

LightShade Shader

A plugin for Cinema 4D

This simple plugin was written to meet a specific purpose I required, but it occurred to me that others might find it useful.

This program is distributed in the hope that it will be useful, but it carries no warranty of any kind. There is no implied warranty of fitness for any purpose and you use it at your own risk.

What it does

This is a channel shader which is intended to be placed in a channel of a material. It can be placed in any channel but the effects vary depending on the selected mode.

In 'Alpha' mode, the shader returns a brightness value between 0 and 1 depending on the illumination of the surface by a specified light or lights. If a surface point is not illuminated by the light, the shader returns 0; if it is brightly lit by the light, it returns 1. Intermediate values can also be returned depending on the brightness of the surface illumination.

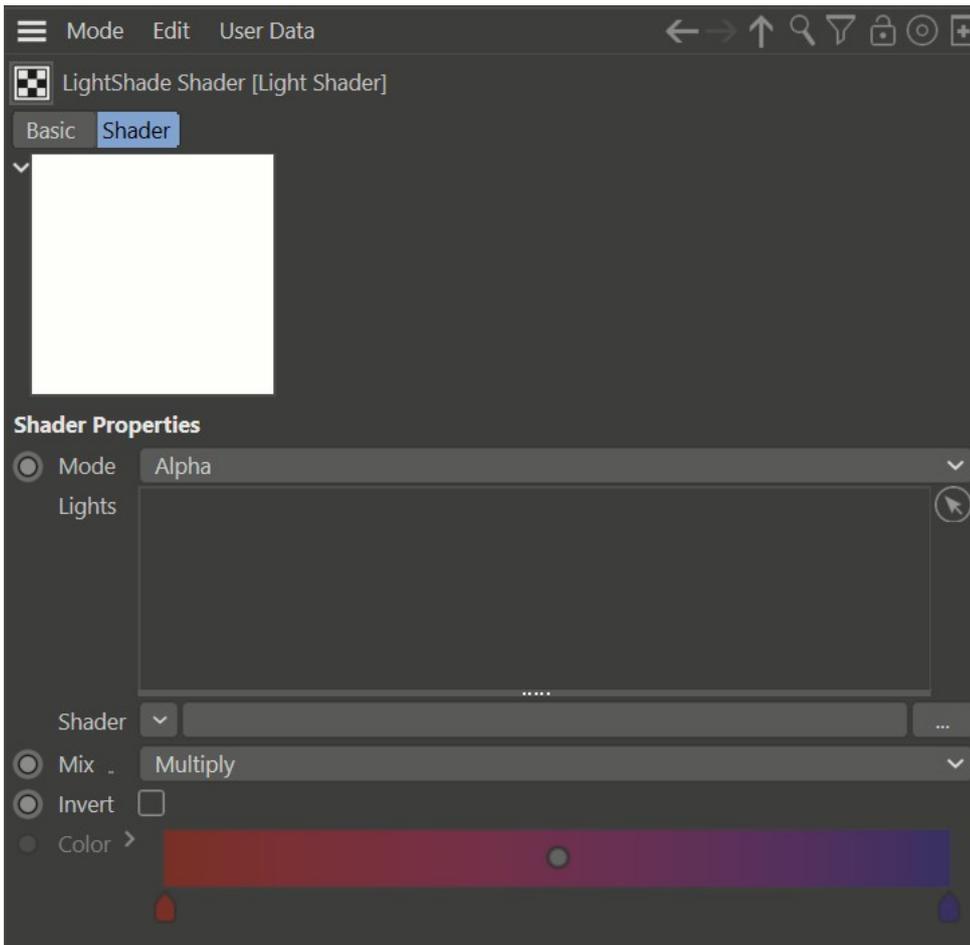
In 'Color' mode, the brightness returned is then used to select a color from a gradient using the value as an index into the gradient.

Possible uses

You can use this shader to:

- color a surface depending on the degree of illumination of the surface
- add transparency to a surface depending on the illumination
- mix two textures on an object using the shader in the alpha channel of a material with the blend between the two controlled by the surface illumination

In addition, the returned value can be combined with another shader for some interesting effects.



Mode

The mode to use. 'Alpha' returns a grayscale value which can be used in any channel which needs one, such as transparency, alpha, displacement, etc.

'Color' mode returns a color. The color used is selected from the 'Color' gradient; a surface brightness of zero selects the color from the left-hand end of the gradient while a value of 1 selects the color from the right-hand end.

Lights

Drag any light you want to use into this list. Each light has two flags. The green tick and red cross icon behave in the usual way: the light won't be used at all (i.e. as though you had removed it from the list) if the red cross is shown.

The second flag is either a blue plus or a red minus. If it is plus, the light brightness is *added* to the overall brightness from all the lights in the list. If it is a minus, the light brightness is *subtracted* from the overall brightness. In all cases, note that the final brightness value always ranges between 0 and 1.

Shader

This link field allows a shader or bitmap to be added. After the overall surface illumination has been calculated, this shader is sampled, and the brightness of the returned color is combined with the surface illumination. The method of combination is given in the 'Mix' parameter.

Mix

Determines how the surface illumination and shader brightness are combined. The options are:

- Multiply: the two values are multiplied together (e.g. if the illumination value is 0.75 and the shader brightness value at the same surface point is 0.5, the final value is $0.75 \times 0.5 = 0.375$)
- Add: the illumination and brightness values are added together but the result will not exceed 1.0
- Subtract: the shader brightness value is subtracted from the illumination value, but the result will not be less than 0

Invert

If this switch is checked the calculated brightness value is inverted. That is, a value of 0 becomes 1, 0.25 becomes 0.75, etc.

Color

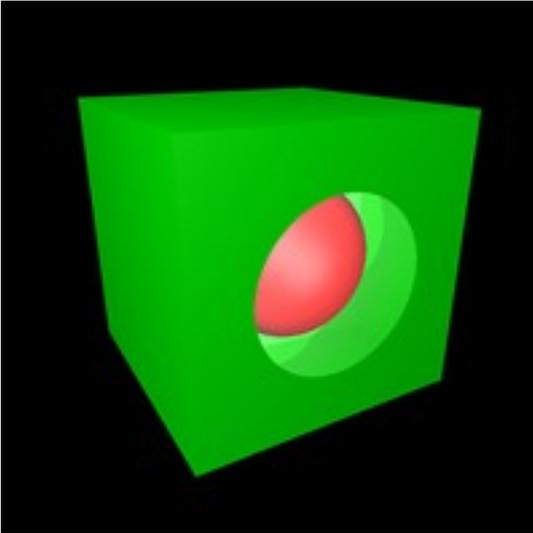
The color gradient to use when 'Mode' is set to 'Color'.

Examples

Here are some examples of what this shader can do.

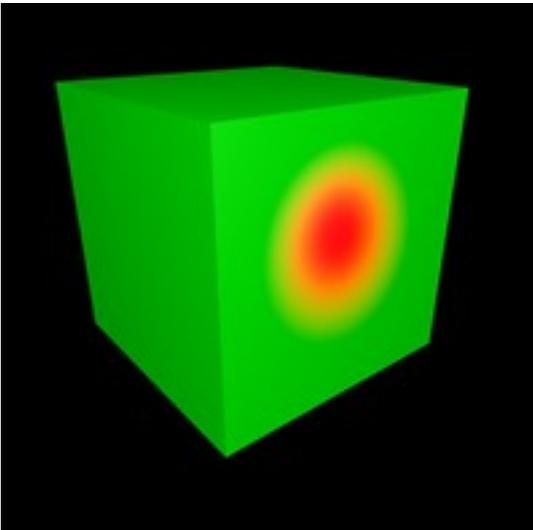
E1. Transparency

In this image, there is a sphere with a red material inside a cube with a green material. The green material has a transparency channel with a LightShade shader in it, so that the sphere is revealed where the spotlight shines. (There is no hole in the geometry, this is entirely due to transparency.)



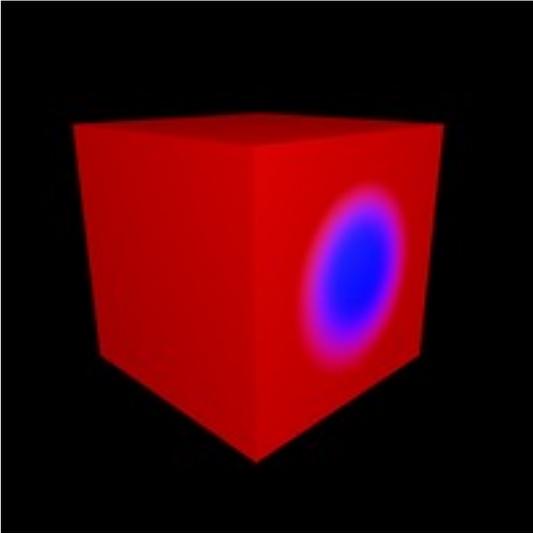
E2. Alpha

Using this shader in the alpha channel allows the mixing of textures on an object according to the illumination. In this scene, the cube has two materials applied, one red and one green. The green material texture tag comes after the red one in the object manager, so normally you would only see green. Using this shader in the alpha channel of the green material and a spotlight to illuminate the surface means that where the light shines, the alpha value of the green material is 1 (completely opaque) but when this is inverted, the red material shows through where the light shines:



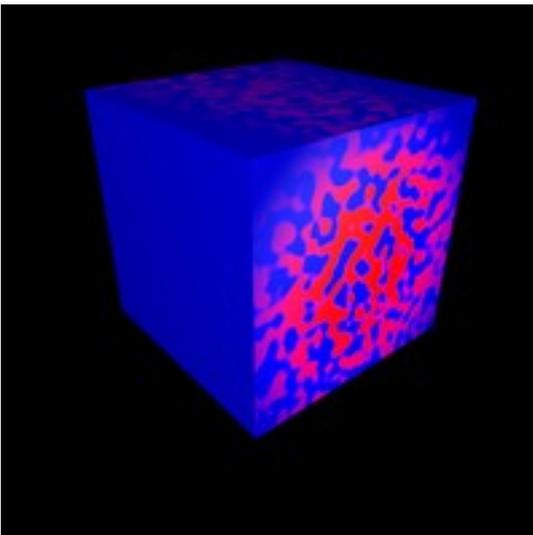
E3. Color

In this image, both the material and the light are white but the LightShade shader is in the color channel of the material. Color mode is used with the default gradient, so where the light illuminates the surface the returned color is blue (which is at the right-hand end of the gradient) and where it does not, the color is red. For degrees of brightness between 0 and 1, a color from the middle of the gradient is used.



E4. Shaders

You can use the 'Shader' link field to combine the light brightness value with a shader. For example, with the same cube and material from the previous example, here a Noise shader has been used in the 'Shader' field and 'Invert' is checked:



Any other shader can be used including bitmaps. On my website you can see an [example animation](#) using an X-Particles 'Sample Shader'.

Known limitations

There are some limitations on this shader.

1. You will only see the result when rendered, not in the viewport. This is because the shader needs data which is only available at render time.
2. It is not possible to add a Layer shader to the 'Shader' link field. Unfortunately the Layer shader cannot be sampled by a plugin in the way that virtually every other shader can, so it simply won't work. It's possible to use a Fusion shader instead.

3. Be aware that the shader returns a value for every surface point, so if you use it in the color channel of a material the actual material color is ignored – the shader takes over completely.

4. The plugin is compatible with Cinema 4D R2024. Versions are also available for C4D R23/S24, R25/S26, and R2023. There is no version for C4D versions before R23.

Steve Pedler
October 2024