Sega Genesis* – Uses 3 bits per channel (9 total bits, 512 total colors).
- *Sega Game Gear* – Uses 4 bits per channel (12 total bits, 4096 colors).

You may also provide your own custom ramp textures to use for mapping.

- With *Custom Luminance*, the luminance of the image color is used to sample along the x-axis of the ramp texture to produce an output color.
- With *Custom RGB*, the individual red, green, and blue channels of the image color are used to sample along the x-axis of the ramp texture to produce individual output red, green, and blue colors.
- With *Custom RGB+Intensity*, the RGB channels are used as described above, then the luminance of the input color is used to sample along the alpha channel of the ramp texture – that value is used to multiply the output RGB value.

If you would rather use sliders rather than color ramps, then the *Custom RGB Sliders* mode exposes integer sliders for each color channel so you can easily set the number of possible color values per-channel.

CRT (MESH)

A version of the CRT effect which can be applied to a regular mesh in your scene. This effect supports most of the parameters of the post-process version, besides the variables relating to screen resolution.

FRAMERATE LIMITER (SCRIPT)

Contained in "Retro Shaders Pro/Scripts", you will find a Framerate Limiter script, which lets you set an FPS upper bound which Unity will aim for. You may find this useful for imitating a poor refresh rate with your game.

DEMO SCENES

The pack contains three demo scenes:

- The *OutdoorDemo* scene contains a terrain-based forest area, showing off the base capabilities of the shaders under a single directional light.
- The *CityDemo* scene contains a small contained city block, showcasing the additional light support of the shaders. This scene works best when under the Forward+ or Deferred rendering paths because Forward has a cap on the number of realtime additional lights it can process.
- The *Bilinear3Demo* scene contains just a couple of crates with N64 bilinear shaders attached.

RETRO GRAPHS PACKAGE (SHADER GRAPH)

This asset pack comes with a small package of shaders recreated in Shader Graph. The purpose of these graphs is to provide a basis for you to create your own effects with the basic features of the Retro shaders, without needing to understand what all the shader code is doing. They do not support all of the features of the main shaders, but I hope that they make it easier to create custom retro effects.

I recommend copying this graph and then modifying the copied version to create your own custom effect.

Warning: The Retro Base Lit graph uses a lot of shader keywords, and as such, you may need to increase the shader variant limit found in **Project Settings -> Shader Graph**. The graph might also be slow when adding/removing/modifying nodes and the generated shader will be large. I have attempted to restrict the number of keywords by removing the ambient light and point filtering toggles.

Copying and modifying one of the code-based shader files, such as Retro Lit, to create your own custom effects will usually (almost always) result in better performance.

SPECIAL THANKS

Many thanks to:

- Contributors to this thread about texel lighting, especially user GreatestBear
- ambientCG for many of the CC0 licensed textures used in the demo
- OpenGameArt for some CC0 licensed textures used in the demo
- Timothy Ahene on Sketchfab for the CRT model used in the promotional art
- Sergej Majboroda on Polyhaven for the sky HDRI used in the promotional art