standardPlus

Version 1.05

User Guide

Table of Contents

1 – Version changes and Bug fixes

Version 1.05

Bug fixed - When trying to create baked lights to illuminate dynamic objects via light probes, the internal Unity lightmap atlas UV coordinates were affecting Standard Plus UV coordinates, thus changing the position of the custom lightmaps in Play Mode and Builds.

IMPORTANT: To use Unity lights set to baked or mixed, please drag and drop the script **EnableBakedLights.cs** on to every object that uses Standard Plus shader. This is only needed when using Unity's lights set to baked or mixed, since otherwise the lightmaps will be positioned incorrectly in Play mode and Builds.

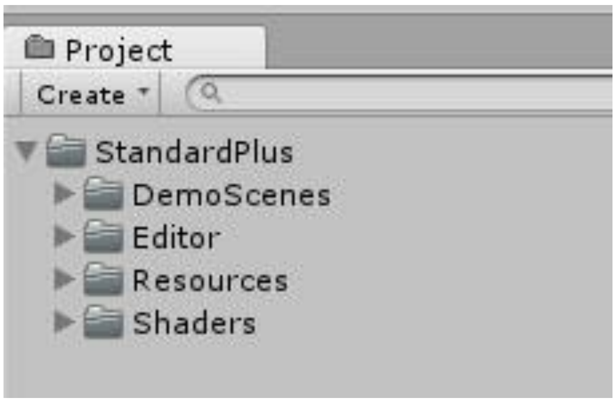
After this, in Edit mode, if the lightmaps appear out of place, just click Play Mode. When back to Edit mode, they will be fixed.

2 – Importing Standard Plus

To start using Standard Plus, download the package from the Asset Store and import it into your project.

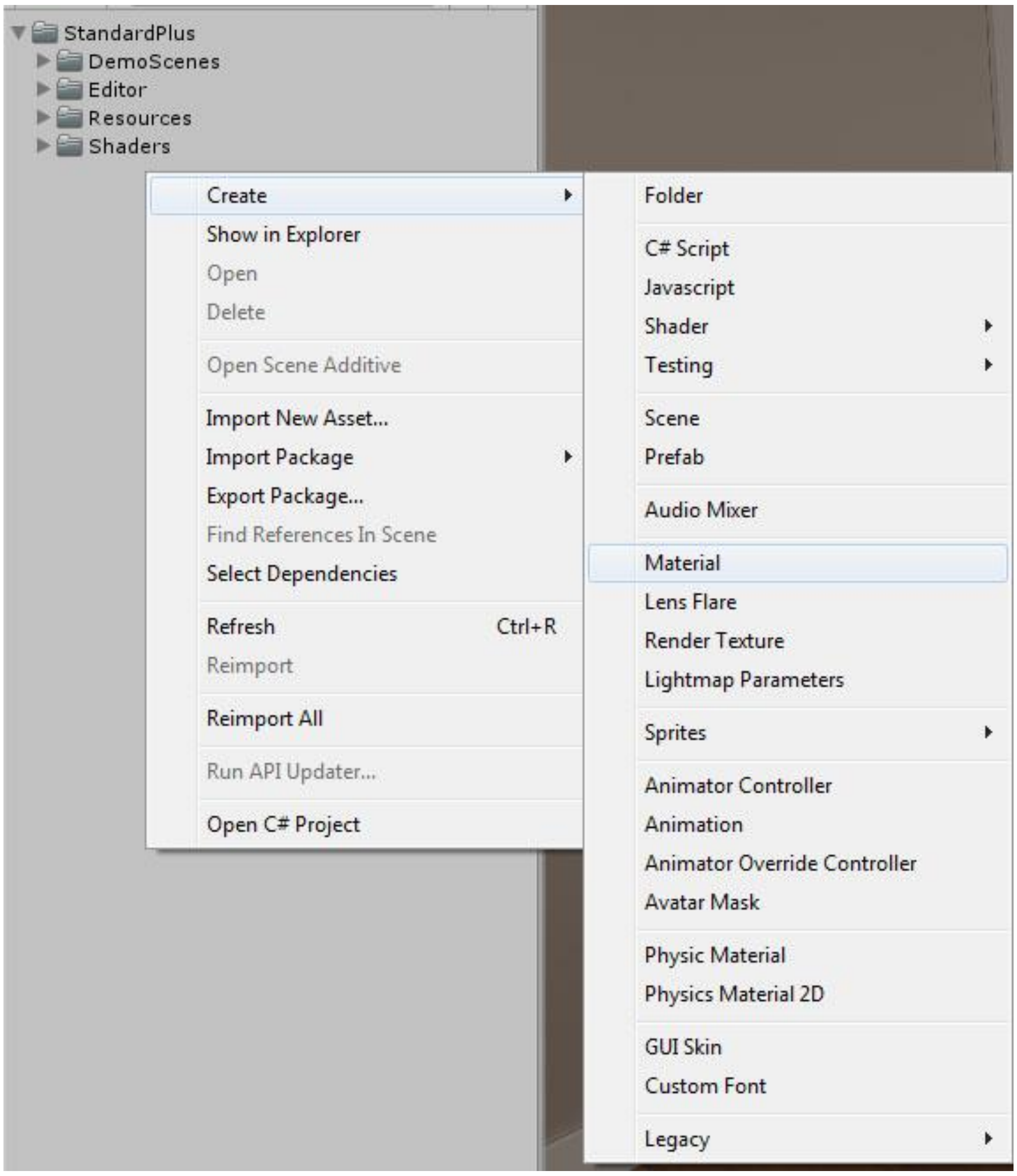
FOTO

Your project now will contain a new folder called Standard Plus, where the shaders, as well as the Demo Scenes, are included.

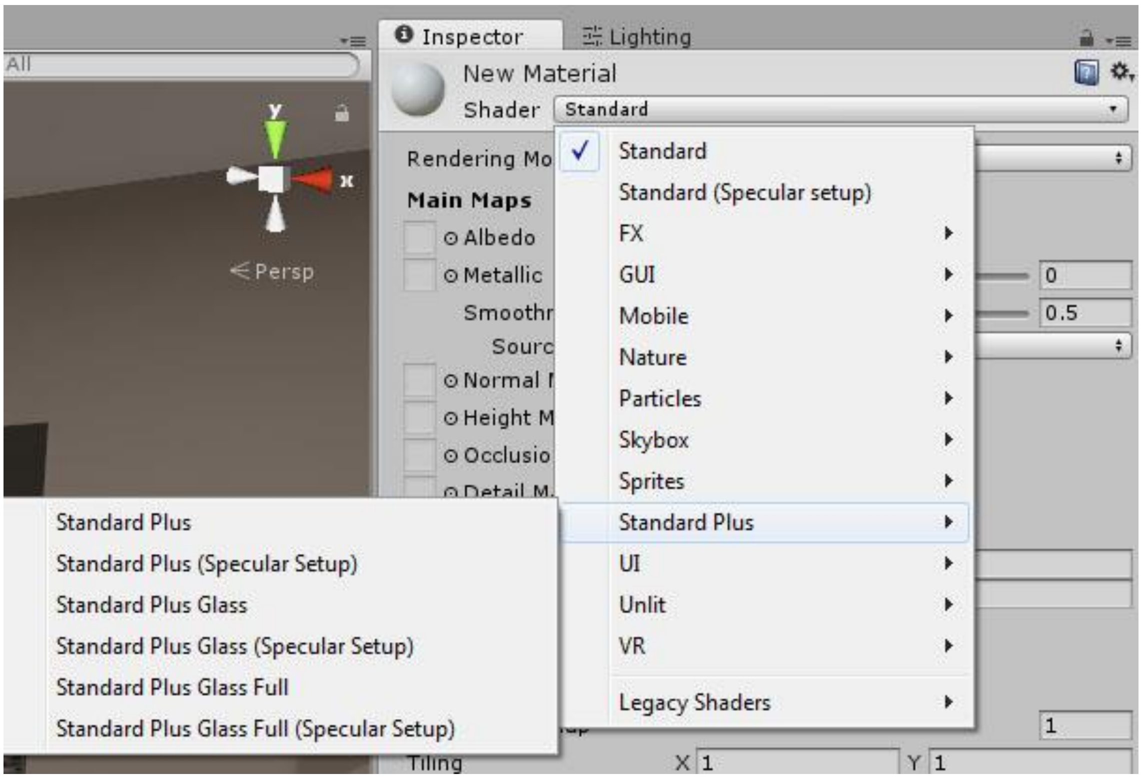


3 – Common Setup

Create a material like you would normally do, via right click on the Project panel, or clicking on the top menu. Create > Material.



The material is created with Unity's default Standard Shader. Click on the Shader drop down list. You will see a new item, Standard Plus. Inside of it there are the following shaders:



Standard Plus - Choose it to use multiple lightmaps and/ or Translucency, using the Metallic workflow.

Standard Plus Specular Setup - The same as above, but using the Specular workflow.

Standard Plus Glass - Choose it to use the optimized glass. Metallic workflow.

Standard Plus Glass Specular Setup – Same as above, but using Specular workflow

Standard Plus Glass Full – Choose this to use glass. This shader is more expensive than the simple Glass. It lets you see other glass objects behind your object. Metallic workflow.

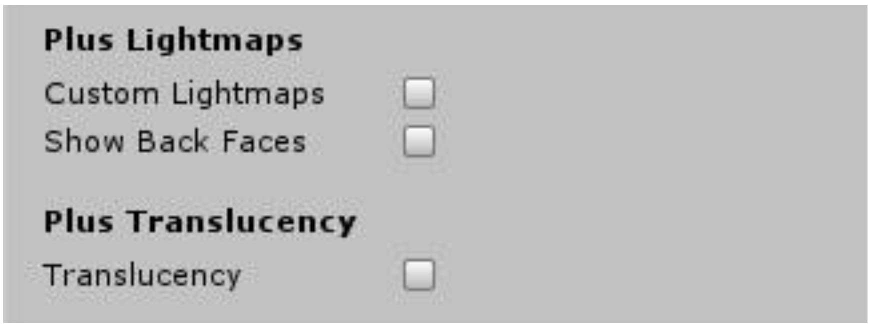
Standard Plus Glass Full Specular Setup – Same as above but using the Specular workflow.

Choose the appropriate shader for your material.

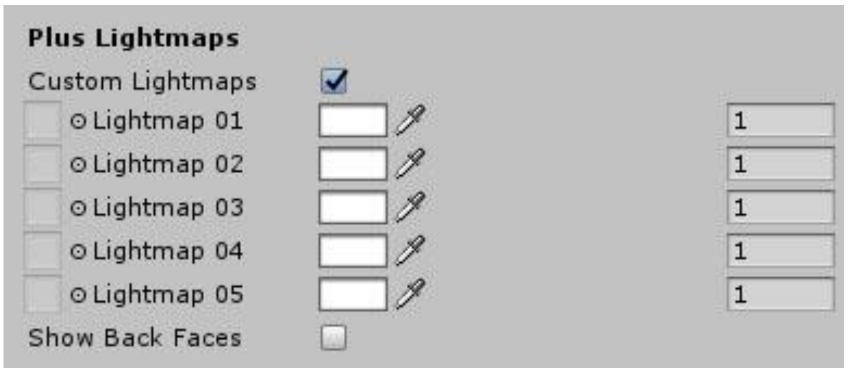
4 – Standard Plus Lightmaps

After choosing the Standard Plus or Standard Plus Specular Setup shader, you will be able to use multiple lightmaps.

Below the parameters common to Unity’s built in Standard shader, there is a new set of options under **Plus Lightmaps**.



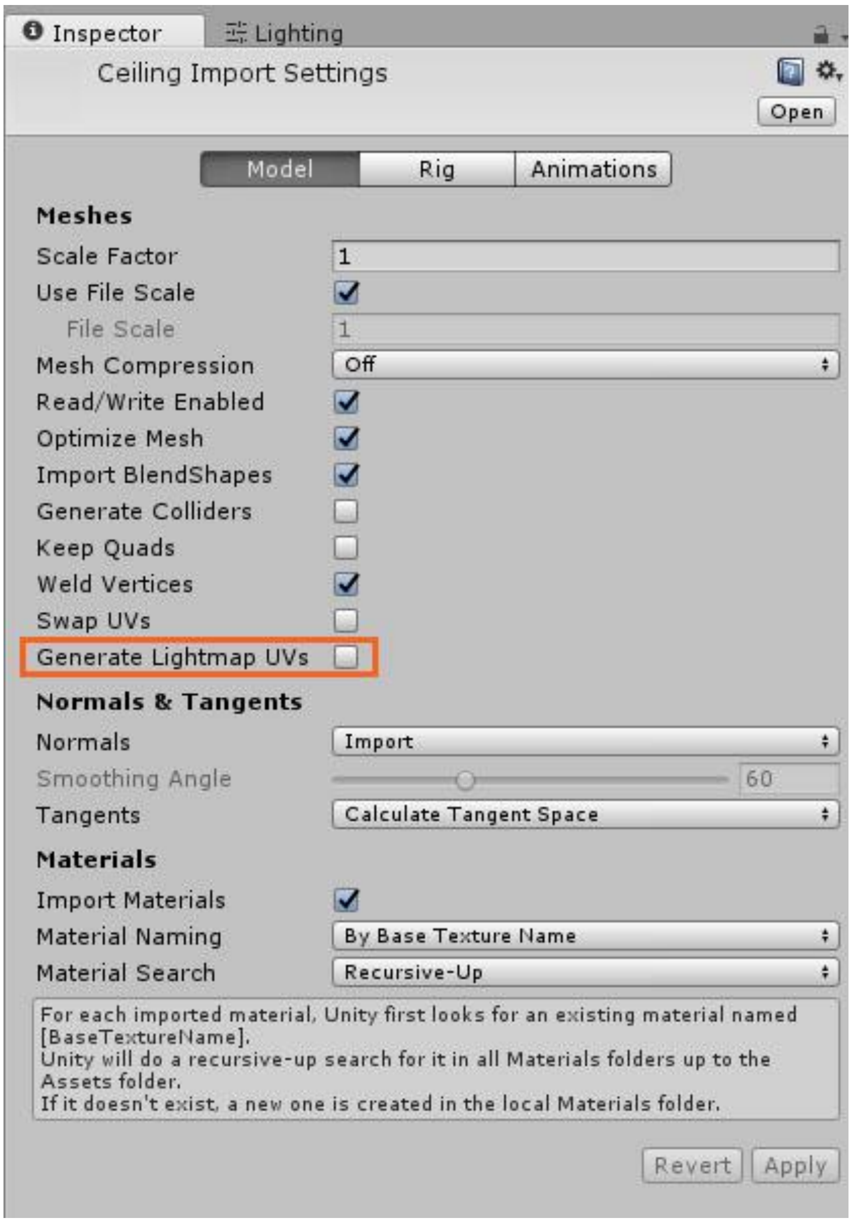
Clicking the box next to Custom Lightmaps, the lightmap slots and their controls will unfold.


Plus Lightmaps

Custom Lightmaps	✓	
Lightmap 01		1
Lightmap 02		1
Lightmap 03		1
Lightmap 04		1
Lightmap 05		1
Show Back Faces	☐	

You can assign up to 5 different lightmaps for each material, either HDR or LDR. Note that by default, the tinting colors are black. This is so you don't get confused by the various lightmaps you add.

It's important that your models are exported with a second UV channel for the lightmaps. In order to preserve your second UV channel when importing your custom lightmaped models, please make sure the "Generate Lightmap UVs" on the import settings window is **UNCHECKED.**



Ceiling Import Settings

Meshes

Scale Factor	1
Use File Scale	✓
File Scale	1
Mesh Compression	Off
Read/Write Enabled	✓
Optimize Mesh	✓
Import BlendShapes	✓
Generate Colliders	☐
Keep Quads	☐
Weld Vertices	✓
Swap UVs	☐
Generate Lightmap UVs	☐

Normals & Tangents

Normals	Import
Smoothing Angle	60
Tangents	Calculate Tangent Space

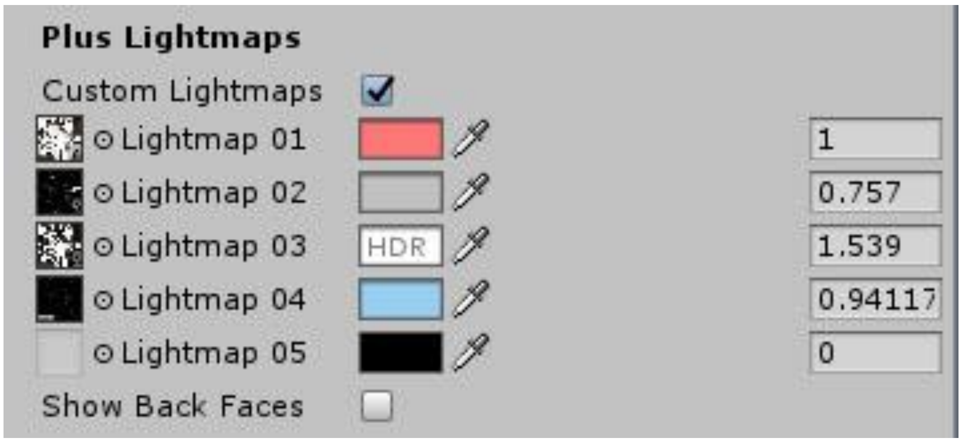
Materials

Import Materials	✓
Material Naming	By Base Texture Name
Material Search	Recursive-Up

For each imported material, Unity first looks for an existing material named [BaseTextureName].
Unity will do a recursive-up search for it in all Materials folders up to the Assets folder.
If it doesn't exist, a new one is created in the local Materials folder.

At the right of the lightmap textures you will find a color picker and a number value. With these controls you can tint and change the exposure of your lightmaps. Notice that you can set your brightness value over 1, because these are

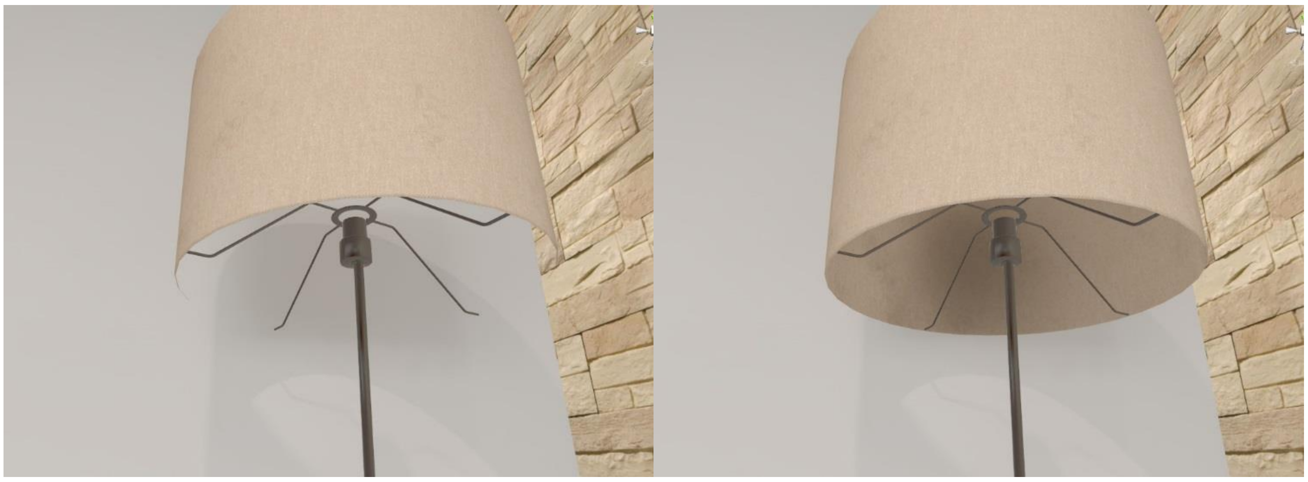
HDR controls. In fact, when you set the number greater than 1, the letters HDR will appear in the color box. This way you can realistically increase the lighting of your scene.


Plus Lightmaps

Custom Lightmaps	✓	
Lightmap 01	🟥	1
Lightmap 02	⬜	0.757
Lightmap 03	HDR	1.539
Lightmap 04	🟦	0.94117
Lightmap 05	⬛	0
Show Back Faces	☐	

Show Back Faces

By clicking the button Show Back Faces, Unity will also display the back of the faces. Unity doesn't show them by default, but the feature is useful when you model simple thin objects with no thickness.

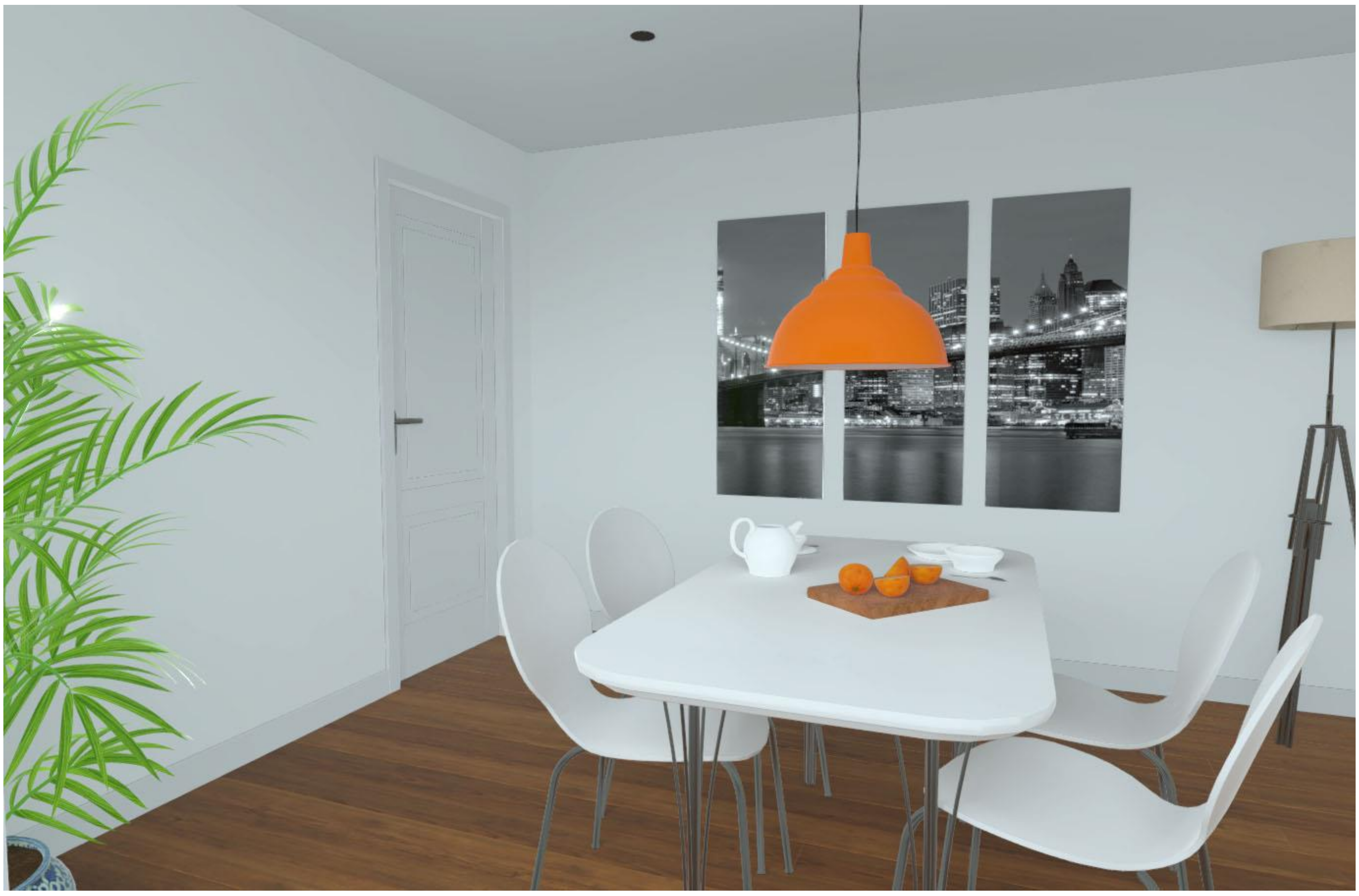


Show back faces OFF *Show Back Faces ON*

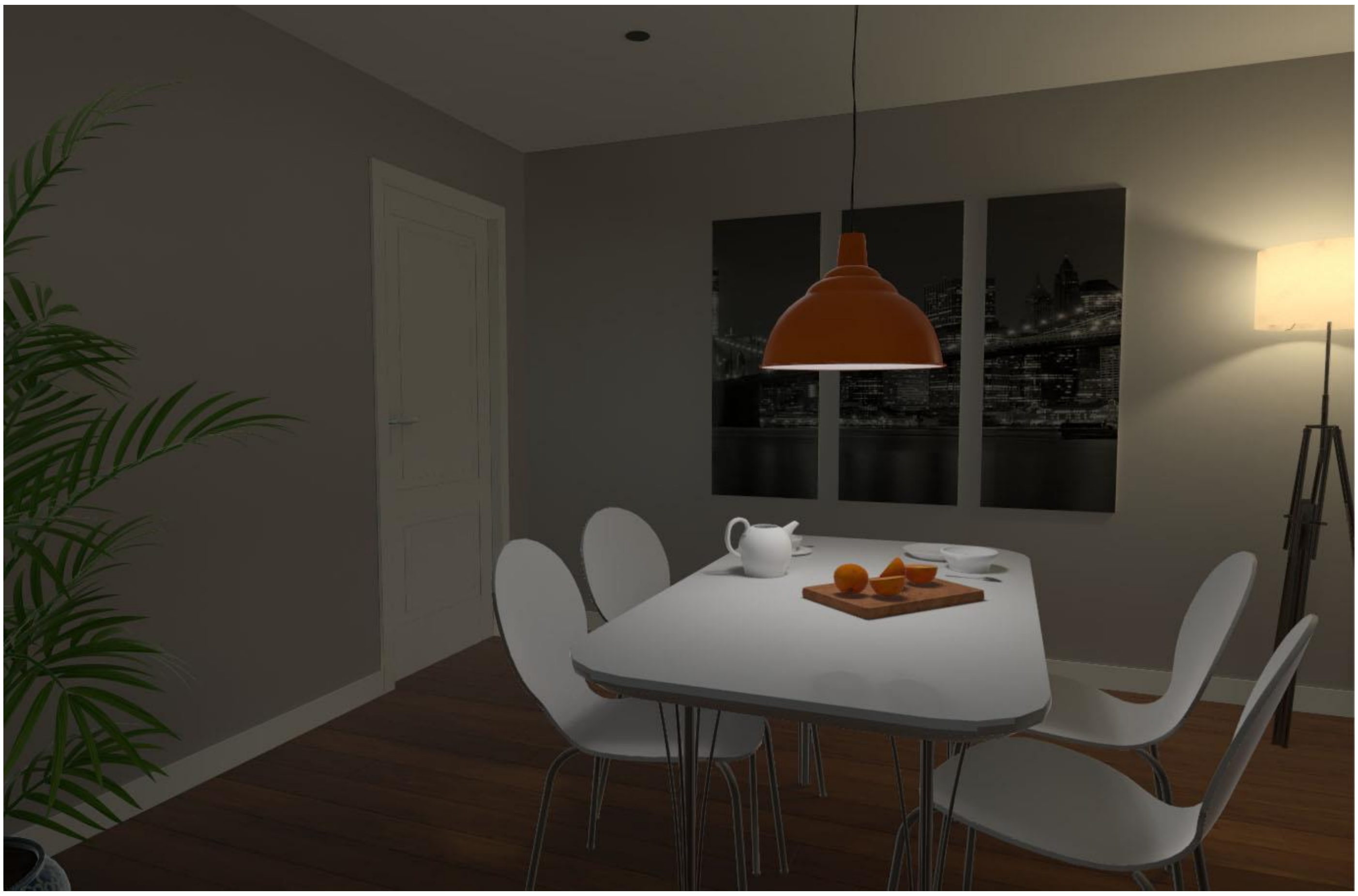
Why multiple lightmaps? Light mixing!

You can use just one lightmap for your materials, representing all the lighting in your scene. But let's say you have 4 lights in the scene, and want them to be controlled in real-time in game. The way to do that is to bake one lightmap per each light. Then import and place them in the available lightmap slots.

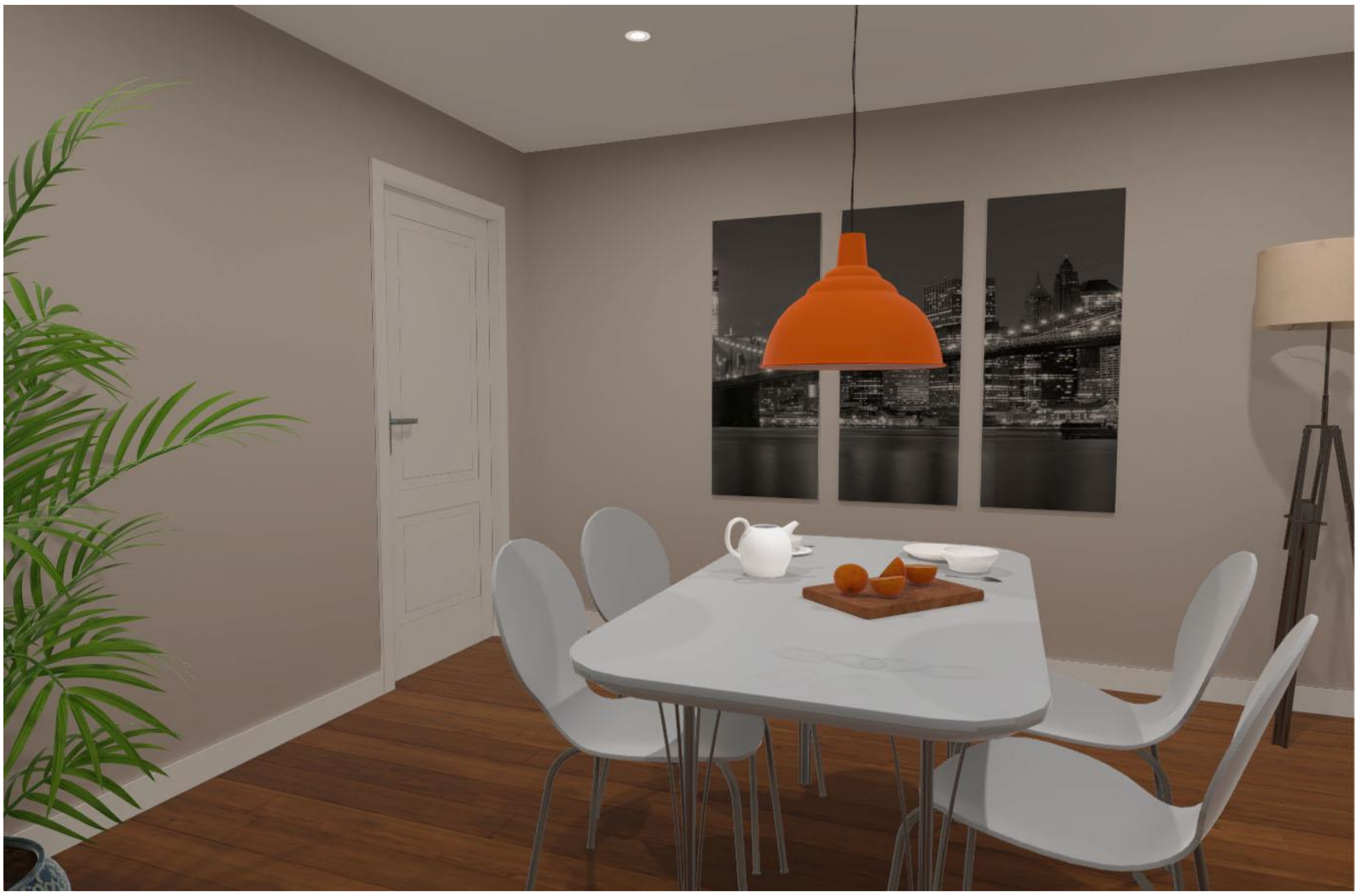
Now, by changing the color picker or the number value, you can turn each light off, on, or any value in between. So you essentially light mix your scene getting different moods or aiding your gameplay.



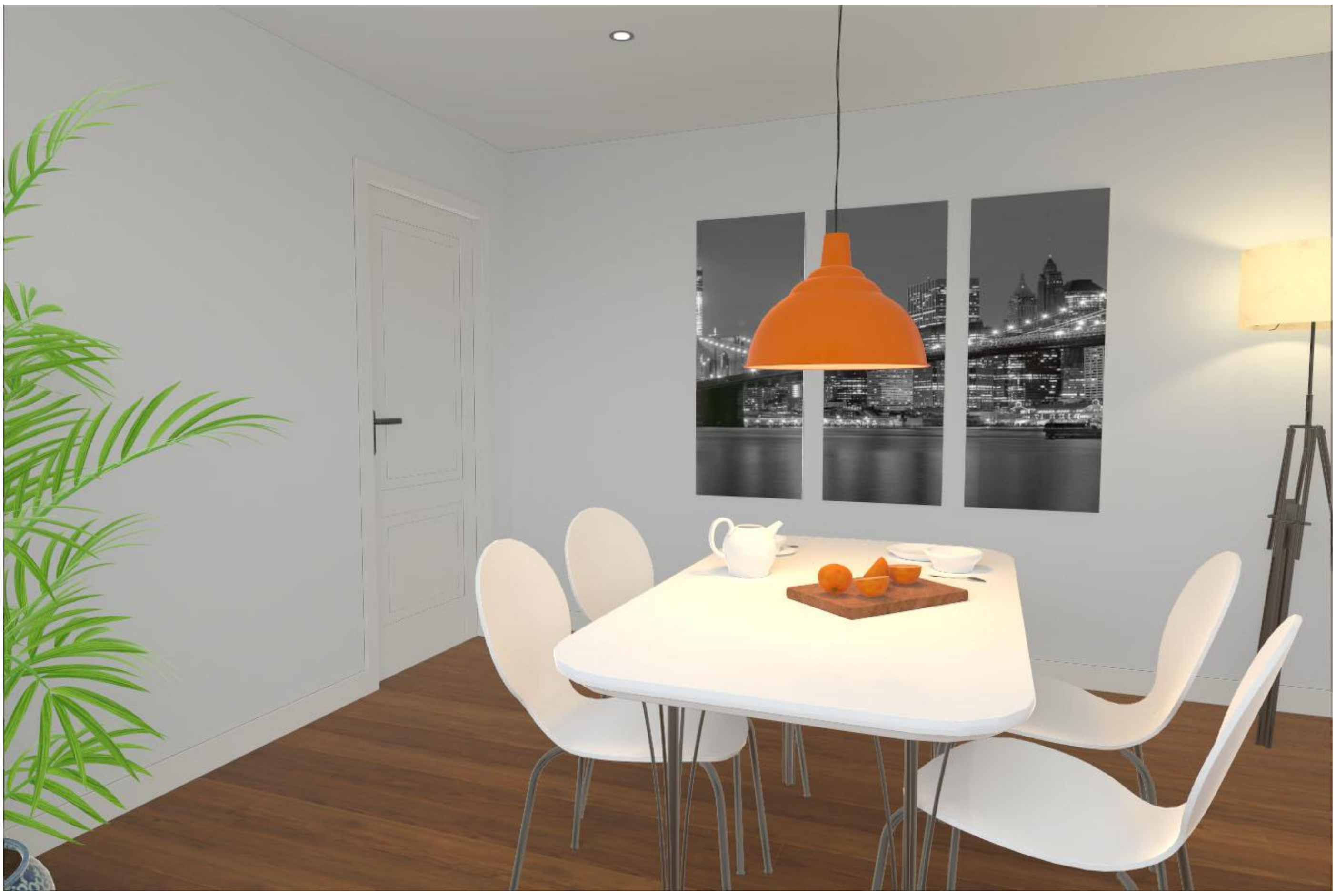
Scene with a lightmap of a soft sky light.



Sky lightmap turned to black, lightmaps from dining lamp and floor lamp turned on. Floor lamp slightly tinted to yellow.



Only the lightmap from the ceiling spots turned on.

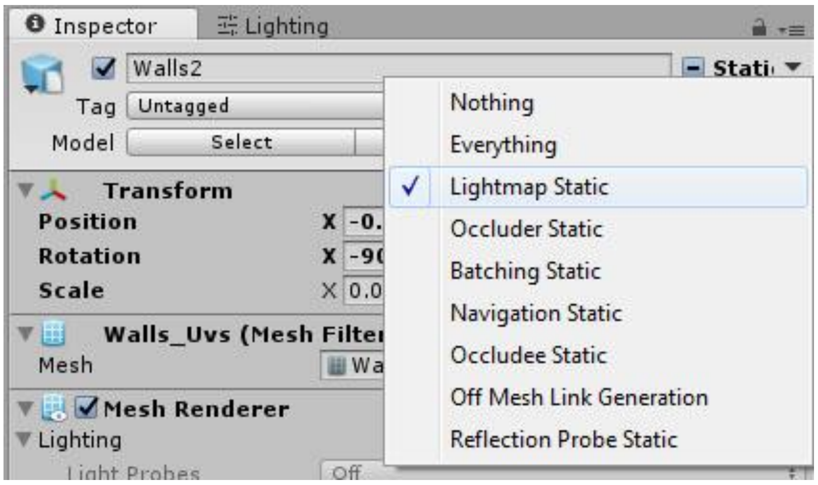


The 4 lightmaps mixed together with various intensities and tints.

Real-Time GI

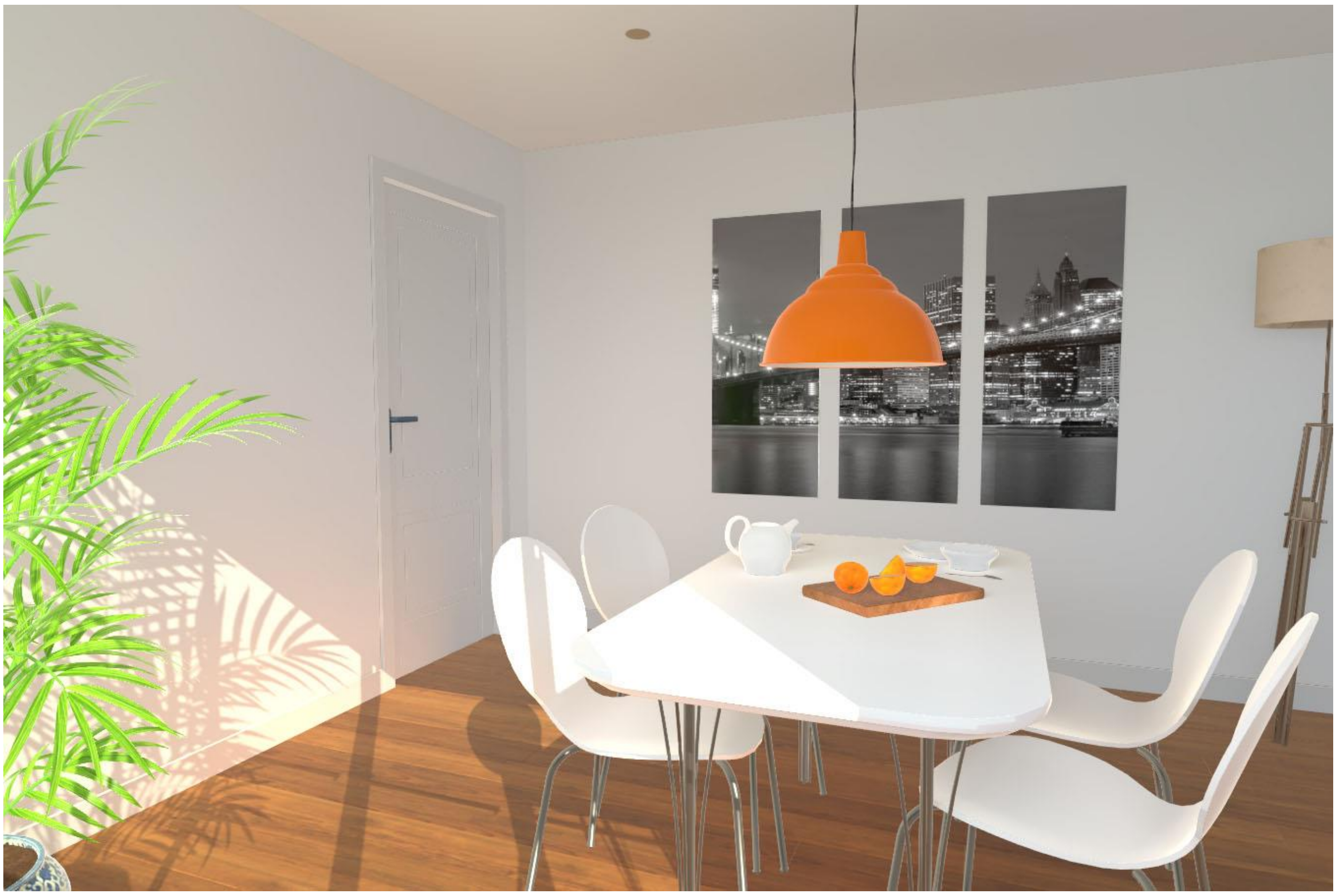
You are not bound to only use your custom lightmaps, you can take advantage of Unity's own realtime GI baking engine to make your scene more dynamic.

In order to use this feature, first, make sure your objects are set to **Lightmap static** on the Inspector panel. If you don't wish to use realtime GI, you don't need to mark them as static, however, now you have to do it, since you are using Unity's lighting baking system.



Then, enable **Realtime Global Illumination** on the **Lighting** panel and turn on **Auto Generate**, or leave it off and generate the dynamic lightmap on demand.

Now, just set the resolution however you like.



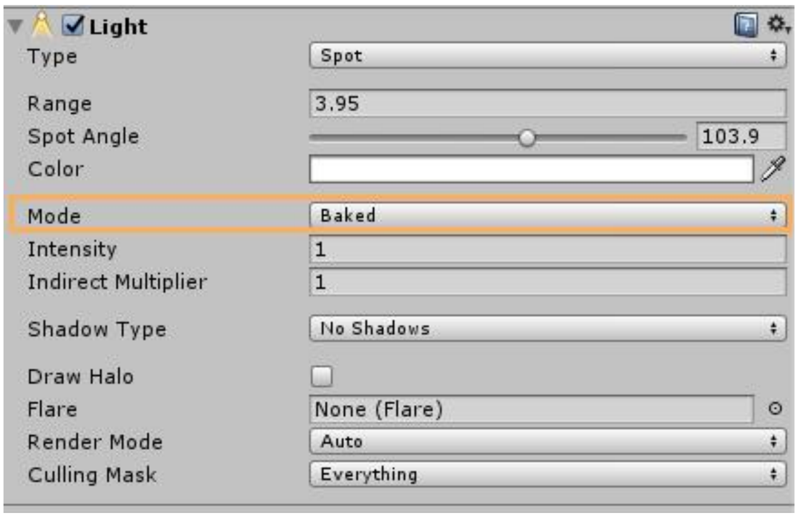
Custom sky lightmap mixed with realtime Unity directional light and realtime GI. You can see, near the bottom left side of the door and on the ceiling, the spill of yellowish sunlight from Unity's realtime baking engine.

Light Probes for dynamic objects

When you use your custom lightmaps, since they are basically just textures, Unity has no way to calculate the light probes to illuminate dynamic objects.

The way to fix this is to place Unity lights roughly at the same position you put your lights on the 3D package when you baked the scene.

Set the lights to baked. That way they will only be used to light the dynamic objects, not your custom lightmapped ones, and will not affect the realtime computations and increase frame rates.



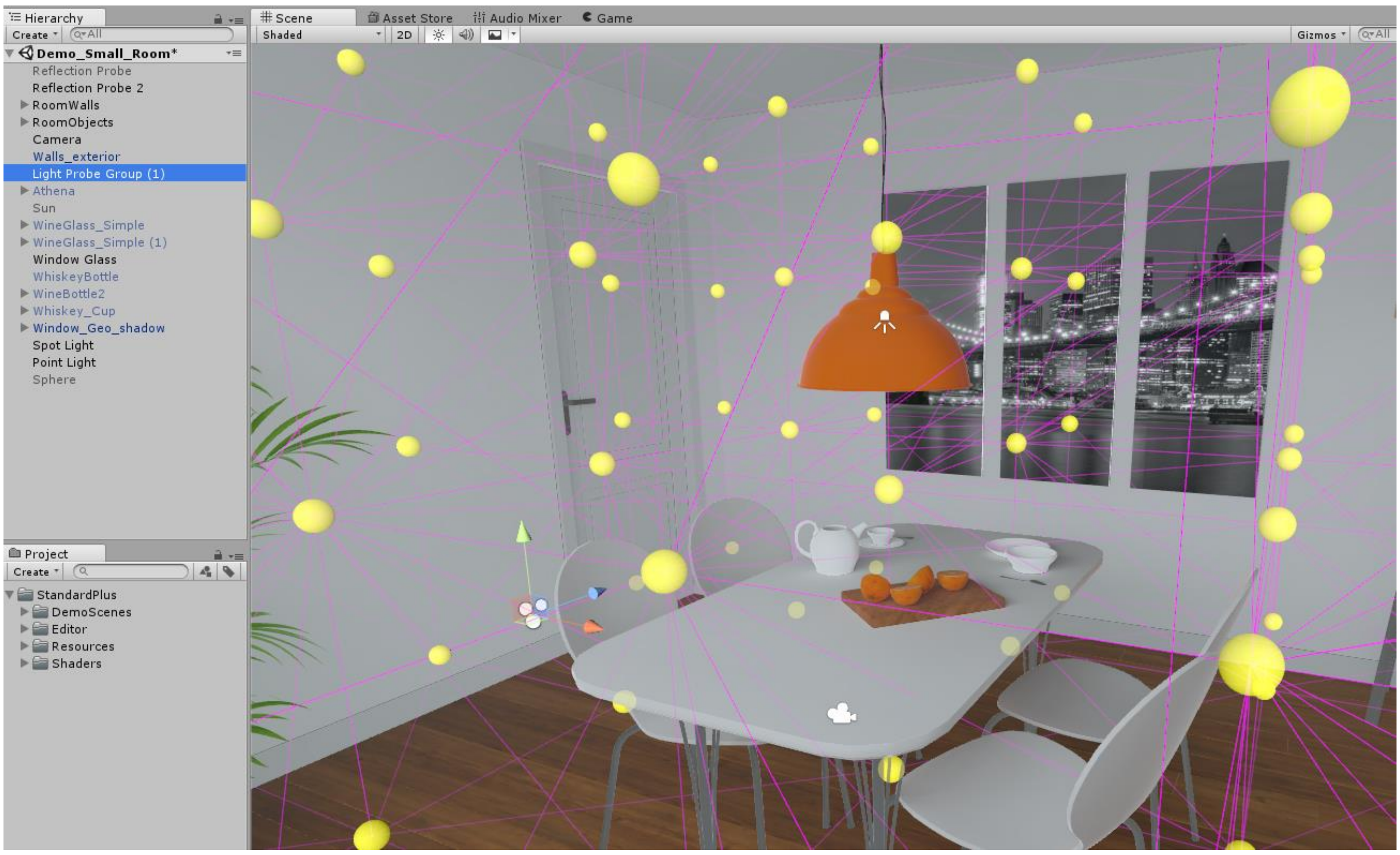
Light

Type	Spot
Range	3.95
Spot Angle	103.9
Color	
Mode	Baked
Intensity	1
Indirect Multiplier	1
Shadow Type	No Shadows
Draw Halo	☐
Flare	None (Flare)
Render Mode	Auto
Culling Mask	Everything

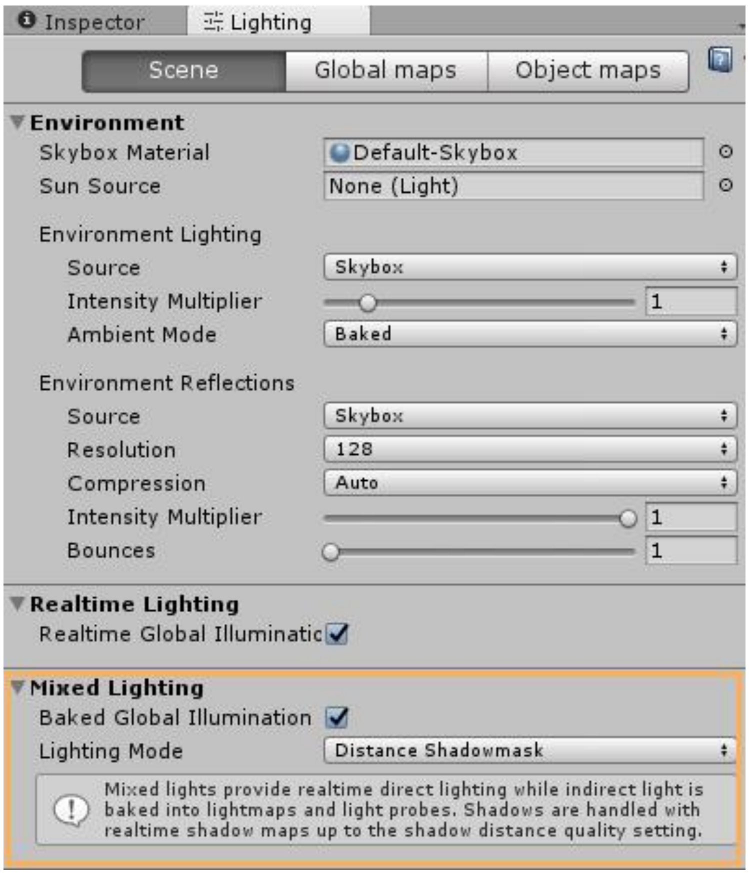
IMPORTANT: To use Unity lights set to baked or mixed, please drag and drop the script **EnableBakedLights.cs** on to every object that uses Standard Plus shader. This is only needed when using Unity's lights set to baked or mixed, since otherwise the lightmaps will be positioned incorrectly in Play mode and Builds.

After this, in Edit mode, if the lightmaps appear out of place, just click Play Mode. When back to Edit mode, they will be fixed.

Create your light probe group as you normally do.



After that, just turn on the **Mixed Lighting** on the Lighting panel.



Environment

Skybox Material: Default-Skybox
Sun Source: None (Light)

Environment Lighting
- Source: Skybox
- Intensity Multiplier: 1
- Ambient Mode: Baked

Environment Reflections
- Source: Skybox
- Resolution: 128
- Compression: Auto
- Intensity Multiplier: 1
- Bounces: 1

Realtime Lighting

Realtime Global Illuminatic ✓

Mixed Lighting

Baked Global Illumination ✓
Lighting Mode: Distance Shadowmask

Mixed lights provide realtime direct lighting while indirect light is baked into lightmaps and light probes. Shadows are handled with realtime shadow maps up to the shadow distance quality setting.

Unity will calculate this baked lighting and now the dynamic objects will receive it.

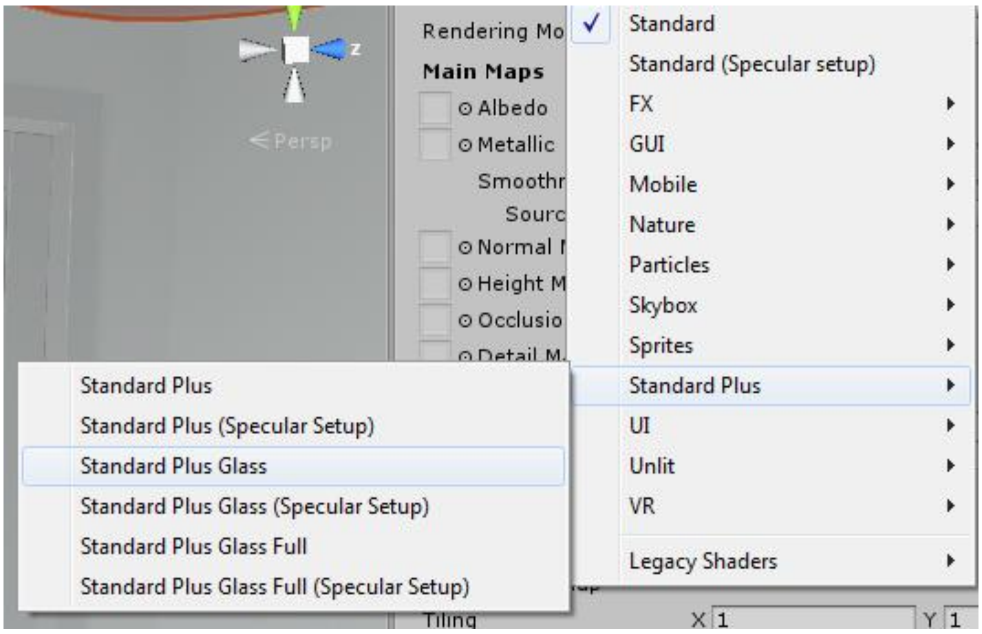
Sphere in the middle receiving baked lighting provided by the light probes.

Mobile limitation

For mobile, make sure you use non HDR lightmaps (LDR, such as PNGs, JPEGS, TIFF, etc), otherwise they will be converted automatically to LDR, causing unexpected appearance.

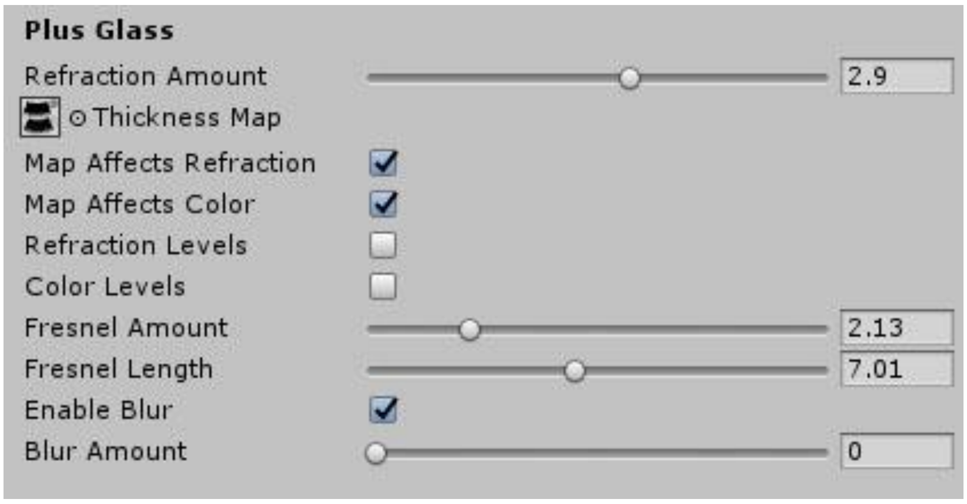
5 – Standard Plus Glass

To use the Glass features, choose the Standard Plus Glass shader.



By default, the shader is presented with the **Albedo Color alpha** on opaque (value 1). Turn it to 0, so the glass can be transparent. You can also choose any value between those two, so you can have a glass not totally transparent.

After the common parameters from Unity's Standard Shader, you will see a section called **Plus Glass.**


Plus Glass

Refraction Amount		2.9
Thickness Map		
Map Affects Refraction	✓	
Map Affects Color	✓	
Refraction Levels	☐	
Color Levels	☐	
Fresnel Amount		2.13
Fresnel Length		7.01
Enable Blur	✓	
Blur Amount		0

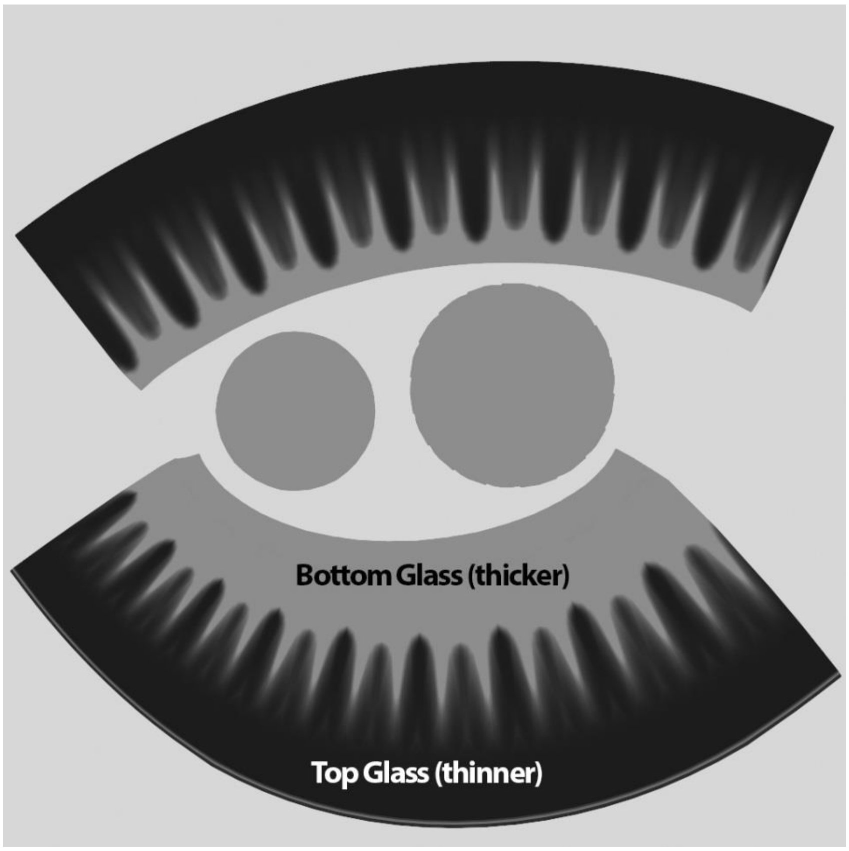
Refraction Amount

The first parameter is the Refraction Amount. This is the overall refraction, used to distort the images behind your glass.

Thickness Map

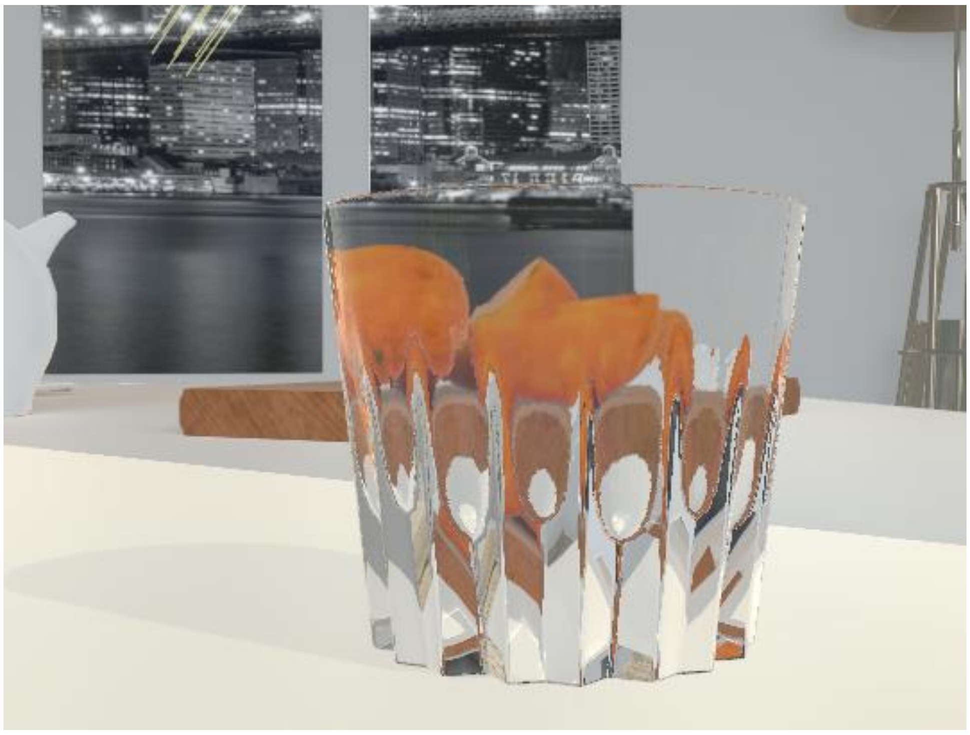
The thickness map is how your glass comes alive. This map is almost like an ambient occlusion map, the difference is it shows darker regions where the object is thinner, whiter regions where the object is thicker.

You create this map basically by inverting the direction of the ambient occlusion rays. Instead of shooting outside, set them to shoot into the model. This way you can probe the model's thickness. You can create this map in your 3D package, with the render of your choice, as well as with tools like Substance Painter.



Thickness map of a corrugated glass. Whiter parts represent thinner glass (less refraction), black parts represent thicker glass (more refraction)

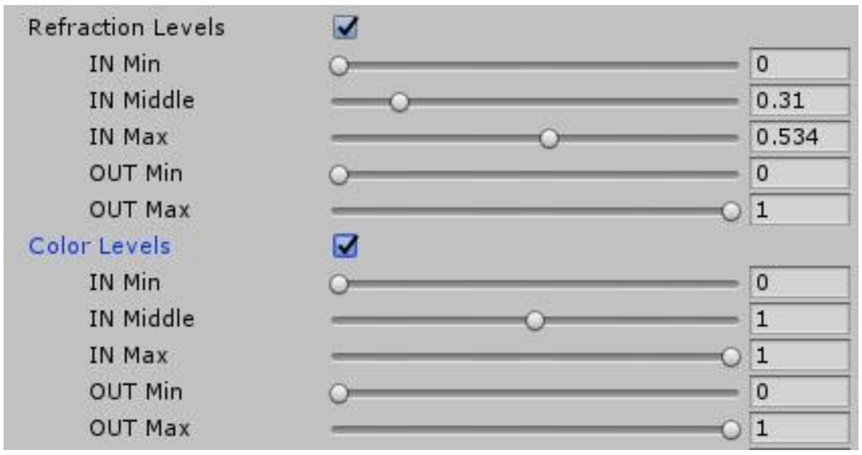
When you assign a new thickness map, you'll see that it drives both the amount of refraction of your glass and its color. Where the object is thicker, the refraction will be stronger and the color more saturated. Of course you can still tweak the refraction by the Refraction Amount slider. You can disable the map affecting refraction or color by unchecking any one of the two following toggles: **Map Affects Refraction** and **Map Affects Color**.



The bottom part of the glass refracts light a lot more. At the top you can see the background with almost no distortion.

Adjust Refraction Levels and Color Levels

You don't have to bake a super precise thickness map, with the correct amount of gray, blacks and whites. You can adjust those values here, with the levels control, right below the thickness map. By enabling it, you will see parameters just like Photoshop's levels adjustments, where you can set the In and Out values for the whites and blacks, as well as the middle gamma. This way you can fine tune how the refraction and color of your glass will look like.



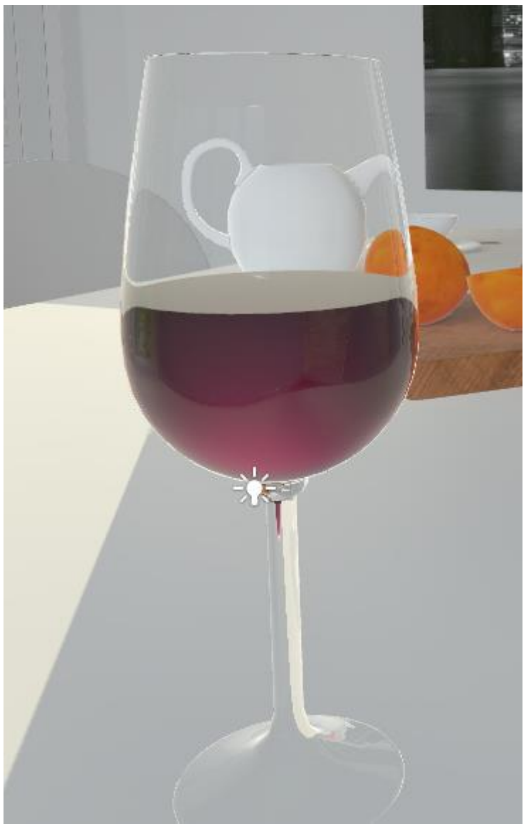
Refraction Levels	✓
IN Min	0
IN Middle	0.31
IN Max	0.534
OUT Min	0
OUT Max	1
Color Levels	✓
IN Min	0
IN Middle	1
IN Max	1
OUT Min	0
OUT Max	1

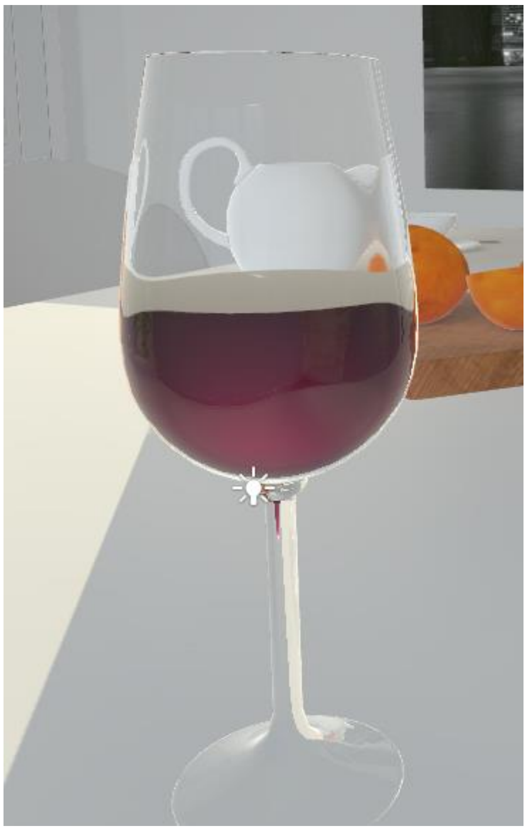
Fresnel Amount and Fresnel Length

While in some cases the thickness map variations may suffice for a good representation of glass, sometimes, when you have thin round glasses, for example, glass cups, you need another feature to fully represent it.

When you see the glass cup, the thin part of the glass right in front of you is almost without refraction, but as soon as the angle between the cup and your vision approaches 90 degrees, you'll see not only more reflection, but more refraction as well, because you're seeing more glass.

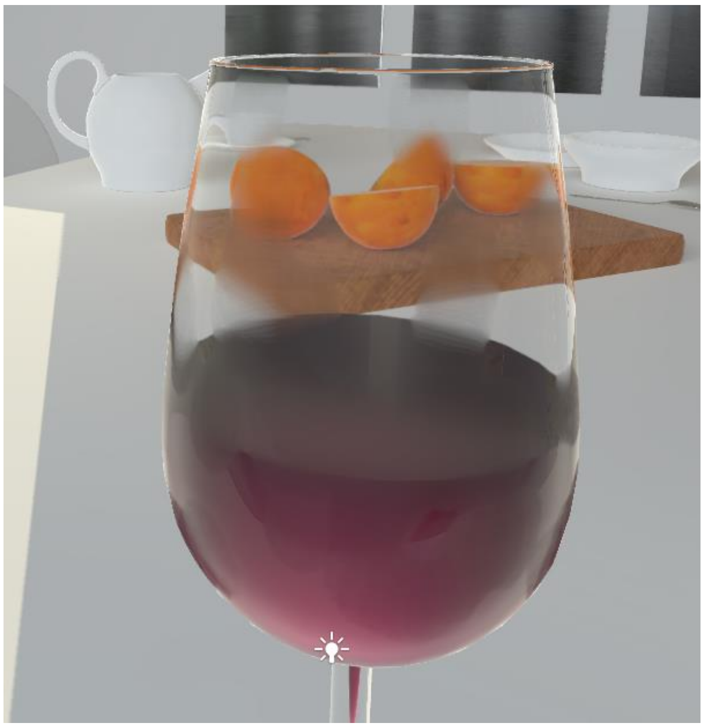
You can simulate this feature with the Fresnel controls. The Fresnel Amount lets you put more or less refraction to the side of the glass, while the Fresnel IOR lets you increase or decrease the length of this effect, i.e., only at the very border or up to the front.


Side borders of the glass with narrow Fresnel.


Side borders of the glass with wide Fresnel.

Blur Amount

The Blur Amount blurs the refraction produced by your material, that is, blurs the objects behind it. It is intended to be used when you add a Smoothness map, for instance, when you want fingerprints on the glass or any sort of dirt. This way, not only the smoothness will be changed by the map, but the refraction behind the object will be blurred too. Of course you can blur the glass, even without a smoothness map. Note that blurs increase computation, so when you don't need it, set it to 0. It will disable the blur calculations.

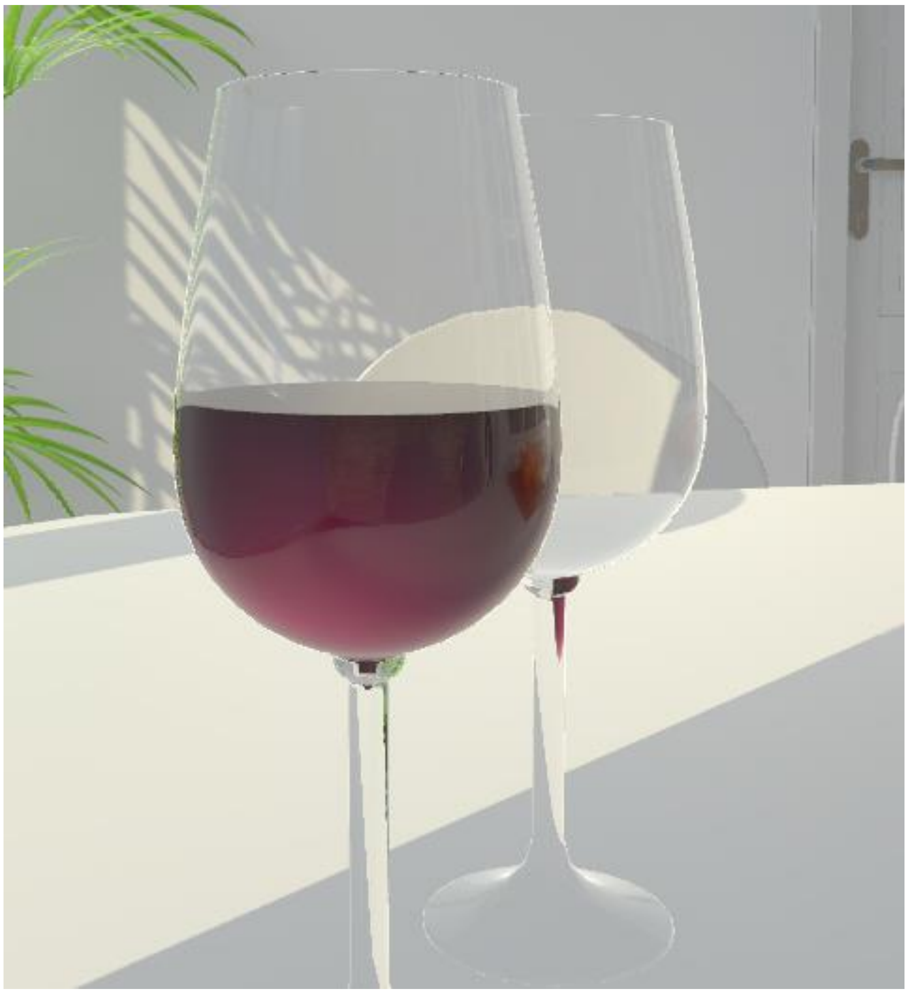


Smoothness map simulating dirt/fingerprints with blur amount set to 5.

Limitation of the Optimized Glass

The optimized glass is very good to be used when you have many objects using the shader. It will not impact negatively your GPU because essentially each glass refraction will use the same internal texture.

A drawback of this is that you will **NOT** see the refraction of other Plus Glass objects behind your object, as the picture shows.



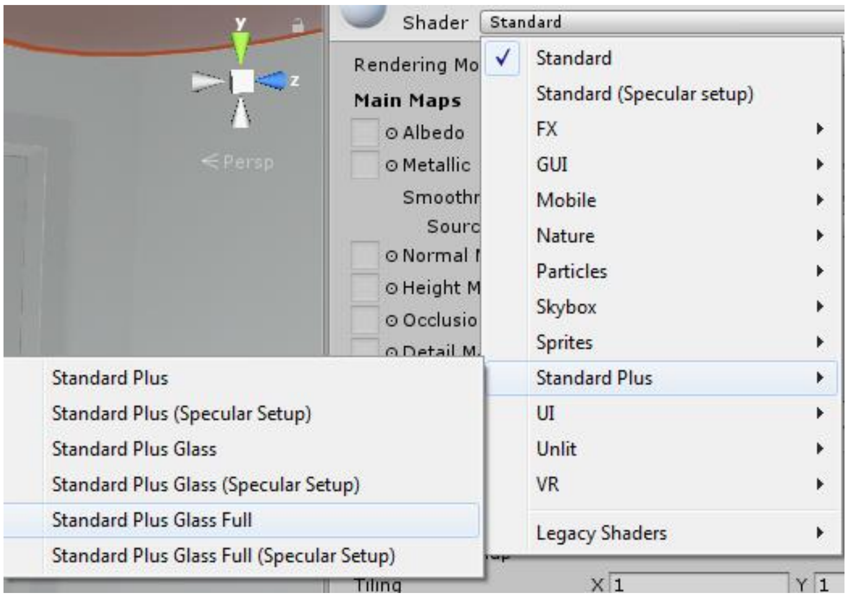
The glass behind disappears in the refraction.

If you don’t need to see other refractive Plus Glasses behind your object, by all means, use the optimized shader.

On the other hand, if you must see other glasses behind, use the **Glass Full** shader shown below.

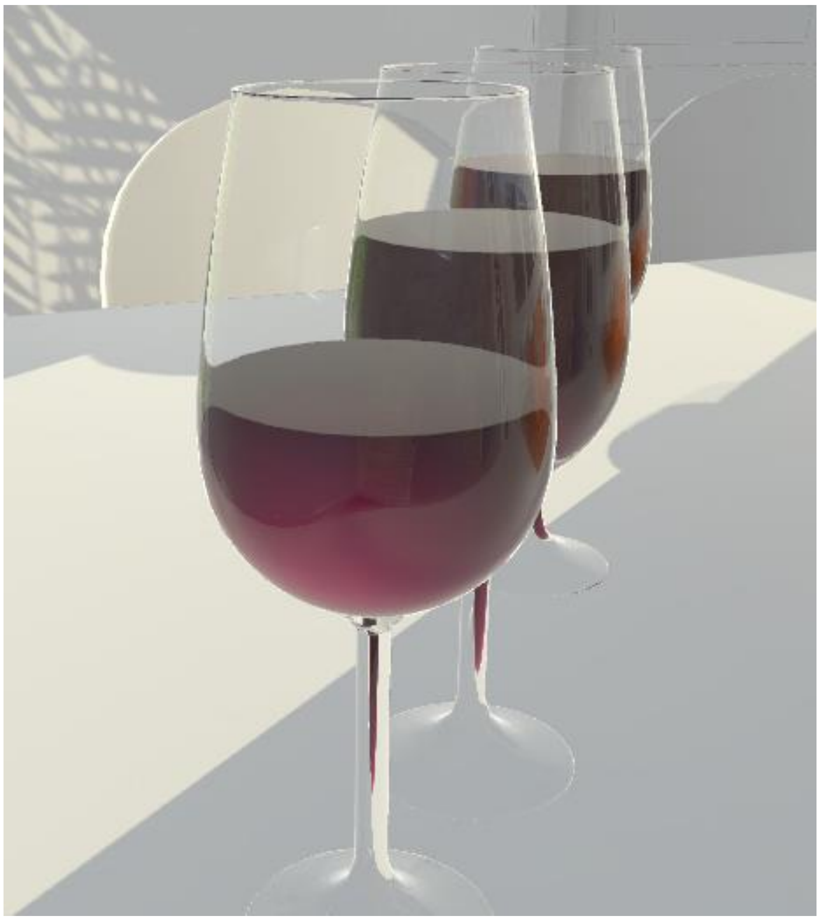
6 – Standard Plus Glass Full

To use the Standard Plus Glass Full, select the shader on the Shader list.



The Standard Plus Glass Full shader has the same controls of the Glass, but also has the ability to show other glasses behind it. You don’t need to enable anything for this to happen. But keep in mind that with each new material using this shader, the system will generate another screen copy of the background, thus making it very expensive if you have, for instance, 10 objects at the same time on screen. When using VR, multiply this by 2, because of the 2 cameras, and you’ll drop the frame rate rapidly.

Use this only when you have an important object to show, that needs to display other Glass objects behind it.



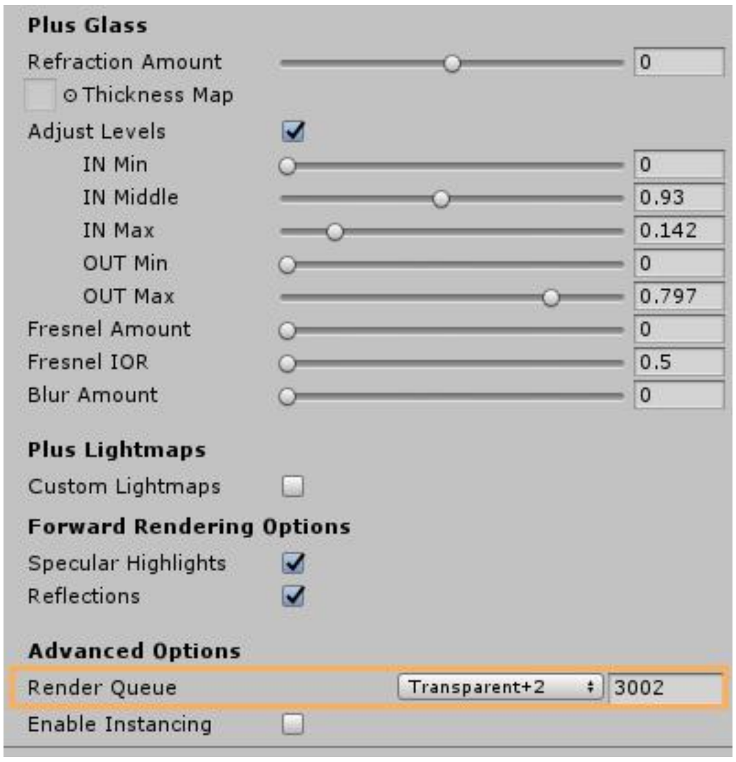
You can see everything refracted behind your glass, with the expense of computational power, of course.

RenderQueue Values

Rendering concave single glass objects with refraction is hard, because transparent objects have an order in which they are drawn on screen. That order is determined by the object's distance from the camera, but sometimes, depending on the angle and the proximity of the objects, the rendering gets confused and objects reverse the order.

The way to remedy this is simple. You have to change the **RenderQueue** value.

The RenderQueue is found near the bottom of the material inspector, in the **Advanced Options**.



Plus Glass

Refraction Amount	0
Thickness Map	
Adjust Levels	✓
IN Min	0
IN Middle	0.93
IN Max	0.142
OUT Min	0
OUT Max	0.797
Fresnel Amount	0
Fresnel IOR	0.5
Blur Amount	0

Plus Lightmaps

Custom Lightmaps	☐

Forward Rendering Options

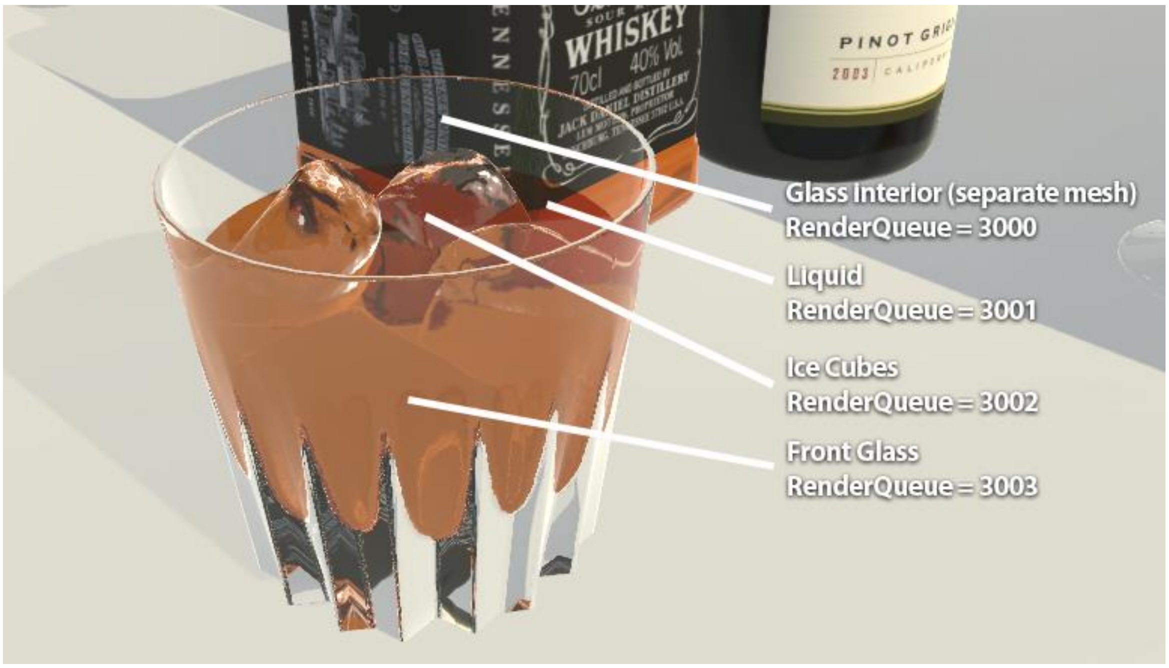
Specular Highlights	✓
Reflections	✓

Advanced Options

Render Queue	Transparent+2	3002
Enable Instancing	☐	

Transparent materials have the RenderQueue set to 3000 by default.

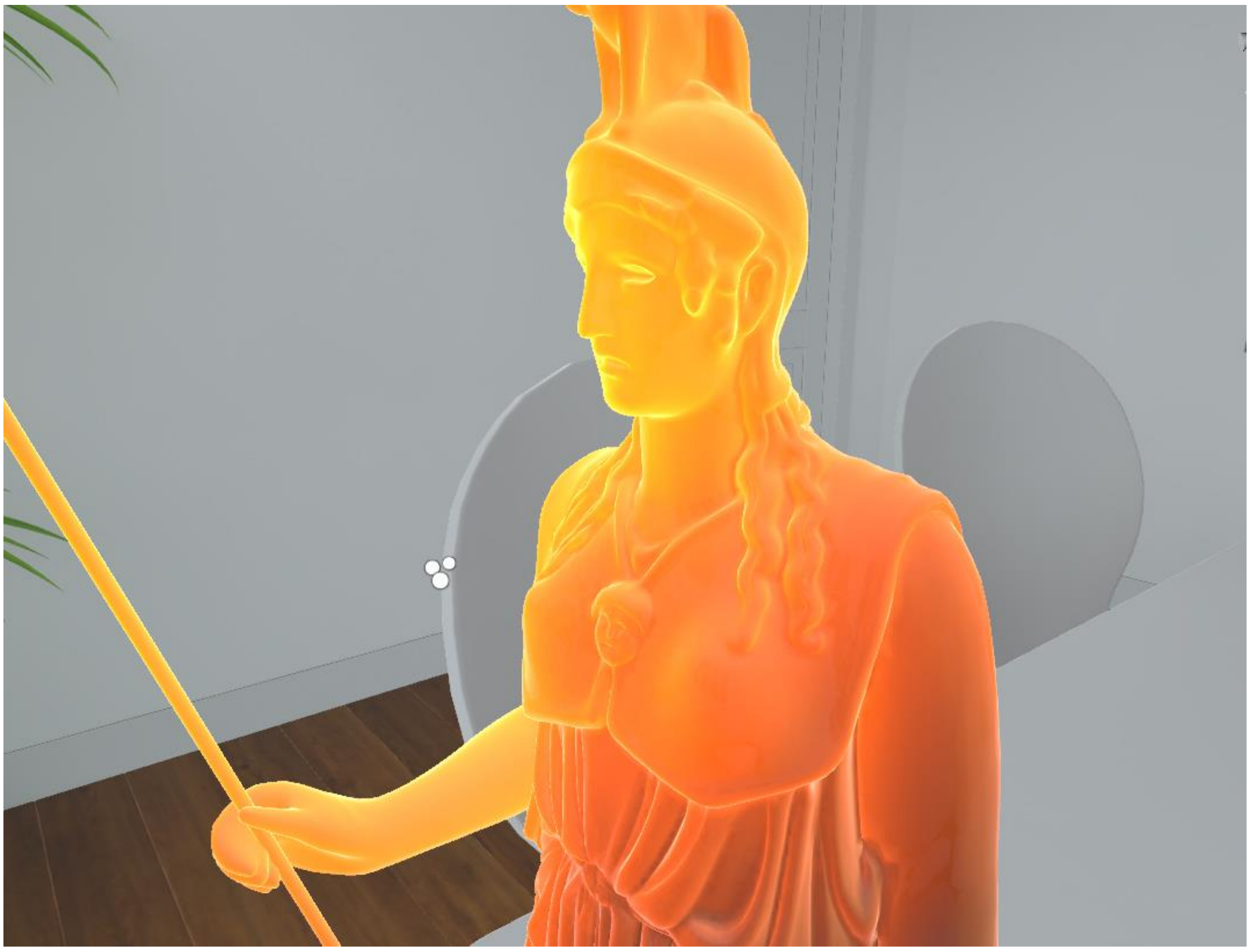
So, when you want to specify exactly the order in which the transparent objects are drawn, set the RenderQueue values to higher numbers. Lowest numbers will be drawn first, and the higher numbers will be drawn for last, on top of everything behind. See the picture below.



All materials here use the Plus Glass Full shader. They are drawn in the correct order, according to their RenderQueue.

7 – Standard Plus Forward Rendering Translucency

The Forward Rendering Translucency allows you to simulate the effect of light passing through your object.

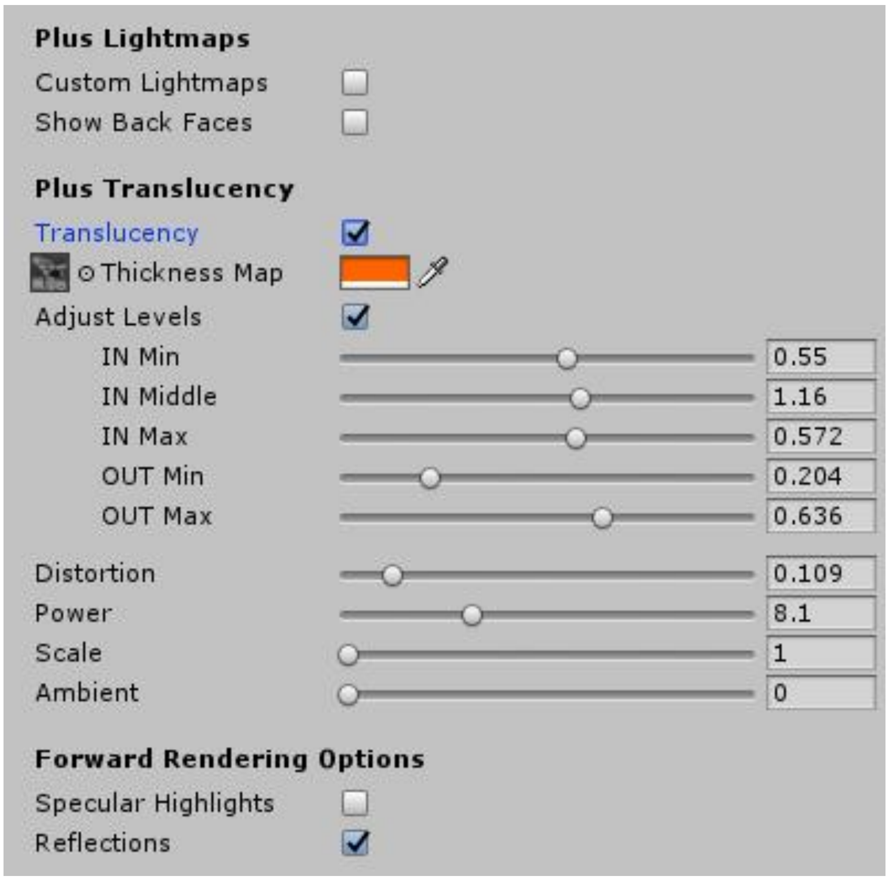


A statue of the goddess Athena showing a translucency effect.

Unfortunately, up to this version (1.0), it only works with the forward rendering path, but I'm working on enabling it on the deferred path for next updates.

To use the Translucency effect, choose the Standard Plus shader, or Standard Plus (Specular Setup).

At the bottom of the material inspector, you will see the translucency properties **Plus Translucency**. When you click on **Translucency**, all the options will appear.



Plus Lightmaps

Custom Lightmaps	☐
Show Back Faces	☐

Plus Translucency

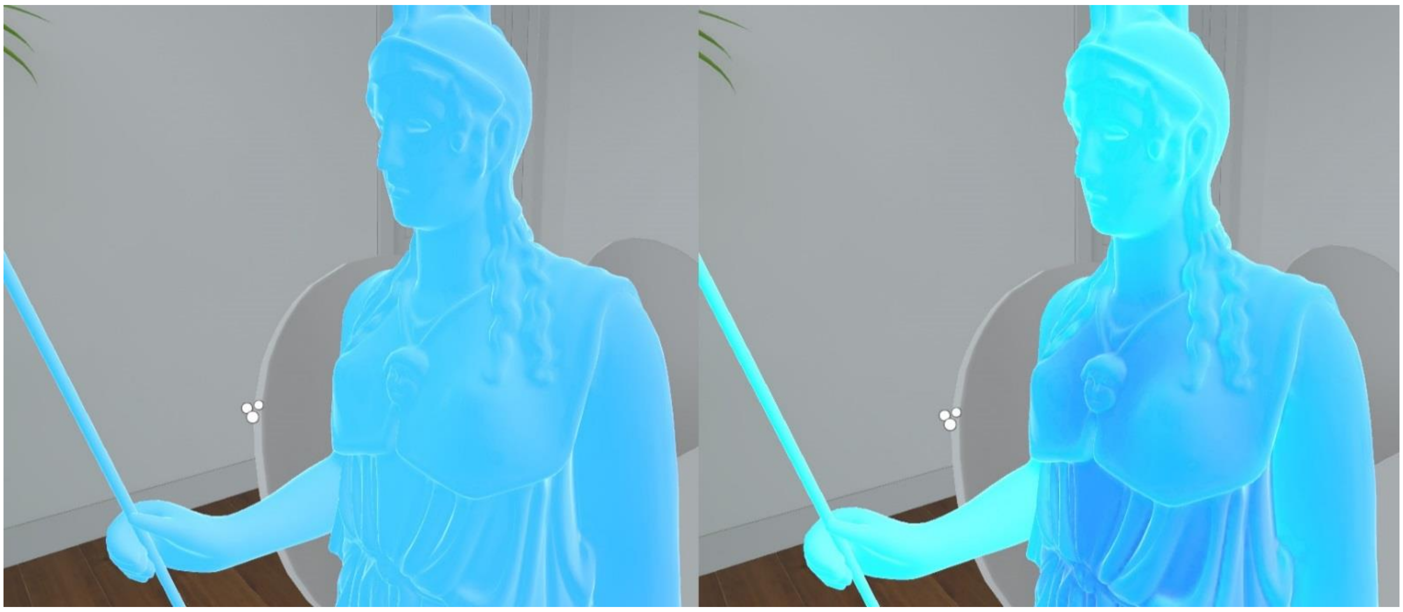
Translucency	✓
Thickness Map	🟧
Adjust Levels	✓
IN Min	0.55
IN Middle	1.16
IN Max	0.572
OUT Min	0.204
OUT Max	0.636
Distortion	0.109
Power	8.1
Scale	1
Ambient	0

Forward Rendering Options

Specular Highlights	☐
Reflections	✓

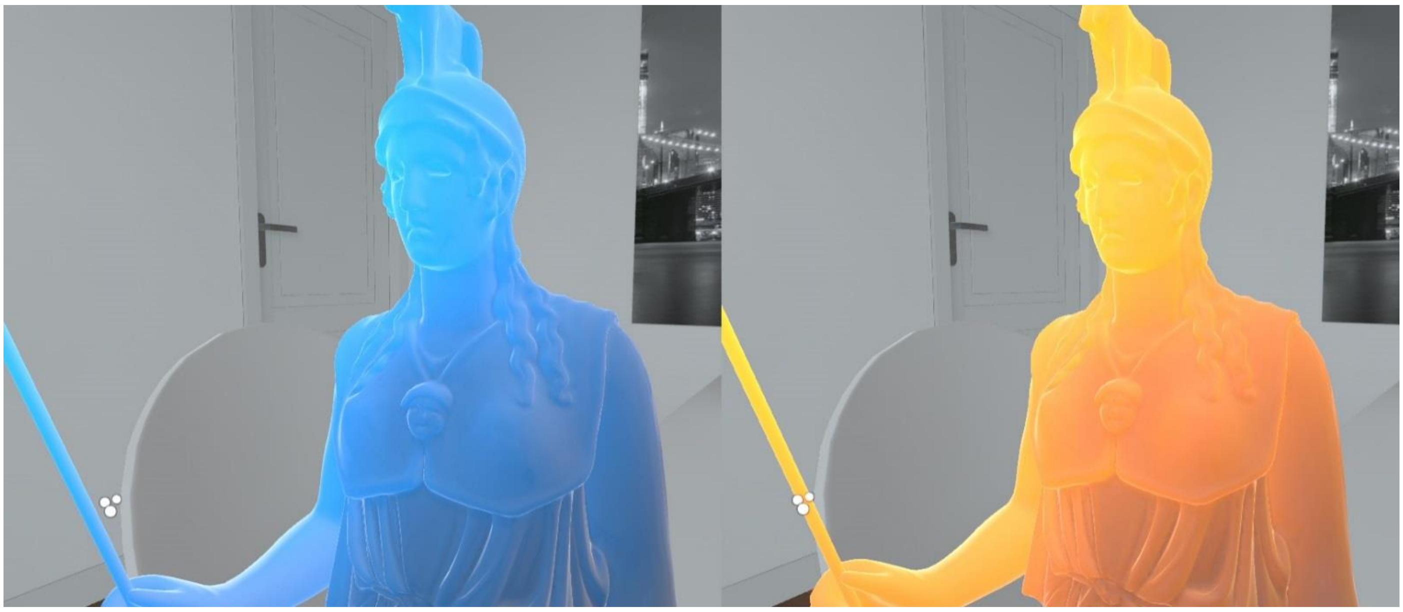
Below is an explanation and example of all the options.

Thickness Map – Like the glass material, the thickness maps represent the thickness of your object and is created the same, by an ambient occlusion map from inverted normals. With this map, you achieve a real variation of thickness in the object, with less light passing through thicker parts.

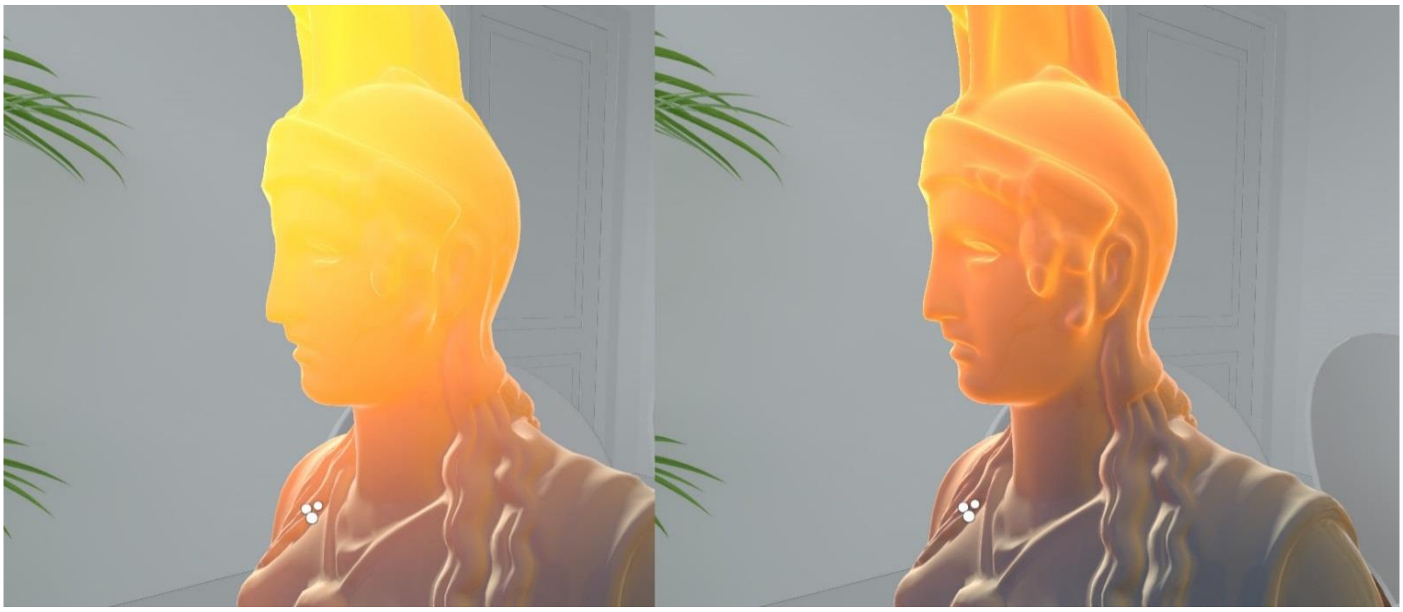


Left side, no thickness map, the color is constant. Right side, with thickness map, the thick parts get darker, thin parts get brighter.

Inside Color – The inside color of your object. This simulates a color that is diffused after the light has penetrated the object. It's independent from the object's albedo color and the light color.

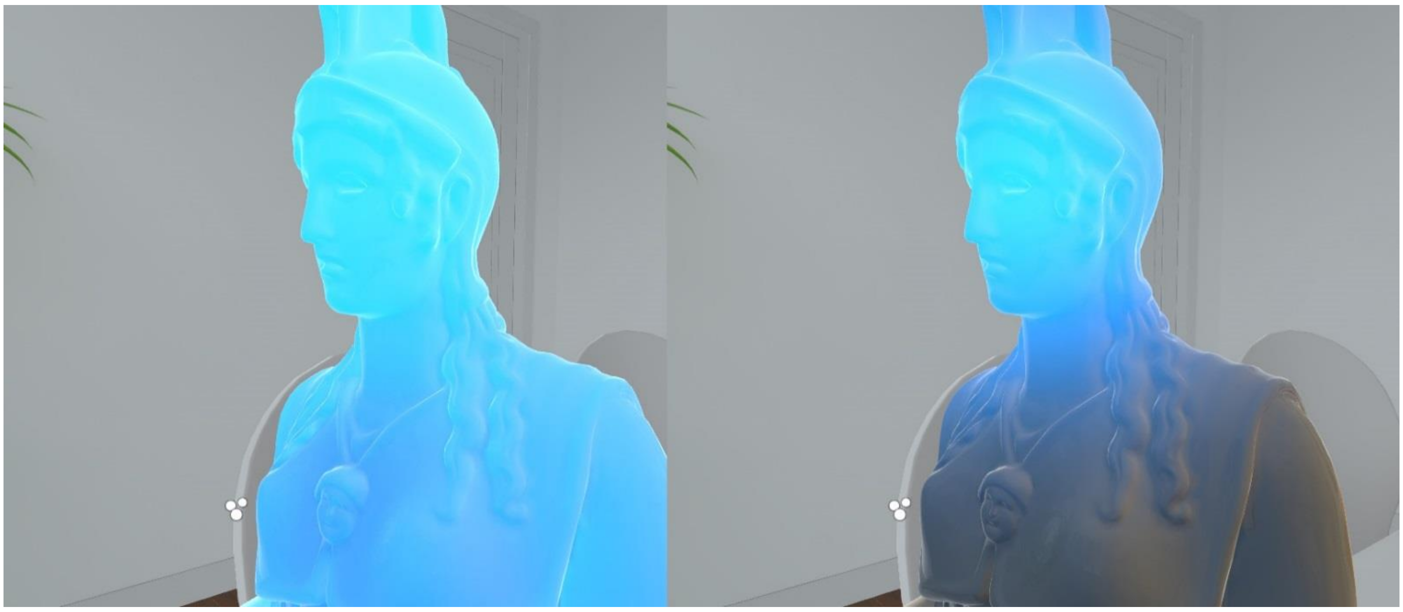


Distortion – This value offset the object's normals, giving almost an impression of a falloff effect, increasing the sense of a subsurface scattering.



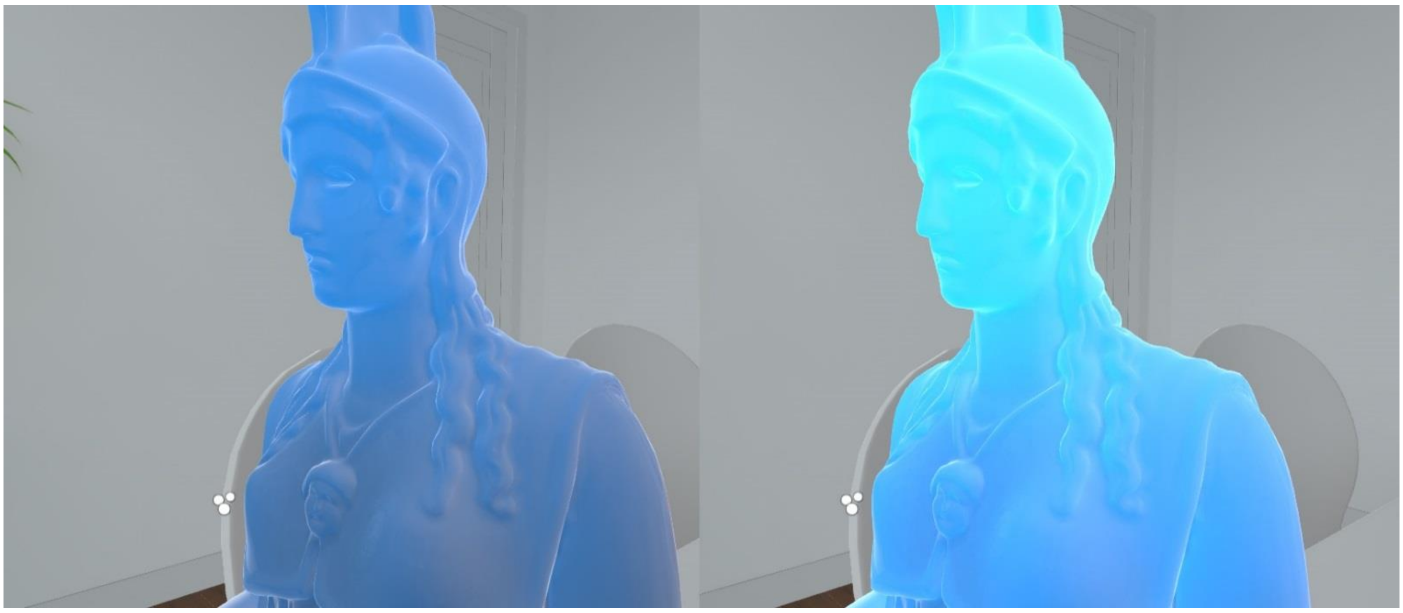
No distortion, Distortion 0.2

Power – Power can be seen as a light focus. As you increase this value, the light passing through gets more condensed, focused, less dispersed.



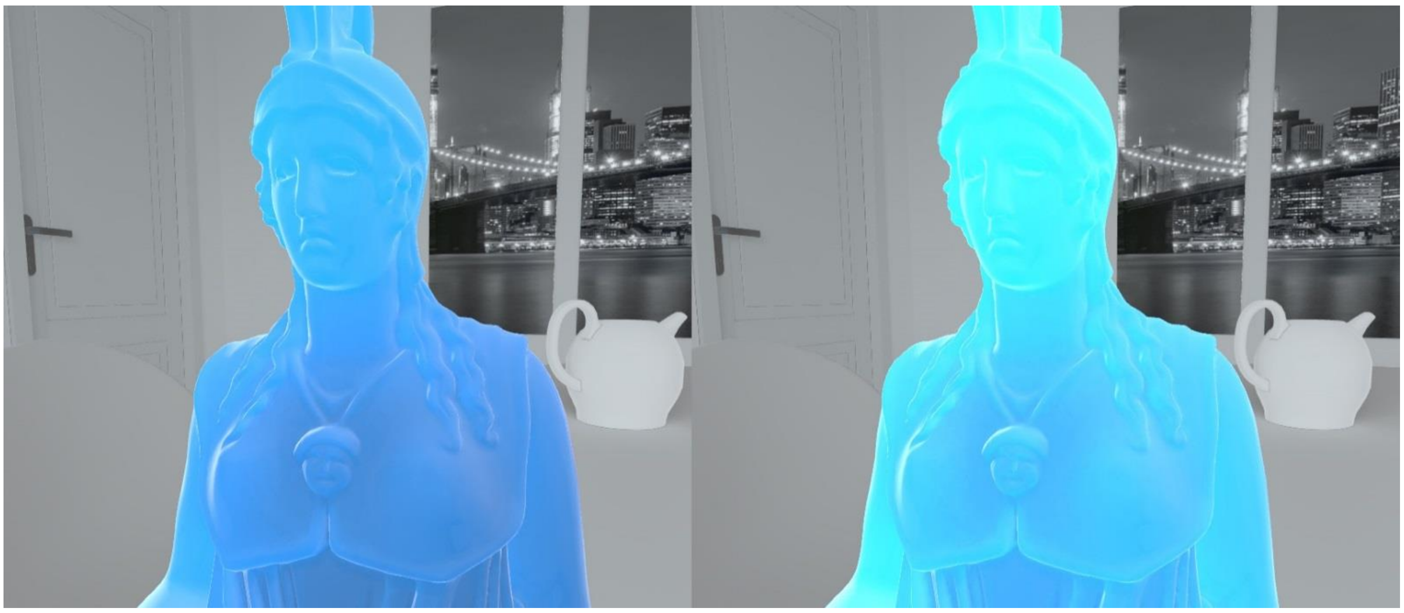
Power 1 Power 21

Scale – Essentially, how much light is transported through the surface.



Scale 3, Scale 10

Ambient – An overall lighting of the object. Increases the inside color uniformly, simulating an ambient light. It has nothing to do with Unity's ambient light or sky light.



Ambient 0, Ambient 5

Shadow limitation

As of version 1.0, if you are lighting a translucent object with a light that produces shadow, the translucency will self shadow. So it's better to turn off the **Receive Shadows** on the object material inspector.

8 – Variable names for accessing parameters via code

Plus Custom Lightmaps

_CustomLightmap = Lightmap texture 1. Defaults to white texture.

_CustomLightmap2 = Lightmap texture 2. Defaults to white texture.

_CustomLightmap3 = Lightmap texture 3. Defaults to white texture.

_CustomLightmap4 = Lightmap texture 4. Defaults to white texture.

_CustomLightmap5 = Lightmap texture 5. Defaults to white texture.

_LightmapTint = Tint color for the lightmap texture 1. Defaults to black color.

_LightmapTint2 = Tint color for the lightmap texture 2. Defaults to black color.

_LightmapTint3 = Tint color for the lightmap texture 3. Defaults to black color.

_LightmapTint4 = Tint color for the lightmap texture 4. Defaults to black color.

_LightmapTint5 = Tint color for the lightmap texture 5. Defaults to black color.

Plus Glass and Plus Glass Full

_ThicknessMap = Thickness map texture.

_RefractionAmount = Overall refraction amount.

Controls for the Refraction or Translucency levels

_INMin

_INMiddle

_INMax

_OUTMin

_OUTMax

Controls for the Color levels

_INMinCOLOR

_INMiddleCOLOR

_INMaxCOLOR

_OUTMinCOLOR

_OUTMaxCOLOR

_SizeBlur = Controls the blur amount.

_FresnelAmount = Controls the Fresnel amount.

_FresnelLength = Controls whether the Fresnel should be only at the borders or not.

Plus Translucency

_ThicknessMap = Thickness map texture.

_TranslucencyColor = Controls the inside color.

_Distortion = Distortion control.

_Power = Power control.

_Scale = Scale control.

_Ambient = Ambient control.