Retro: Pixelator

Pixelator is a powerful post-processing effect for Unity's Universal Render Pipeline (URP) that allows you to create a wide variety of retro-inspired visuals. It offers highly customizable pixelation, advanced color manipulation including posterization and dithering, unique visual effects like beveling and chromatic aberration, and a suite of color filters.

Parameters

Below is a detailed description of all the parameters available in the Pixelator settings.

Main Settings

- **Intensity**: `float` (Range: 0.0 to 1.0, Default: 1.0) Controls the overall intensity of all combined Pixelator effects. An intensity of 0 means the effect is not applied.

Pixelation

- **Pixelation Mode**: `enum` (Default: Rectangle) Determines the shape and algorithm used for pixelation. Available modes:
  - `Rectangle`: Classic square/rectangular pixelation.
  - `Circle`: Pixels are represented as circles.
  - `Triangle`: Pixels are represented as triangles.
  - `Diamond`: Pixels are represented as diamonds.
  - `Hexagon`: Pixels are represented as hexagons.
  - `Leaf`: A leaf-like pattern for pixel cells.
  - `LED`: Simulates an LED screen display.
  - `Knitted`: Simulates a knitted or cross-stitch pattern.
- **Pixel Size**: `float` (Range: 0.0 to 1.0, Default: 0.75) Controls the perceived size of the pixels. The exact interpretation can vary slightly per mode.

Mode-Specific Pixelation Parameters:

- **For `Rectangle`, `Triangle`, `Hexagon`, `Leaf` modes:**
  - **Screen Aspect Ratio**: `bool` (Default: true) If true, uses the screen's aspect ratio to calculate pixel scaling. If false, `Custom Aspect Ratio` is used.
  - **Custom Aspect Ratio**: `float` (Range: 0.2 to 5.0, Default: 1.0) Only used if `Screen Aspect Ratio` is false. Defines a custom aspect ratio for pixel scaling.
  - **Pixel Scale**: `Vector2` (Default: (1,1)) Allows non-uniform scaling of pixels along X and Y axes.
- **For `Circle`, `LED` modes:**

    - **Radius**: `float` (Range: 0.0 to 1.0, Default: 0.5) Controls the radius of the circular or LED elements within each pixel cell.
    - **Background**: `Color` (Default: Black) The color shown in the gaps between circular/LED elements if their radius doesn't fill the cell.

- **For `Knitted` mode:**
    - **Threads**: `int` (Range: 1 to 8, Default: 3) Simulates the number of threads in the knitted pattern.
    - **Pixel Scale**: `Vector2` (Default: (1,1)) Controls the scale of the knitted pattern.

Gradient Mapping

Maps the screen colors to a custom gradient texture.

- **Gradient Intensity**: `float` (Range: 0.0 to 1.0, Default: 0.0) The intensity of the gradient mapping effect. 0 means disabled.
- **Gradient**: `UnityEngine.Gradient` (Default: Grayscale gradient) The gradient to use for color mapping. This is baked into a texture at runtime.
- **Luminance Min**: `float` (Range: 0.0 to 1.0, Default: 0.0) The minimum input luminance that maps to the start of the gradient (for Luminance mode).
- **Luminance Max**: `float` (Range: 0.0 to 1.0, Default: 1.0) The maximum input luminance that maps to the end of the gradient (for Luminance mode).
- **Mapping Mode**: `enum` (Default: CIELAB) Determines how colors are mapped to the gradient.
    - `Luminance`: Maps based on the input pixel's luminance.
    - `CIELAB`: Maps based on perceptual color similarity in the CIELAB color space, finding the closest color in the gradient.
- **CIELAB Samples**: `int` (Range: 2 to 64, Default: 16) Number of samples taken along the gradient texture when `Mapping Mode` is `CIELAB`. Higher values are more accurate but slower.
- **Apply Luminance**: `bool` (Default: true) If true, the luminance of the original pixel (or mapped luminance for Luminance mode) is multiplied with the final mapped gradient color. This can preserve some of the original brightness characteristics.

Dithering

Simulates colors by using patterns of a limited color palette.

- **Dither Intensity**: `float` (Range: 0.0 to 1.0, Default: 0.5) The strength of the dithering effect.
- **Pattern Scale**: `int` (Options: 2, 4, 8, Default: 4) The size of the Bayer matrix used for ordered dithering (2x2, 4x4, or 8x8).
- **Threshold Scale**: `float` (Range: 0.0 to 1.0, Default: 0.75) Adjusts the influence of the dither pattern.
- **Color Steps**: `int` (Range: 2 to 16, Default: 8) The number of discrete color steps per channel that the dithering will attempt to simulate.

Posterization

Reduces the number of distinct colors in the image.

- **Posterize Intensity**: `float` (Range: 0.0 to 1.0, Default: 0.5) The overall strength of the posterization effect. 0 effectively disables it.
- **RGB Steps**: `Vector3Int` (Range: 2 to 256 per channel, Default: (24,24,24)) Number of color steps for Red, Green, and Blue channels respectively when RGB posterization is active.
- **Luminance Steps**: `int` (Range: 2 to 256, Default: 24) Number of steps for the luminance channel when Luminance posterization is active.
- **HSV Steps**: `Vector3Int` (Range H: 2-64, S: 2-32, V: 2-32, Default: (24,24,24)) Number of color steps for Hue, Saturation, and Value channels respectively when HSV posterization is active.
- **Gamma**: `float` (Range: 0.1 to 3.0, Default: 1.0) Applies gamma correction before posterization and de-correction after. Values other than 1.0 can change perceived brightness and color relationships.

Bevel Effect

Adds a pseudo-3D bevel based on color differences, giving a chiseled look.

- **Bevel**: `float` (Range: 0.0 to 10.0, Default: 1.0) The strength and depth of the bevel effect.

Chromatic Aberration

Simulates lens distortion by offsetting color channels.

- **Chromatic Aberration Intensity**: `float` (Range: 0.0 to 10.0, Default: 1.0) The overall strength of the chromatic aberration effect.
- **Offset**: `Vector3` (Default: (1.0, 2.0, -1.0)) The amount by which the Red, Green, and Blue channels are shifted, respectively.

Color Filters

Applies various stylistic color filters.

- **Filters Intensity**: `float` (Range: 0.0 to 1.0, Default: 0.0) The global intensity for all active color filters. Lerps between the original color and the filtered color.

- **Sepia**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of the sepia tone filter.

- **Cool Blue**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a cool blue tint filter.

- **Warm**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a warm orange/yellow tint filter.

- **Invert Color**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of the color inversion filter.

- **Hudson**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a filter emulating the Hudson Instagram effect (cool, vignetted).

- **Hefe**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a filter emulating the Hefe Instagram effect (high contrast, warm, vignetted).
- **X-Pro**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a filter emulating the X-Pro II Instagram effect (high contrast, saturated, warm cast, vignette).
- **Rise**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a filter emulating the Rise Instagram effect (soft, warm, desaturated).
- **Toaster**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a filter emulating the Toaster Instagram effect (strong vignette, warm center, burnt edges).
- **Infrared**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of an infrared photography simulation filter.
- **Thermal**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a thermal camera/heat map style filter.
- **Duotone**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of the duotone filter, which maps image luminance between two chosen colors.
    - **Color A**: `Color` (Default: Dark Blue) The first color for the duotone effect (typically for darker areas).
    - **Color B**: `Color` (Default: Bright Yellow) The second color for the duotone effect (typically for lighter areas).
- **Night Vision**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a night vision goggle simulation filter.
- **Pop Art**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a Pop Art style filter using a limited, vibrant color palette.
- **Blueprint**: `float` (Range: 0.0 to 1.0, Default: 0.0) Intensity of a blueprint-style filter with edge detection.
    - **Edge Color**: `Color` (Default: Light Blue) The color for detected edges in blueprint mode.
    - **Background Color**: `Color` (Default: Dark Blue) The background color for blueprint mode.
    - **Edge Threshold**: `float` (Range: 0.05 to 0.5, Default: 0.1) The threshold for edge detection in blueprint mode.
