



Flat Lighting

by Bogdan Gochev

Table of Contents

[Flat Lighting](#)

[Introduction](#)

[Usage](#)

[Flat Lighting](#)

[Lighting](#)

[Global Gradient](#)

[Lightmapping](#)

[Light Sources](#)

[Custom Shadows](#)

[Bake Shaders](#)

[Flat Lighting Transparent](#)

[Flat Lighting Surface](#)

[Flat Lighting Cell/Toon Surface](#)

[Flat Lighting Sources](#)

[Spot Light](#)

[Point Light](#)

[Direct Light](#)

[Shadow Projector](#)

Introduction

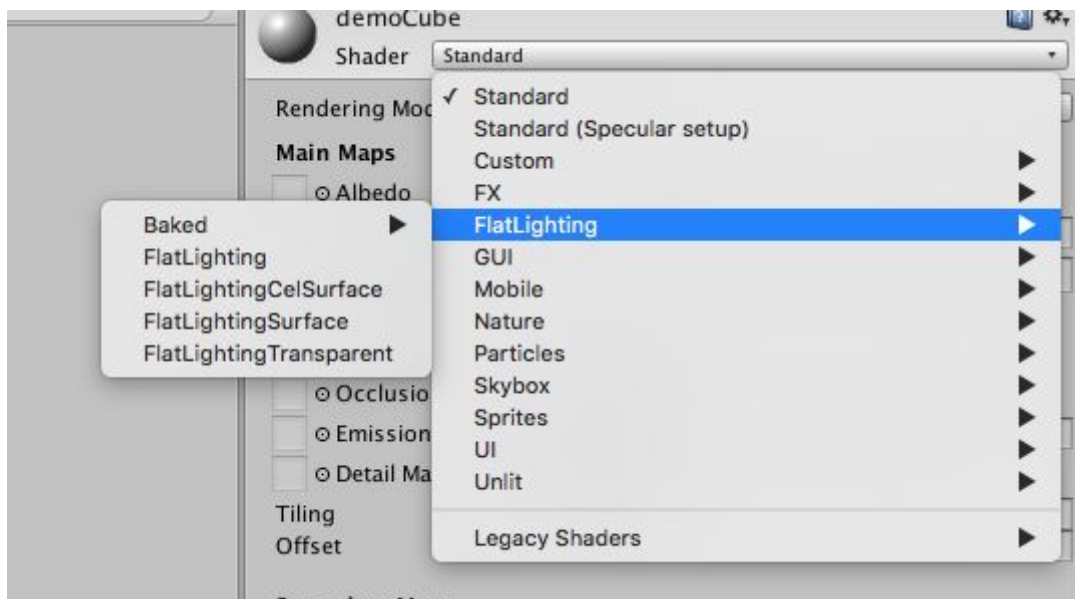
Flat Lighting is a highly optimized set of shaders and custom light sources that lets you create stunning visual effects similar to games like Monument Valley.

There are a variety of demo scenes and light setups, from which you could learn. I was using the scene “Material Test Scene” as a playground to test materials with different settings. It is a complex scene, where all the important settings are used.

When changing one color in the Flat Lighting shader, the scene seems to break up completely. This is because choosing the right colors is the most important thing in making the scene look good.

Usage

The Flat Lighting system consists of 4 main shaders:



There is also a set of custom light sources.

Flat Lighting

This is the main shader that can receive flat lights and custom shadows.
The shader is divided into segments

demoCube

Shader FlatLighting/FlatLighting

▼ Lighting

Use Vertex Colors

ON OFF

Use Symmetrical Colors

ON OFF

Space

World

Light +X

Light -X

Light +Y

Light -Y

Light +Z

Light -Z

Main Texture

None (Texture)

Tiling

X 1

Y 1

Offset

X 0

Y 0

Select

▼ Gradient

Use Gradient

ON OFF

▼ Lightmapping

Use Lightmapping

Custom Unity OFF

▼ Light Sources

Ambient Light

ON OFF

Directional Light

ON OFF

Spot Light

ON OFF

Point Light

ON OFF

▼ Custom Shadows

Receive

ON OFF

Cast

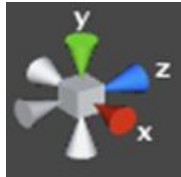
ON OFF

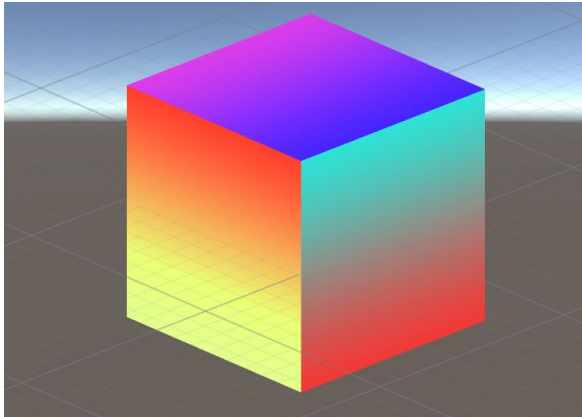
Bake Shader

!

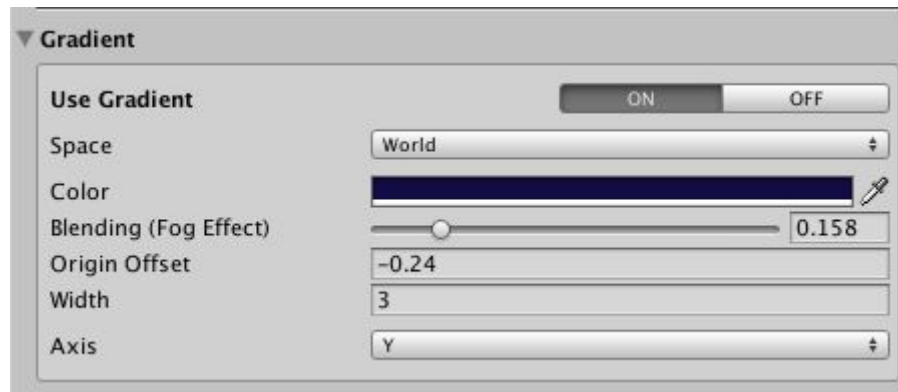
Creates a new shader that only includes the used options. Use this when you have problems compiling the big shader on mobile.



Lighting



Use Vertex Colors	The activation of this enables the shader to paint the model with the vertex colors stored in each vertex.
Space	<p>Indicates in what space the color axis below is interpreted. The options are:</p> <ul style="list-style-type: none"> • Local/Model space – the coordinate system of the model; • World space – the coordinate space of the game world. <p>It is very easy to experiment with these colors when you rotate the model. With the local space selected, the colors are maintained when the object is rotated in the world and appear to be painted on it.</p>
Use Symmetrical Colors	<p>The models are painted in different colors depending on which Axis and which direction are facing. It can be helpful to look at the unity Axis gizmo for Axis orientation (<i>see the image below</i>).</p>  <p>There you have the positive Axis directions in color, and the negatives in grey. This setting lets you select if you want to work only with Axis or if you want to work with their directions as well.</p>
Use Gradient Axis	Instead of using flat colors for the Axis, use gradients between 2 colors. You can also specify the gradient origin offset and the

<div data-bbox="199 212 774 1019"> <div>▼ Lighting</div> <div> <div>Use Vertex Colors</div> <div>ON OFF</div> </div> <div> <div>Use Symmetrical Colors</div> <div>ON OFF</div> </div> <div> <div>Space</div> <div>Local</div> </div> <div> <div>Use Gradient Axis</div> <div>ON OFF</div> </div> <div> <div>Light +X</div> <div> <div></div> <div></div> <div>0</div> <div>0.75</div> </div> </div> <div> <div>Second Color</div> <div> <div></div> <div></div> <div>0</div> <div>0.75</div> </div> </div> <div> <div>Light +Y</div> <div> <div></div> <div></div> <div>0</div> <div>0.75</div> </div> </div> <div> <div>Second Color</div> <div> <div></div> <div></div> <div>0</div> <div>0.75</div> </div> </div> <div> <div>Light +Z</div> <div> <div></div> <div></div> <div>0</div> <div>0.75</div> </div> </div> <div> <div>Second Color</div> <div> <div></div> <div></div> <div>0</div> <div>0.75</div> </div> </div> <div> <div>Main Texture</div> <div> <div>Tiling</div> <div>X 1 Y 1</div> <div>Offset</div> <div>X 0 Y 0</div> <div>None (Texture)</div> <div>Select</div> </div> </div> </div>	<p>width of the blending zone.</p> 
<p>Light +/-X, +/-Y, +/-Z</p>	<p>Different colors for the different Axis and directions.</p>
<p>Main texture</p>	<p>A texture that is applied to the model using the UV channel 0.</p>

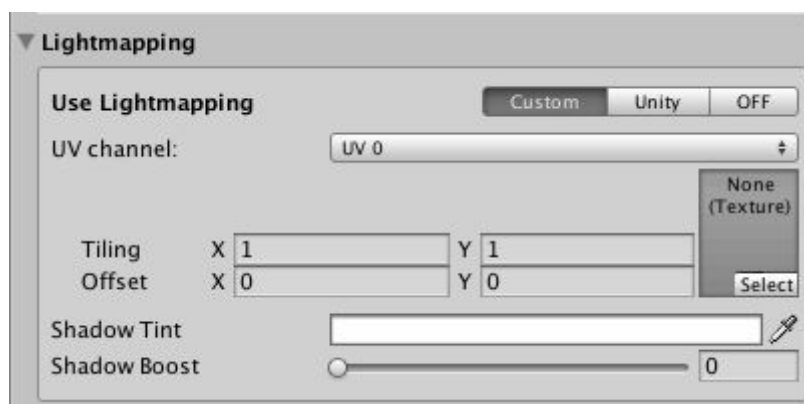
Global Gradient



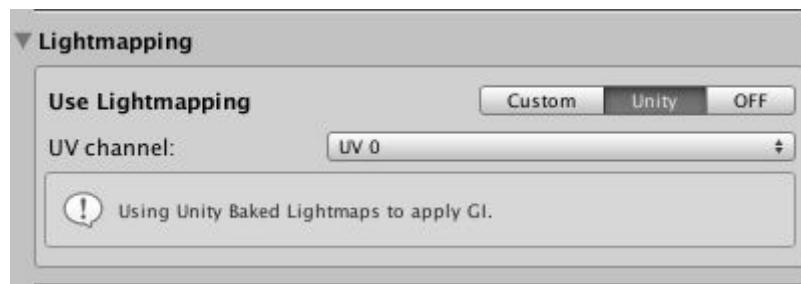
Space	Local/Model space or World space for Axis, Origin Offset and Width settings.
Color	The color of the gradient.
Blending (Fog Effect)	Indicates if the gradient is a solid color or is blended with the object color. If solid color is selected, a fog effect can be achieved.
Origin Offset	Offset from the origin of the object or the global origin of the world, which depends on the selected Space setting.
Width	The width of the transition zone between the gradient color and the object normal color.
Axis	<p>The axis around which the gradient will be evaluated:</p> <p>Y: </p> <p>X: </p>

	 <p>Z:</p>  <p>Free:</p> <p>The Axis sliders that appear, when Free mode is selected, act like a unit sphere coordinates (from -1.0 to +1.0).</p>
--	---

Lightmapping

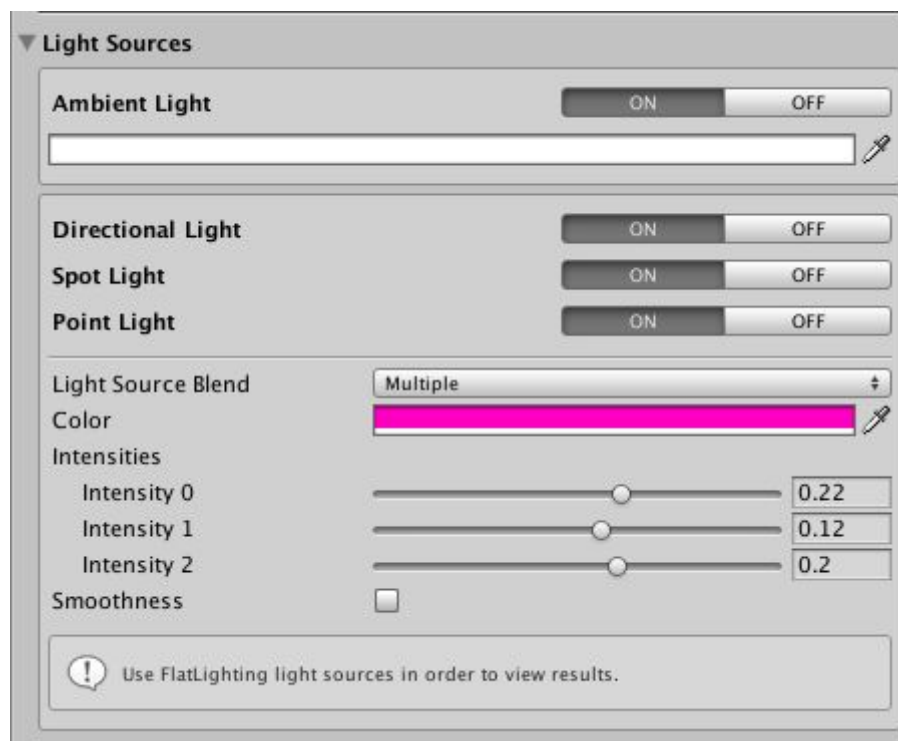


Custom	When you have baked the lightmap from outside of unity.
UV channel	The UV channel that will be used to apply the lightmap texture. There are 2 options: UV channel 0 or UV channel 1
Shadow Tint	The color which is applied to the lightmap texture.
Shadow Boost	Shadow intensity.



Unity	The lightmapping information is obtained from the Unity lighting system. You have to bake the lighting information in order to see any changes.
UV channel	If your model lightmapping UVs were generated by Unity at import time, please select UV channel as it's the default channel, which Unity uses to store lightmapping UVs.

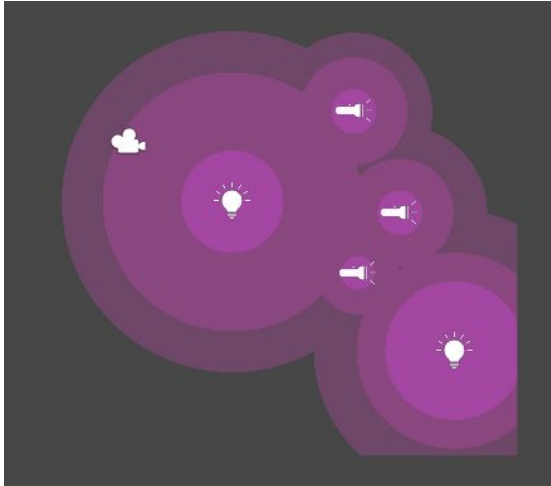
Light Sources



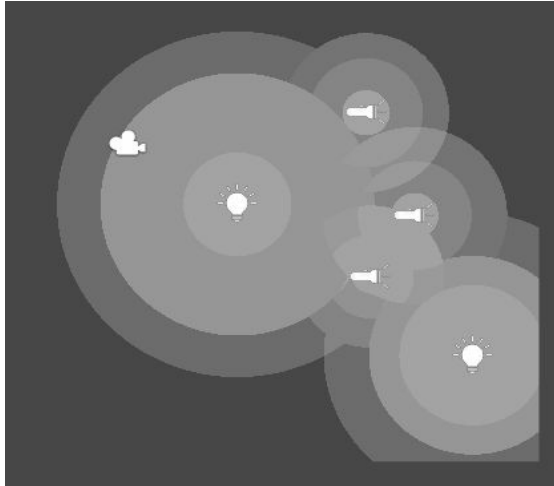
The ambient light is just a color added to the material in order to brighten it globally. The FlatLighting shaders have been optimized exceptionally for performance. Because of that if there is no need of using the Flat Light sources, they should be turned off. In order this material to be affected by the custom light sources, they must be activated.

Light Source Blend:

- Individual: Every light source uses its own light settings (Color, Intensity, Smoothness).
- Multiple: Every light source that affects the material will use the global light settings for Color, Intensity and Smoothness in order to achieve better blending between the lights. The distance and other light source dependent settings will still be used individually.



Multiple

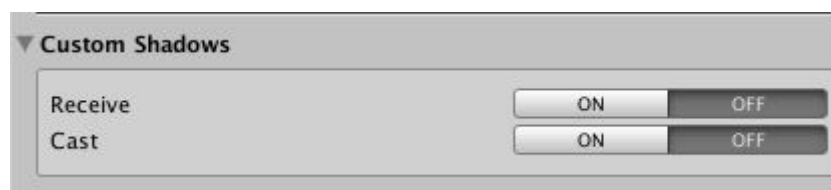


Individual

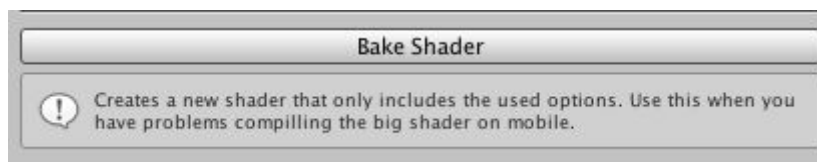
Custom Shadows

If the material receives custom shadows, the option “Receive” must be activated and a Flat Lighting Shadow Projector component must be enabled in the scene. More on how to use Flat Lighting Shadow Projector can be found in the relevant chapter.

In order to cast Custom Shadows on other objects, it is enough just to activate “Cast”.



Bake Shaders



Most of the Flat Lighting shader files contain a lot of variants. The usage of these variants provides the opportunity to have many shaders and effects in one file (instead of creating hundreds of shader files). Turning the on/off optional parameters automatically switches shaders inside the material editor without the user even noticing that.

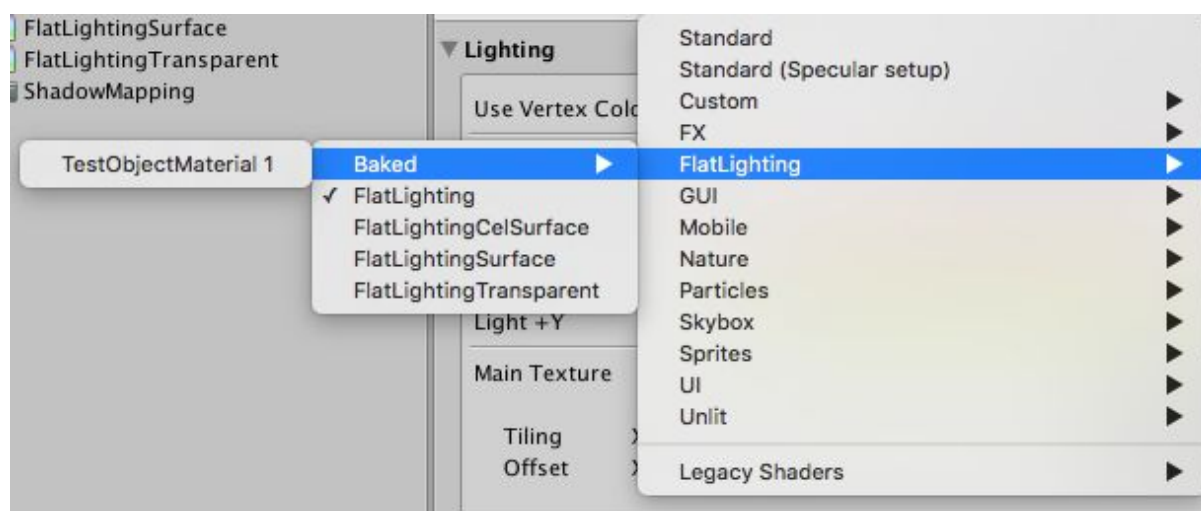
Disadvantage of these variants is that they increase shader file size (that goes into build) and compilation time. Some mobile devices may even crash because of the huge memory required for shader compilation.

For example, FlatLighting shader includes 4608 variants!

Baking removes all variants from shader and generates one shader file with all used optional parameters as one effect.

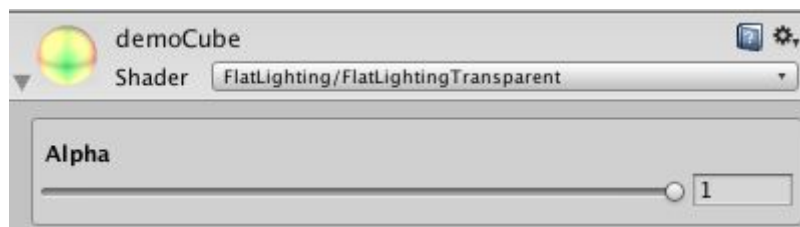
The "Bake Shader" can be found at the end of a material editor, in the inspector. After baking it is not possible to turn on/off optional parameters (they will be greyed out), but it will still be possible to change config settings of the parameters turned on. This new shader can be used by other materials and objects and acts as a normal shader, just with parameters on/off "hardcoded".

The baked shaders will be stored into "FlatLighting/Shaders/Lighting/Baked" and will be available to use by other materials in the normal shader menu.



Note: The shader baking just removes variants and reduces file size and shader compilation times, without having impact on their performance.

Flat Lighting Transparent



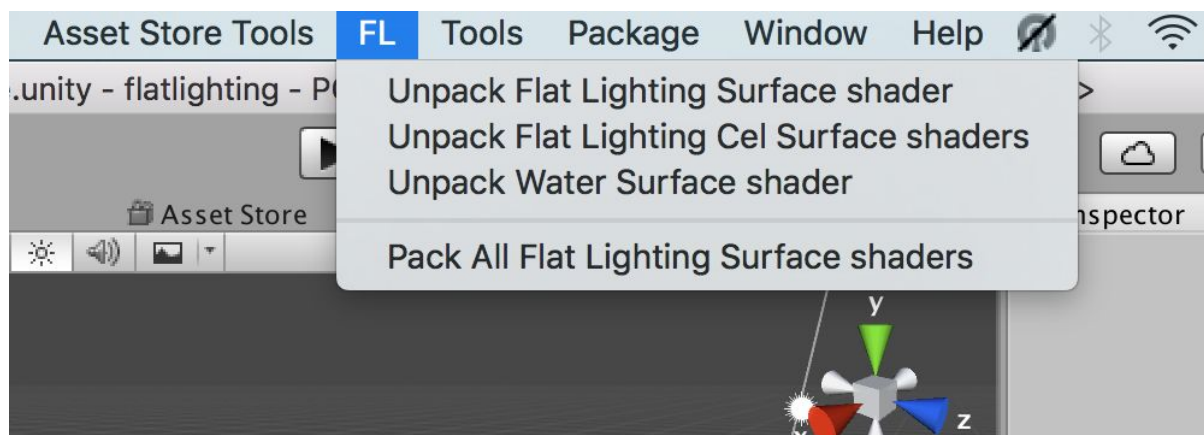
This shader has almost the same settings as the FlatLighting but in addition it has Alpha settings, which control the transparency of the material.

Flat Lighting Surface

This shader is affected by the standard Unity lights, lightmapping and all other light interacting Unity effects. That is because it is a surface version of the Flat Lighting shader. That is also why it has only Lighting and Gradient settings.

Because this family of shaders takes a long time to be compiled by the current Unity compiler, they are packed so when the asset is imported into a new project it doesn't take long time.

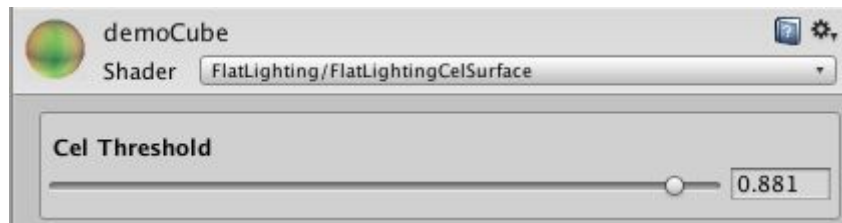
In order to use the Surface shaders first you have to **unpack** them.



After that the Unity shader compiler will start compiling the newly unpacked shaders. It could take up to 30 mins for the compilation to complete so, please, be patient. After the unpacking and compiling had finished you can start using the Surface family of shaders normally from the Flat Lighting Shaders menu.

Note: there are a couple of baked surface shaders included, but they are just to demonstrate the shaders capabilities and if you want to use the full potential of them you will have to unpack them.

Flat Lighting Cell/Toon Surface

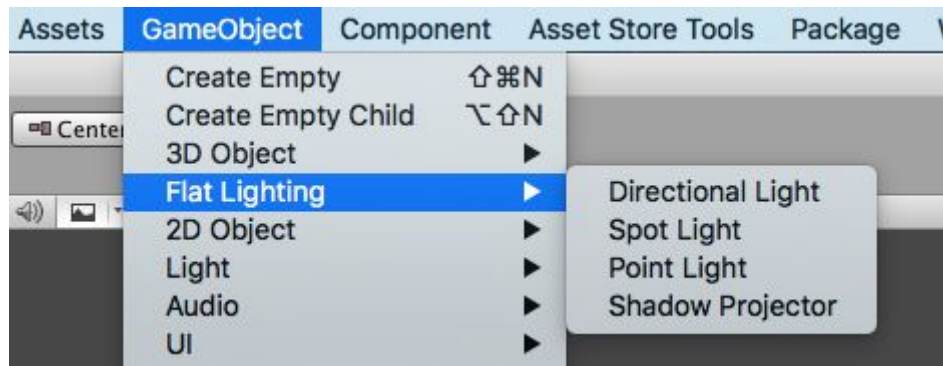


This shader is similar to the Flat Lighting Surface, and is a surface shader, too. The difference is the Cel Threshold, which simulate a toon/cel shading with the normal Unity lights. The goal is to achieve the same flat lighting effect that we get from using Flat Lighting, but instead of using the custom light sources and shadow mapping, we use Unity's one.

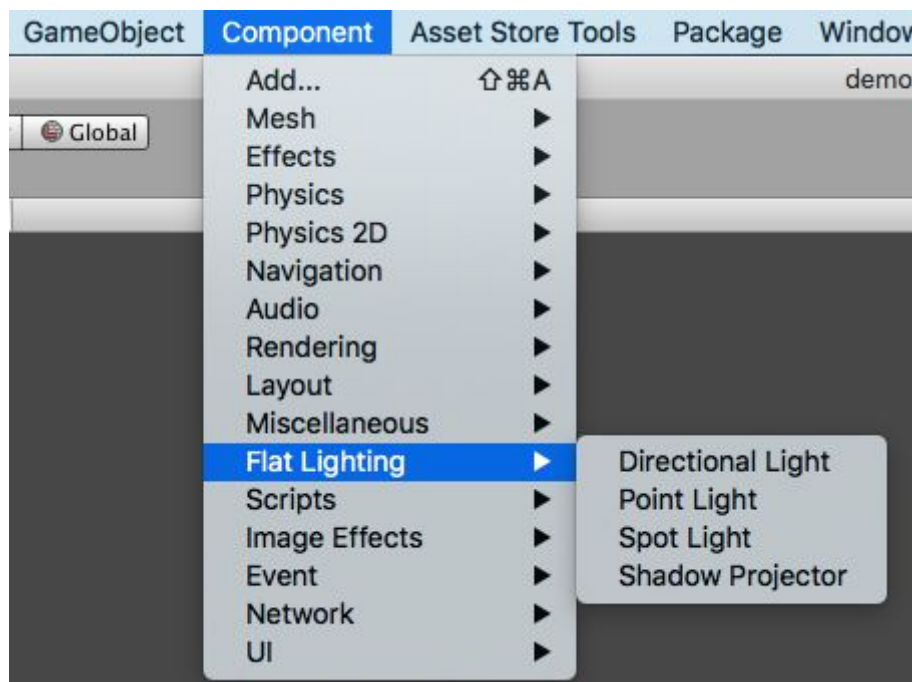
Flat Lighting Sources

There are 4 types of light sources: Spot, Point, Direct and Global Direct. Also, there is another type of source, although it is not a light source – it is a shadow mapping source called Shadow Projector.

You can add Flat Lighting sources from the Game Object menu.

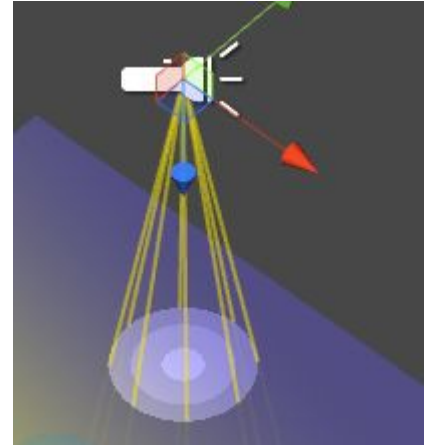
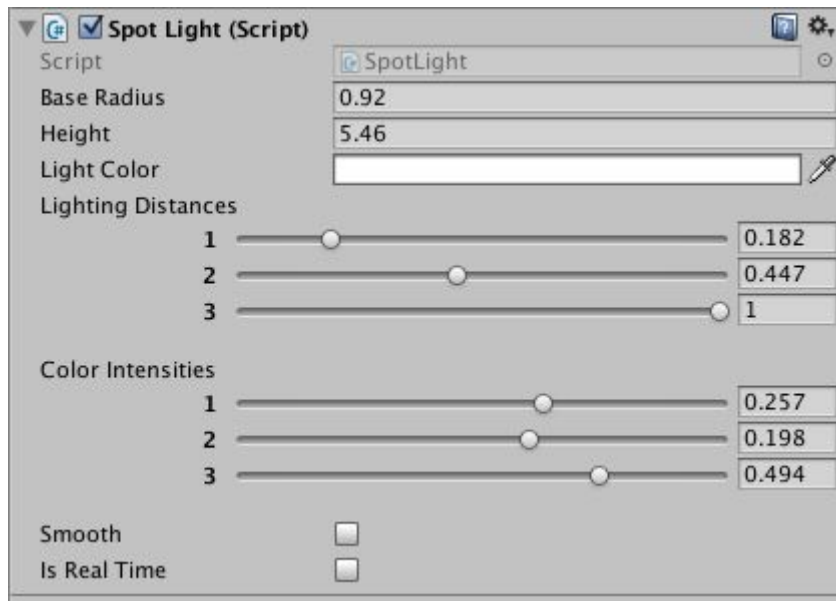


Or add them as components to existing GameObjects:



Note: It is important to include "Hidden/ShadowDepth" shader in the Always Included Shaders in Unity's Graphics build settings in order to have the flat light sources working when the build is exported.

Spot Light

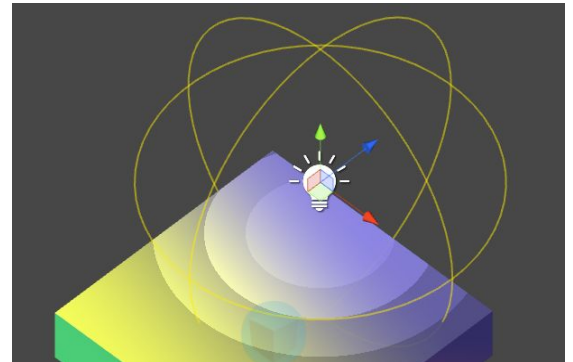
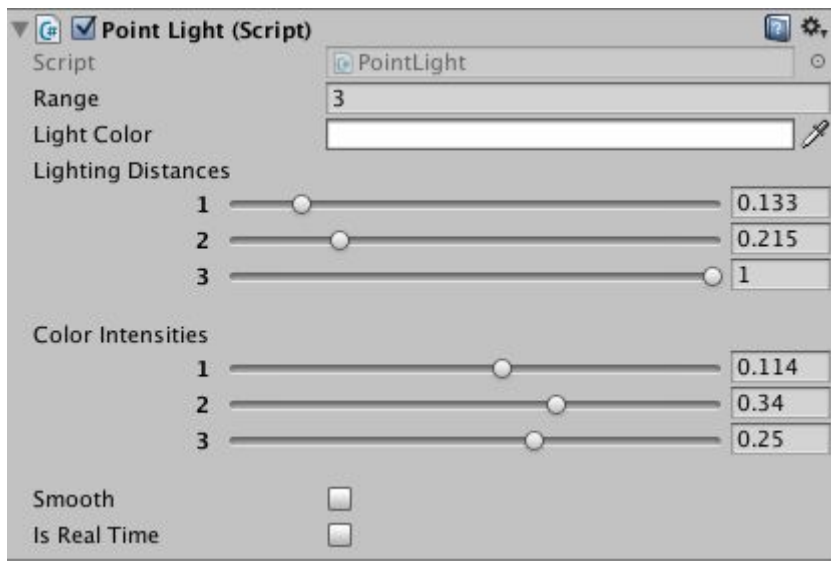


This component can be attached to any game object and acts as a spot light. You can have as much as 25 spot lights simultaneously enable per scene.

Base Radius	The radius of the spot light.
Height	The distance that the spot light illuminates.
Light Color	The color of the light.
Lighting Distances	These 3 components affect the size of the 3 light rings.
Color Intensities	These are the intensity/color power of the different light rings.
Smooth	Indicated if the the light is sharp or soft.
Real Time	Indicates if the light source is dynamic or static.

Note: It is important to include "Hidden/ShadowDepth" shader in the Always Included Shaders in Unity's Graphics build settings in order to have the flat light sources working when the build is exported.

Point Light

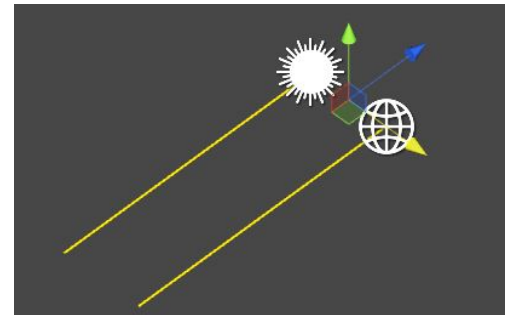
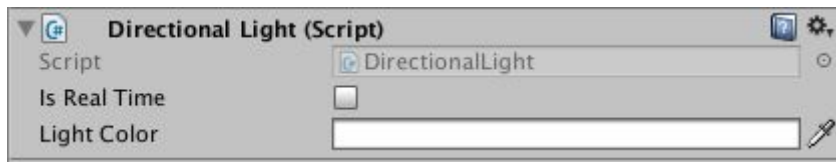


This component can be attached to any game object and acts as a point light. You can have as much as 25 point lights simultaneously enable per scene.

Range	The scaling/distance that the light affects.
Light Color	The color of the light.
Lighting Distances	These 3 components affect the size of the 3 light rings.
Color Intensities	These are the intensity/color power of the different light rings.
Smooth	Indicated if the the light is sharp or soft.
Real Time	Indicates if the light source is dynamic or static.

Note: It is important to include "Hidden/ShadowDepth" shader in the Always Included Shaders in Unity's Graphics build settings in order to have the flat light sources working when the build is exported.

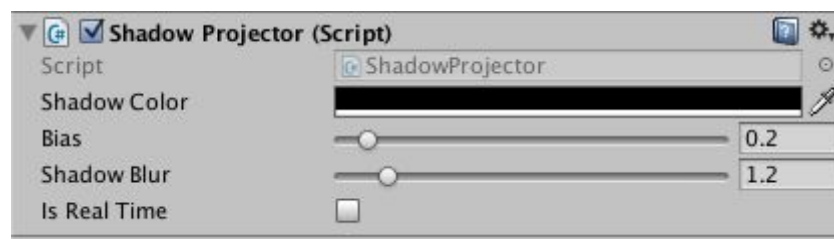
Direct Light



When this light are selected in the editor, the yellow ray shows the direction of the light. You can have as much as 5 direct lights simultaneously enable per scene.

Note: It is important to include "Hidden/ShadowDepth" shader in the Always Included Shaders in Unity's Graphics build settings in order to have the flat light sources working when the build is exported.

Shadow Projector



These components require a Camera – if there is not a Camera attached to the Game Object, it automatically attaches one. The camera is used for shadow mapping, so it will not interfere with your main camera.

You can have as much as 1 shadow projector simultaneously enable per scene.

Shadow Color	The color of the shadows.
Bias	How much the shadows should overlap with the object that it is casting them.
Shadow Blur	Shadow blur/softness.
Real Time	Indicates if there is a shadow casted by these projectors, dynamic or static.