

LutLight2D

Color Replacement Lighting

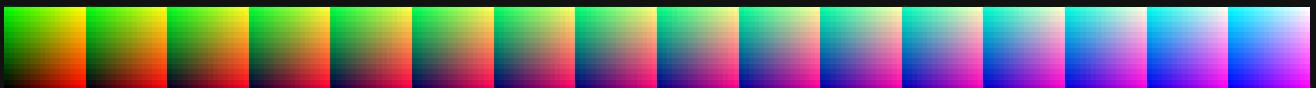
This is a short guide explaining how to start using color replacement lighting. It briefly explains how it works and what to do to configure a material for the minimal example.

Quick Guide

- The Material is created via LutLight2DAsset [Create\2D\LutLight2D Asset](#)
- For lighting uses texture from **Urp2DRenderer**, so for the shader to work by default **UrpAsset** must be configured
- Examples with techniques can be found in **ProjectSamples**

How does color replacement work?

Regular color replacement or correction works through Lut (Look-Up Table), which is a 3D texture with XYZ coordinates from which color is selected for replacement based on its RGB components (used as coordinates). Lut tables are not stored in 3D due to compatibility issues, but they are used as an unfolded 2D texture from which color is taken, simulating a Lookup.



- neutral unity lut

* Color is usually taken with interpolation because the texture resolution is limited and does not allow for all shades to be conveyed. However, for PixelArt color replacement, especially for lighting, it can be used without interpolation, similar to the Point Filter in a regular texture. This is done to prevent colors from blending.

Main Idea

The main idea of this asset is to use manually defined color gradients for each color in the original image.

Common lighting usually works in such a way that the color is simply multiplied or added to the color from the light texture, this adding new colors and shades where they shouldn't be, breaking the stylization defined in the palette, which doesn't work as depicted in the artwork.



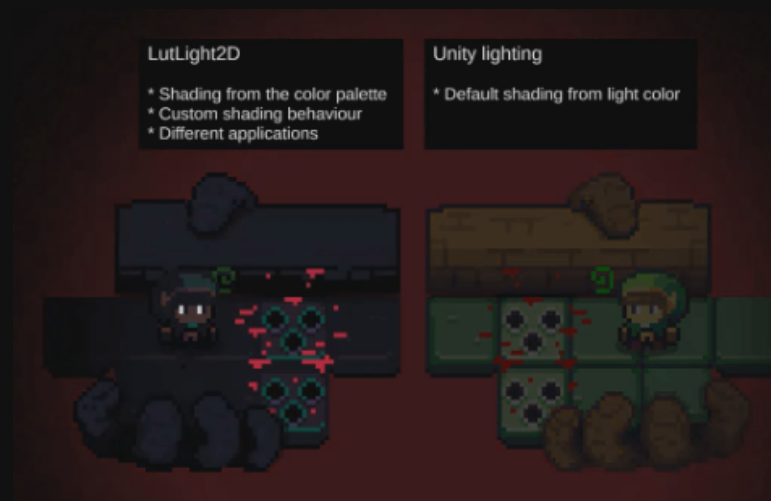
In pixel art transition from light to shadow is usually defined using Color Ramps, with different colors used depending on the stylization (usually each material has its own gradient)



LutLight generates a set of Lut tables for each color gradient, where each Lut represents a Row from the gradient table to which the base color is replaced.

Due to the fact that the lighting colors are defined manually, it became possible to create materials that are visible only in light or shadow, reacting differently to illumination. Allows to define shading patterns, create stylization elements

like outlines in shadows, objects that become brighter in shadows, or objects that are not completely shaded, cartoon effect.

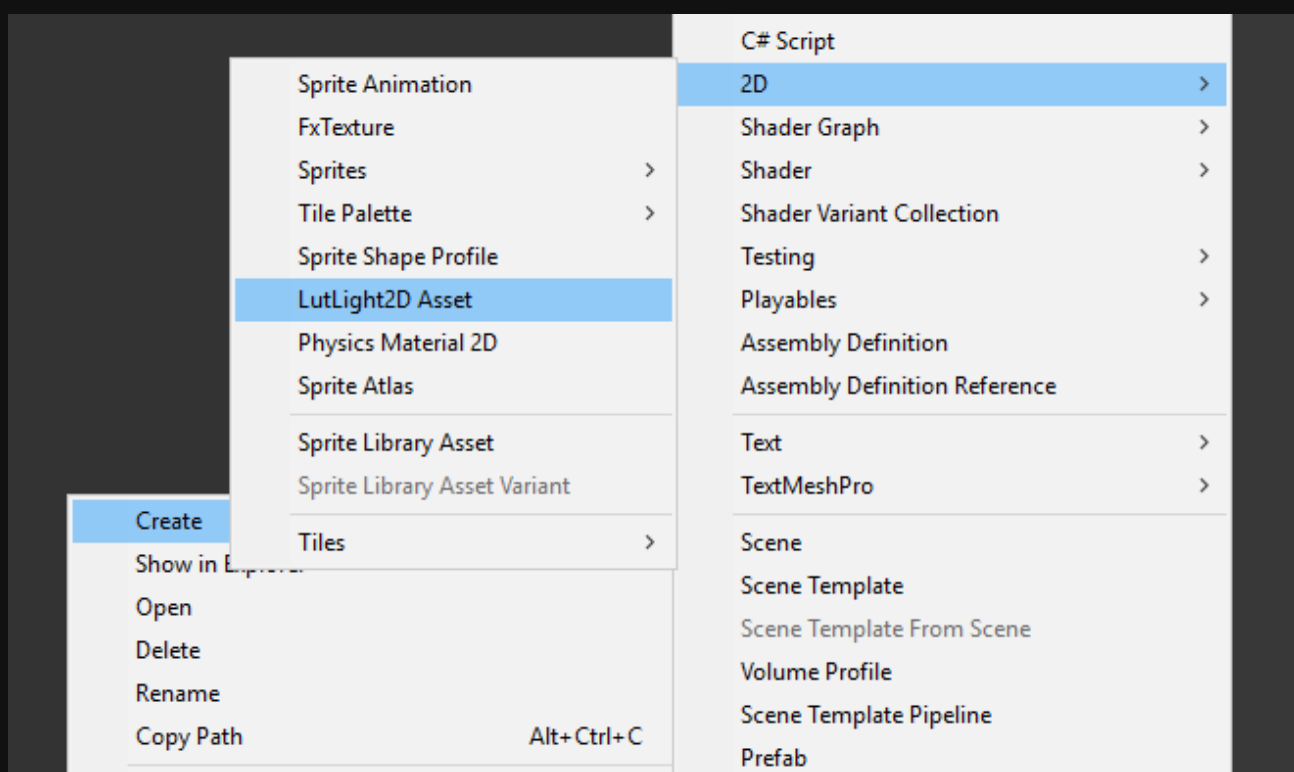


How to Use?

- shader uses the lighting texture from the Urp 2D Renderer so Urp Asset must be configured ! (but can use a custom one)

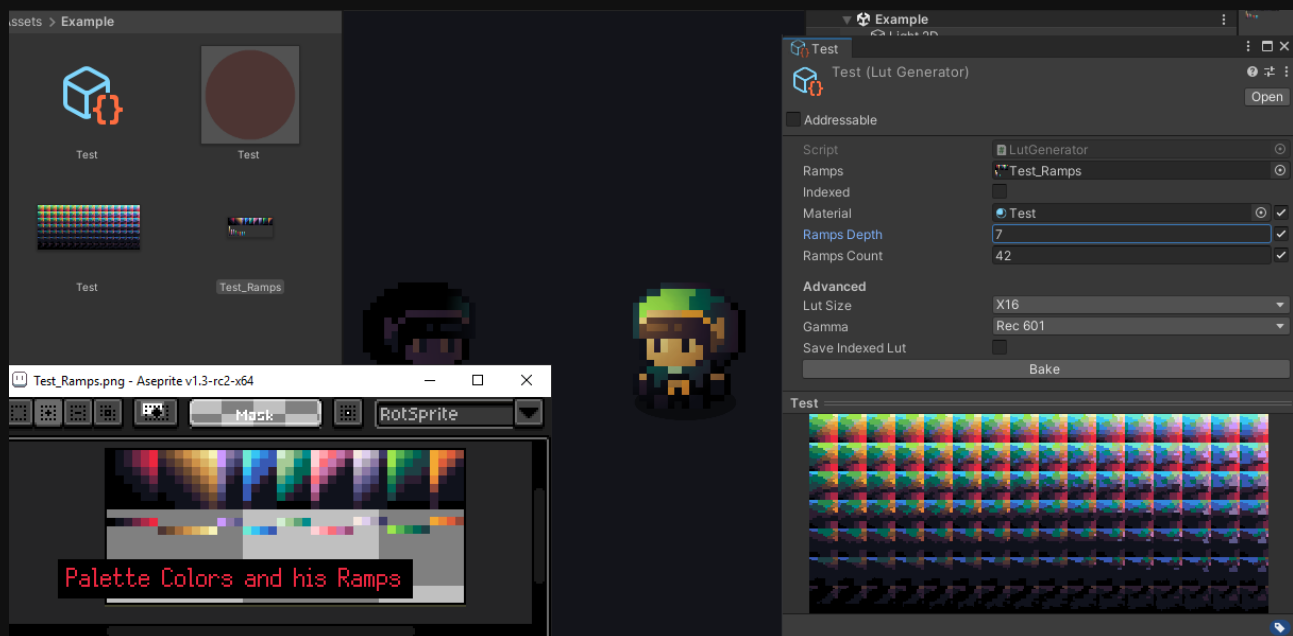
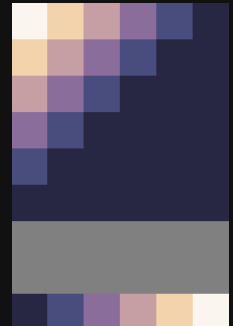
To make the material work, it needs a set of LUT tables. To simplify their creation and material configuration, there is a LutGenerator that converts a PNG image containing these ramps into a texture with a set of LUT tables.

To create it, you need to select Create\2D\LutLight2D Asset.



After that, it will create a file with a monochromatic gamma from which a set of LUT tables and a material to which all of this is applied will be generated.

The file with the ramps of the materials consists of gradients going from top to bottom, where the topmost color is used as the key for color replacement. In other words, each color in the palette should have its own shading ramp.



By default, the main color is preserved and used as an indexer for overwriting in the LUT table and is not replaced. This means that at maximum brightness, it will always look the same. However, sometimes you need to have several materials that look the same at maximum brightness but react differently to light. For this purpose, there is an indexing mode, in which the first column of gradients is used for indexing the color of the original image.

Describing each example in detail would take a long time, but knowing how everything works, you can see examples in Project Samples, where some techniques of using lighting through color replacement are implemented.

