

NPR by Paidia Technologies

About the library

The NPR library is a collection of post-processing (a.k.a. “image effects” in Unity) Non-photorealistic Rendering (NPR) effects. Post-processing effects in general provide very easy, straightforward and flexible stylization that is completely independent of the models, textures and materials of your game.

Quick start guide

Demo scene is located under Assets/PaidiaTechnologies/NPREffects/Demo/Demo.unity

To add the NPR effects to your scene, simply **attach the NPREffects script to your camera**, located at **Component/Image Effects/NPR/NPREffects**. Only one instance of NPREffects is allowed on each camera. The rendering style is controlled through the script parameters which you can set in the inspector. A good way to get started is to add the “NPR Presets” scripts (see the Utility scripts section) which contains many preset parameter settings to give you an idea about the possible styles and the meaning of the parameters.

Utility scripts

In addition to the NPREffects main effect, the library contains several utility scripts, which are:

- **Component/Image Effects/NPR/ImageProcessor** : this is an implementation of a **general multi-pass post-processing effect**. Simply add your (possibly multi-pass) image processing shaders to its “Shaders” array. The rest, such as compatibility checks, shader and material creation and the execution of the multi-pass rendering are managed by the script.
- **Component/Image Effects/NPR/Control/NPR Presets** : we created this script to help you get started with the parameterization and to make sure you don’t miss an interesting parameter setting. The script contains many preset effects which directly set the parameters of the NPREffects main script. You can also add this script through the inspector view of the NPREffects script using the “Add preset selector script” button.
- **Component/Image Effects/NPR/Control/JointFadeController** : as most effects (Edges, Simplification, Desaturation) have their own “fade” option, it is useful to have this single script which controls the fading options of all effects together. You can also add this script through the inspector view of the NPREffects script using the “Add joint fade control script” button.
- **Component/Image Effects/NPR/Utility/MirrorY** : on some platforms you might experience a vertical flip of the rendered image. Use this script to flip the image back.

Implementation and parameters in a nutshell

Here we provide only a short explanation, for more detailed description and references to the algorithms that were used as a basis for the implementation of our effects we refer to our conference paper titled “Post-Processing NPR Effects for Video Games”, Magdics et al. VRCAI 2013, pp. 147-156, available at: <http://tig01.udg.edu/~rgarcia/publicaciones/Magdics13d.pdf>

- **Edges:** two types of edge detectors are implemented: one runs on the color image (“ImageSpace”), the other works on the depth-normal image (“Geometry”). In both cases, the “strength” of an edge is a floating point value between 0 and 1. The parameters are the following:
 - **Only Edges:** render only the edges.
 - **Edge Type:** selects between color and depth-normal (or combined) edges
 - **Edge color:** color of the edges.
 - **Draw with rendered color:** when only edges are drawn, the edge color can be set to the originally rendered color.
 - **Background texture:** when only edges are drawn, a background texture can be assigned. The edge color will be mixed with this background.
 - **Thickness:** thickness of the edges.
 - **Sharpness:** in general, stronger edges are drawn with more intensive colors. Increasing (decreasing) sharpness results in more (less) intensive colors for weaker edges.
 - **Detailedness:** controls the sensitivity of the “ImageSpace” edge detector. Lower (higher) detailedness results in more (less) edge lines. It is implemented as a prefiltering (blurring) pass: lower detailedness means stronger prefiltering.
 - **Threshold:** the threshold value of the “ImageSpace” edge detector. While this can also control the detailedness, assigning higher values it can be used to create interesting effects.
 - **Fade:** if this is selected, edge thickness varies based on depth or radial distance from the center of the image. A similar technique is often used by artists to emphasize depth: they draw thicker lines in the foreground and thins for the background. The focus point has the thickest edges.
- **Simplify:** this effect is intended to reduce texture and color complexity of the image, creating a somewhat similar effect to toon shading. Texture details are removed using an edge preserving blur, while color complexity is reduced using luminance quantization. Control the effect with its parameters:
 - **Smoothing amount:** higher value: more texture details are removed.
 - **Quantization amount:** higher value means the luminance is more quantized, i.e. there are less colors in the output.
 - **Fade:** the amount of smoothing can vary based on depth or the radial distance from the center of the image. The focus point has the weakest smooth, i.e. the most details.
- **Shadow Coloring:** pixels which are in shadow can be recolored. For instance, in comics constant black or complementary color (of the shadowed surface) are often used for shadows. As there is no direct access to the shadow map in Unity, the shadowed pixels are obtained by rendering the scene again with identical camera settings but without shadows and comparing the result with the original image. Note that this additional rendering can reduce performance for complex scenes. Only the basic camera settings are copied, additional image effects are not. Thus, to work together with other image effects, make sure they are executed after the NPREffects script (placed under it in the inspector view).

- **Smooth Shadow:** when set, shadow color is weighted with the “intensity” of the shadow (works especially well together with soft shadows). Turn it off to completely replace shadowed pixels with the shadow color.
- **Shadow Color:** the color with which shadowed pixels are replaced/blended.
- **Hue offset:** the hue (see the HSV or HSL color spaces) of the recolored shadows can be shifted to create interesting effects. For instance, when the Intensity Scaler is set to 0 (i.e. only the surface color is rendered), and Hue offset to 0.5, the shadow will be drawn with the complementary color of the shadowed surface.
- **Intensity Scaler:** how to mix between the surface color and the shadow color. 0 means only the original surface color is used (everything will look as if shadows were turned off).
- **Hide edges:** turn this on to avoid edge drawing on the shadowed pixels.
- **Negative edges:** when set, edges in shadows are drawn with “negative” colors. For instance, when only edges are drawn and pure black shadows are selected, set this parameter to have white edges in the black shadows instead of black (and thus invisible) edges.
- **Desaturation:** color saturation is reduced based on the fade settings.
- **Color transfer:** the image is re-colored, the color histogram is changed to resemble that of a user-defined source image.
 - **Source:** an array of textures containing the source (a.k.a. exemplar) image(s). Only one can be active at a time. The color statistics (average color and variance) are obtained from this source image and “transferred” to the camera’s rendered image, so the rendered and the source image will have similar color mood.
 - **Apply Before Other Effects:** certain effects, such as edge coloring, shadow coloring or desaturation may change the color distribution of the render image and thus affect the result of the color transfer. Turning this parameter on executes color transfer **before** all other effects. For instance, executing color transfer after desaturation may also change the desaturated (greyish) color, while executing color transfer before desaturation results in the “regular” behavior of desaturation.
 - **Color space:** the color statistics that are extracted from the source image can be transferred to the rendered image in different color spaces, producing different results. The “best” option depends on the source image and also on your personal taste.
 - **Current style:** the index (in the Source array) of the source image to be used.
 - **Variance scale:** the transferred color variance is scaled by this parameter. Lower values produce less colorful images and may result in flickering, while higher values may lead to the overflow of pixel values but reduce flickering.
- **Fade effects (fade control parameter):** fade is intended to create variable abstraction level on the image. Currently, the following parameters can “fade”: edge thickness (Edges), smoothing amount (Simplify) and color desaturation (Desaturate).
 - **Fade Type:** fade depends either on the distance from camera (i.e. the pixel depth – “DEPTH_BASED”) or the radial distance of the pixel from the image center (“RADIAL”).
 - **Focus:** in the focus point (e.g. focused depth value) the image are the most detailed and details are the most emphasized (i.e. thickest edges, weakest smoothing and maximum saturation). Note that when DEPTH_BASED fade is set, the Focus value is in the same range as the depth buffer values, so it is affected by the camera’s near and far plane settings.
 - **Max value:** the fading parameter (edge thickness, etc.) can be further scaled. For example, set it to 2 to increase the color saturation of the pixels near the focus.

- **Change speed:** how fast the parameter fades. The minimum value practically means no fade, high values result in abrupt change.
- **Levels:** the fading parameter (edge thickness etc.) can be quantized, “Levels” controls the number of bins. See the DESATURATE_RADIAL_CIRCLES preset effect for an example.
- **Both direction:** when selected, fading also happens from the focus point towards the origin of the distance calculation (i.e. towards the camera for DEPTH_BASED, or the image center for RADIAL).

Compatibility issues

- Camera’s Depth Texture is not supported in several platforms, which means **depth-based fade effects don’t work on these platforms**.
- On some machines we experienced **vertical flip** even for the simplest image effects, contrary to the documentation: <http://docs.unity3d.com/Manual/SL-PlatformDifferences.html> . If the flipping occurs, use the MirrorY utility script to flip the image back.
- **Shadow effects don’t work together with anti-aliasing** yet (e.g. “Beautiful” or “Fantastic” rendering quality settings by default) and thus these are disabled automatically.
- As shadows use two different cameras, **low depth precision may cause flickering with shadow effects**. Try using higher precision depth buffer (higher number of bits or increase the camera’s near plane parameter) to avoid this.
- We noticed that some effects don’t work properly in **Webplayer mode when built under Mac** and viewed on Windows. Weplayer build made on Windows was working on both platforms.

Performance Issues

Every effect runs in image space, meaning that performance drops with increasing pixel resolution. Additionally, shadow recoloring requires an additional rendering of the scene, which might result in a great performance drop for complex scenes.

To increase your FPS rate, try the followings:

- Turn off shadows
- Reduce the “Smoothing Amount” parameter of the Simplify effect
- Switch to “Geometry”-type edges

Contact

Paidia Technologies, <http://paidiatech.com>

Please send us feedback or bug report to paidia.tech@gmail.com