# Color Texture Generator

v1.1.1

This simple editor tool will let you generate color textures that can be used in UI, shaders, or scripts to achieve various effects.

Core features:

- Generate a texture from Unity's built-in gradient.
- Generate a texture from the list of specified colors and their weights.
- Save created gradients and color ramps in scriptable objects, so they can be modified later.

[0] Installation

After you download and import the package from the Asset Store, a new menu item will appear under the 'Window' tab. Click on it to open the window.
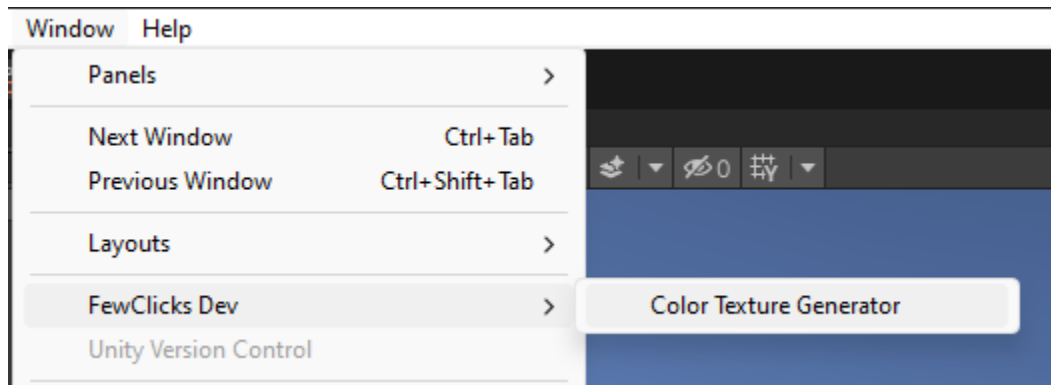
[1] Gradient

In this tab, you can generate a texture from Unity's built-in gradient. Texture's export settings are shared between both tabs.
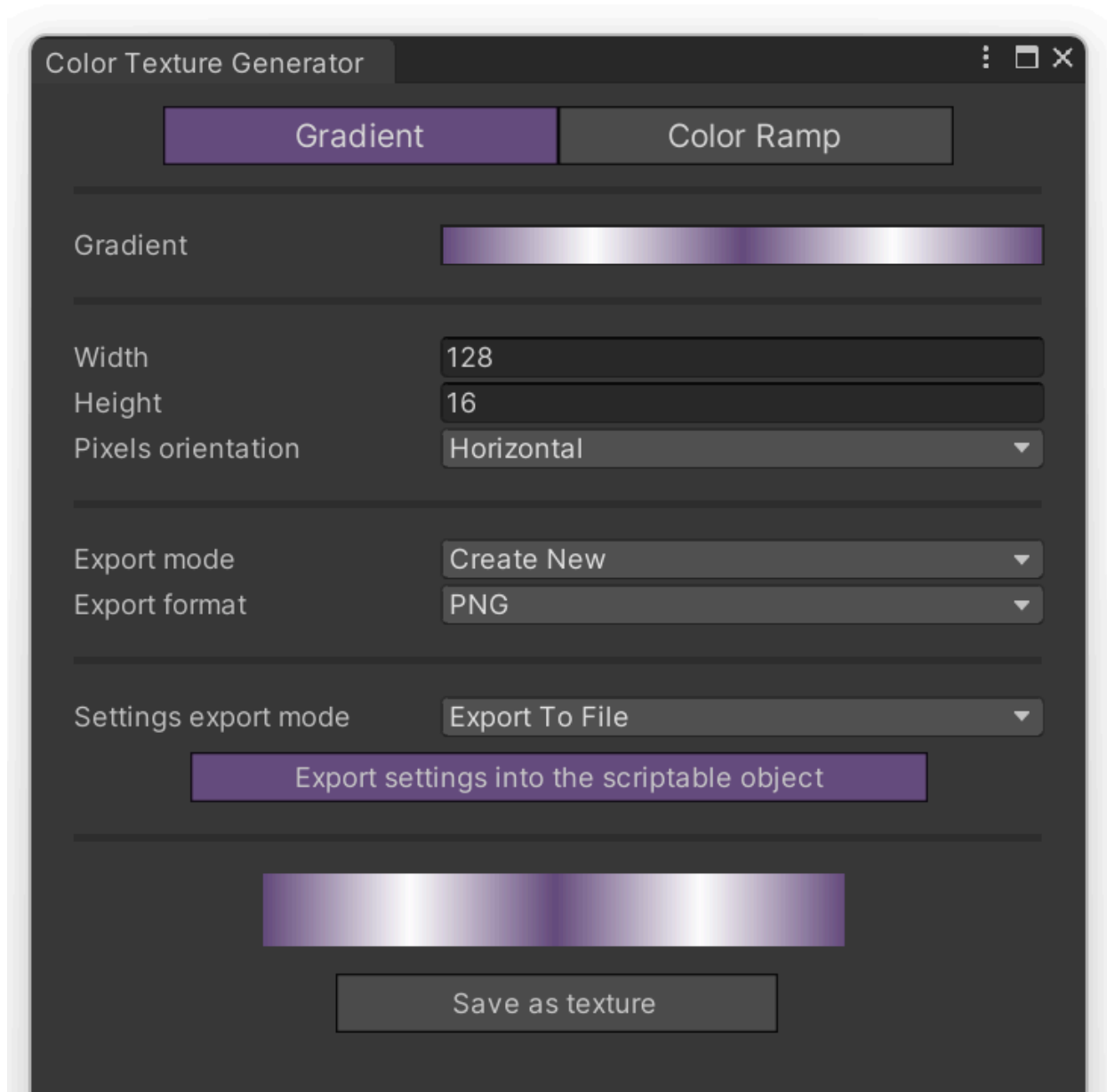
Width - the width of the generated texture in pixels.

Height - the height of the generated texture in pixels.

Pixels orientation - orientation of the pixels in the generated texture. You can choose from horizontal, reversed horizontal, vertical, and reversed vertical.

Export mode - this option allows you to override the existing texture or create a new one.

Export format - texture format. You can choose from PNG, TGA, and JPG.

Settings export mode - this option allows you to create a new gradient file (scriptable object) with all settings from the window, or override settings in the existing file.

Gradient object

This scriptable object is used to store gradient reference, with texture to override and export settings.
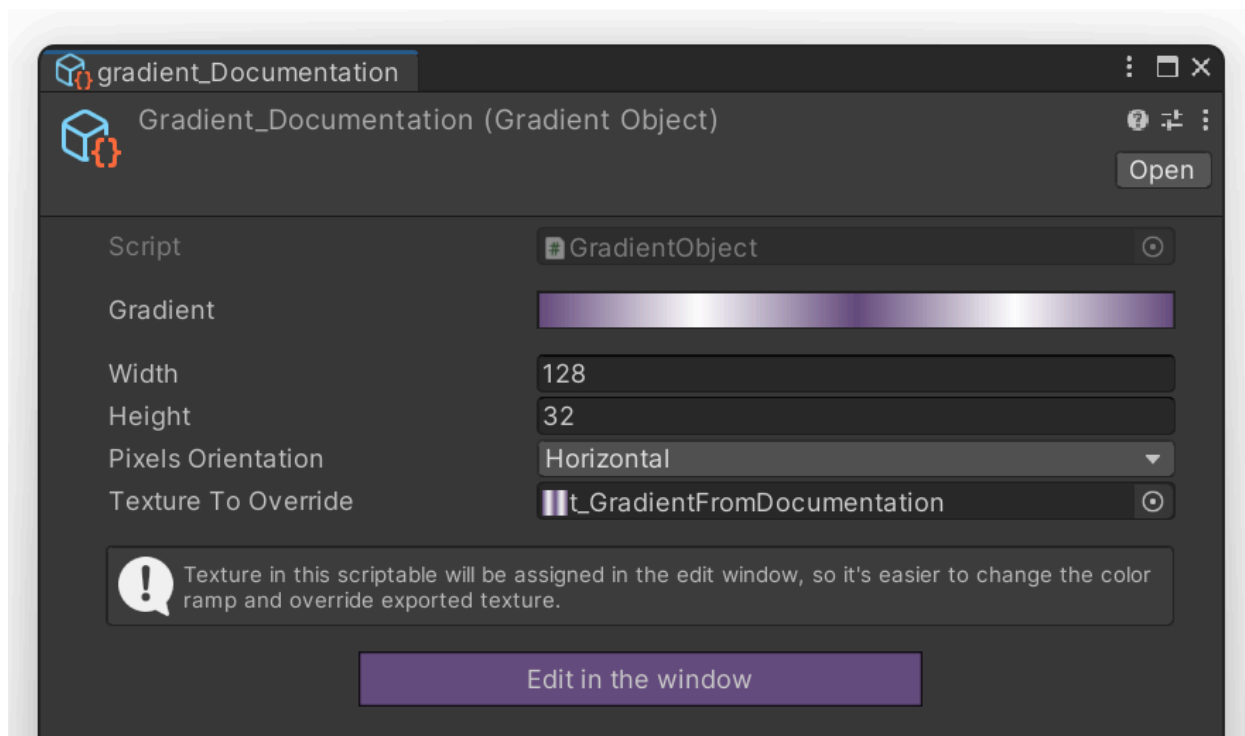
The button at the bottom can be used to quickly apply settings in the Color Texture Generator Window. You can use a scriptable object to update already generated gradients without losing references to them. Simply click the 'Edit in the window' button, make changes, override settings, and generate the texture.

[2] Color Ramp

In this tab, you can generate a texture from a list of colors, with their relative coverage and optional blending. As it was mentioned before, texture's export settings are shared between both tabs.

Color Texture Generator

| Gradient | Color Ramp |

| Index	Color		Coverage	Color lerp	
01		🖉	1	■ 1	X
02		🖉	1	■ 1	X
03		🖉	1	■ 1	X
04		🖉	1	■ 1	X
05		🖉	1	■ 1	X
06		🖉	1	■ 1	X
07		🖉	1	■ 1	X
08		🖉	1		X

| Reset list | Reset coverage | + |

Width   128
Height   32
Pixels orientation   Horizontal ▼

Export mode   Override Existing ▼
Texture to override ▐▌t_ColorRamp_Documentation_01 ⊙

Settings export mode   Override Existing ▼
Color ramp to override colorRamp_Documentation_01 (Col ⊙

| Override color ramp in the attached file |

| Override texture |

Index - index of the color. You can choose up to 8 colors (but feel free to increase or decrease this value in the ColorRamp.cs file).

Color - a color that will be used in the generated texture.

Coverage - relative color coverage on the texture. If you set one color coverage to two, when leaving all others to default one, it will be twice as wide.

Color lerp - can be used to lerp current color with the next. It's also relative to all color coverages.

Here, you can see how changing the coverage affects the generated texture.
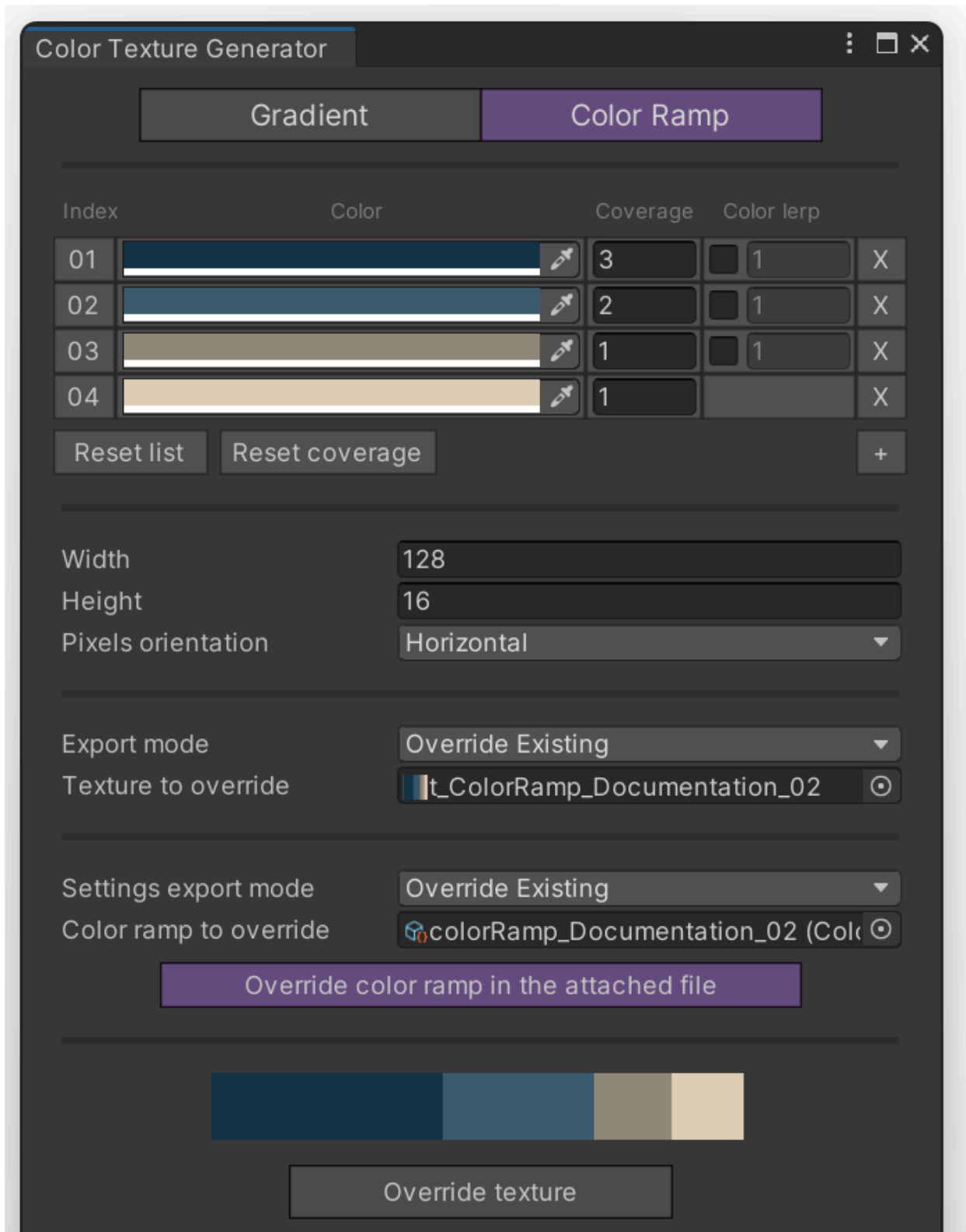
The picture below shows how texture generation is affected by enabling the color lerp flag.
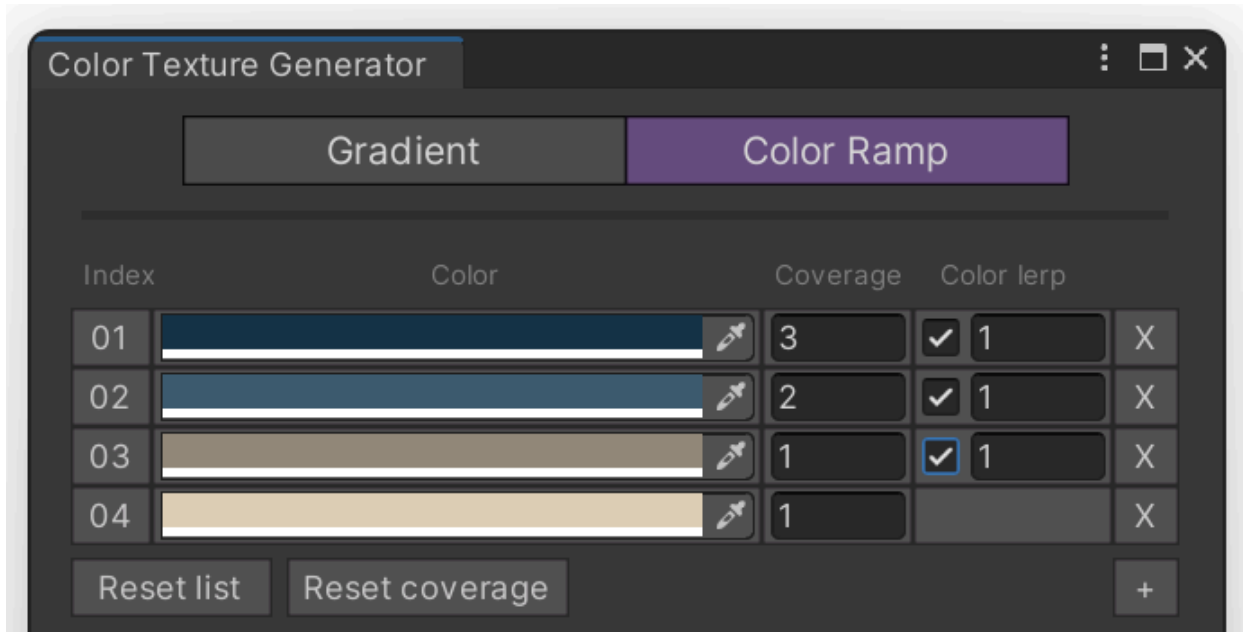
Color ramp object

This scriptable object is used to store color ramp colors, with texture to override and export settings.

The button at the bottom can be used to quickly apply settings in the Color Texture Generator Window. You can use a scriptable object to update already generated color ramp textures witho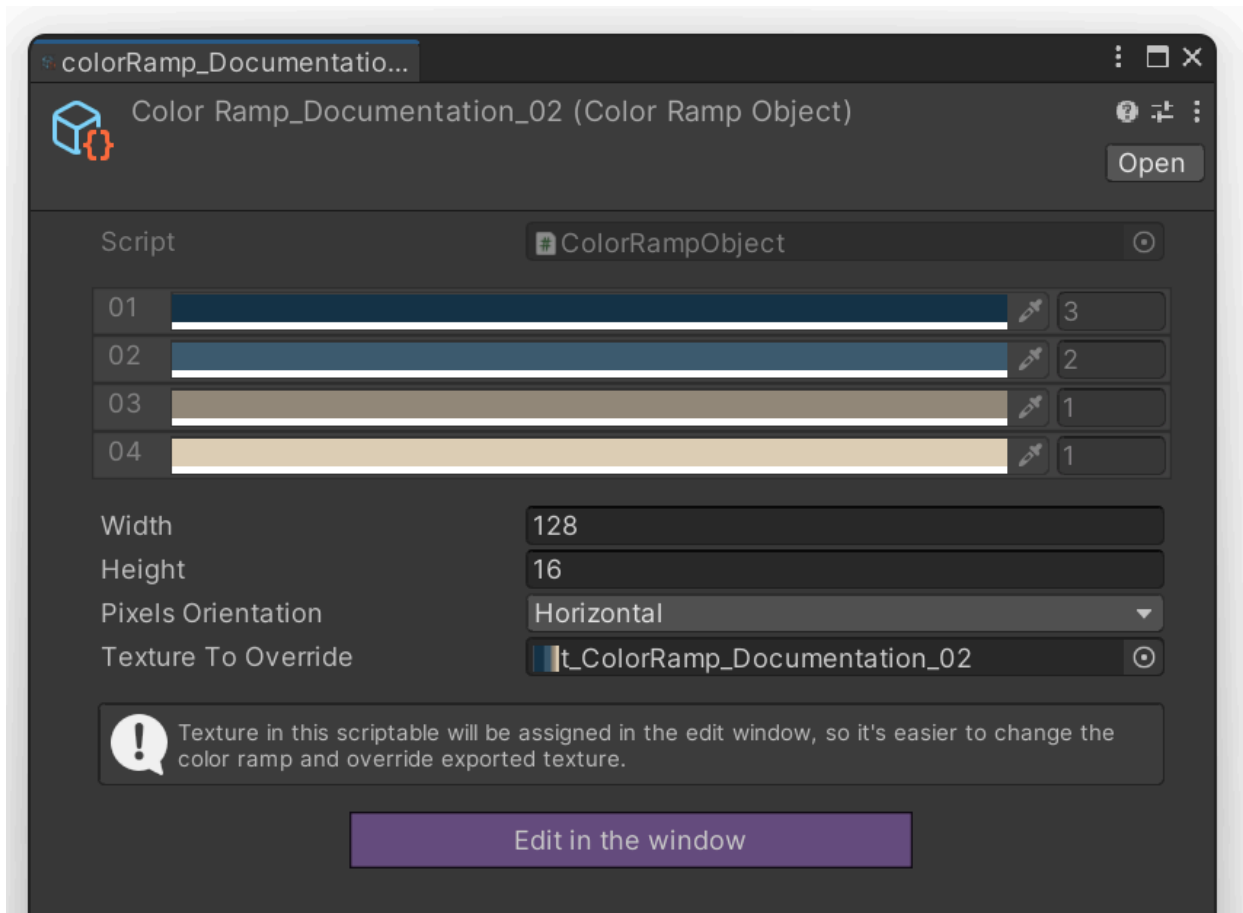ut losing references to them. Simply click the 'Edit in the window' button, make changes, override settings, and generate the texture.

[3] FAQ

Does the export format matter?

Your generated texture should look the same in every format, except for JPG, where you can't save alpha values. You should stick with the one that you already use in the project. Here is a Reddit link that will explain the differences.

https://www.reddit.com/r/explainlikeimfive/comments/yp499/eli5_what_is_the_difference_between_bmp_jpg_png/

What's the perfect texture size?

Depending on the use case, you should always go with as small a texture as possible to save some space. If you plan to use the generated texture in the shader, each value should be a power of two (16, 32, 64, etc.). For textures that will be used as sprites, it's enough that every texture dimension can be divided by four.

Another factor that you should consider is the precision of color change that you want to achieve.

[4] Review and Feedback

If you enjoyed using this tool please consider leaving a review on the **[Unity Asset store](#)**!
Thank you very much for any feedback.

Need some help or have an idea how to make this tool even better? Don't hesitate to write me an email at **contact@fewclicksdev.com**.