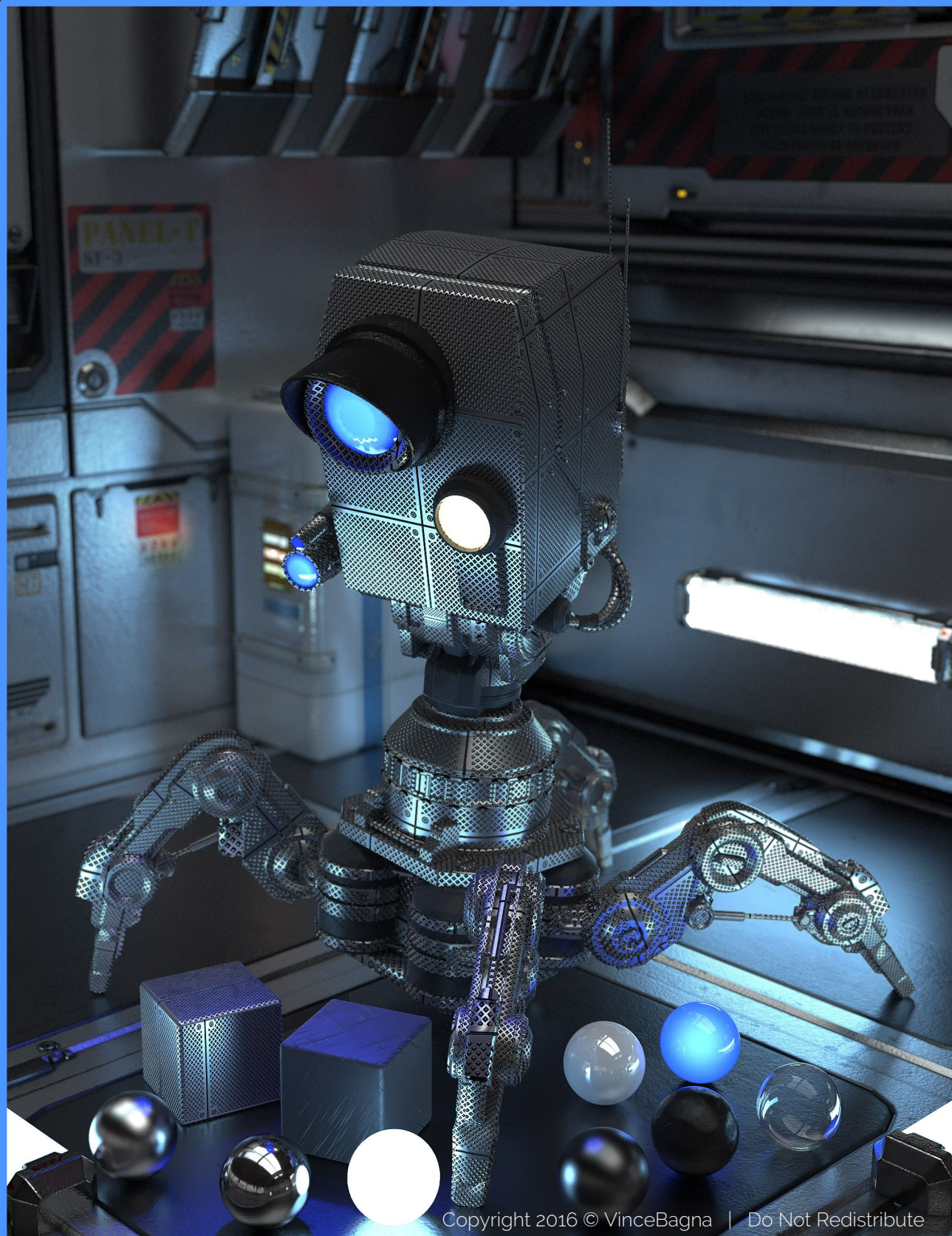


VinceBagna

VB SuperShaders

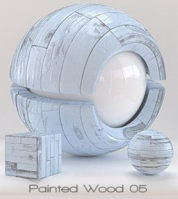
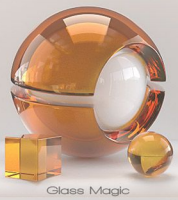
for Poser11 SuperFly

Volume 2



VB SuperShaders for Poser11 - V2

<i>INTRODUCTION.....</i>	<i>3</i>
<i>GENERAL SETTINGS.....</i>	<i>5</i>
<i>BUMP.....</i>	<i>9</i>
<i>GLASS.....</i>	<i>14</i>
<i>LIGHTS.....</i>	<i>15</i>
<i>METAL.....</i>	<i>18</i>
<i>MISCELLANEOUS.....</i>	<i>19</i>
<i>WOOD.....</i>	<i>21</i>
<i>NOTES.....</i>	<i>23</i>





Introduction

First of all, I will try to explain a little bit how the new Poser render engine SuperFly works. Hey, don't be scared, it will be veerrry simple, you'll see ;)

The last iteration of Poser has added (among other exciting features) a new render engine called **SuperFly**, and based upon a renderer called **Cycles**. Cycles is used as an add-on renderer in Blender. It's been a long time since the addition of the last render engine in Poser, FireFly, and it was really time to upgrade that.

You have perhaps already heard about this new hype everywhere, **PBR**, no? Keep calm, no secret name here, **PBR** is used for **Physically Based Rendering**. What is this? Basically, it's a new way to treat lighting and surfaces by changing how light bounces off. In short, no need to trick now to make your renders look realistic. The physical behavior of the light and the shaders (energy conservation, Fresnel...) is already computed by the render engine, so the result is instantly realistic. There is way more to say about that, but it's the most important thing you should know. I said I was explaining in short, no? ;)

If you really want the full story about how PBR works, then you could go there to have more infos ---> [BASIC THEORY OF PHYSICALLY BASED RENDERING](#).

Cycles is not a PBR, so SuperFly is not either... Wow! Don't look at me like that! I explained what SuperFly is not, so you can understand easier what it is. FireFly is not a PBR, it's better than that! It's a probabilistic ray-tracing engine that adds randomness when it traces the paths of rays of light at slightly different angles. For that, it uses BSDF shaders (Bidirectional Scattering Distribution Function) which determine in which direction the ray will bounce.

In one simple sentence: **you can have realistic results, and also unrealistic ones!** Isn't it great?!

You are probably beginning to think I've lost my mind, we could do unrealistic renders before SuperFly, and realistic ones too. But we were very limited, in a way or in another. Now it's supra easy to get realistic results, and not less easy to have unrealistic ones. By unrealistic I mean: we can get ARTISTIC results, If we could render with real light behavior ONLY, with shaders that work like they should do in real life ONLY (think metal, glass and all), it would be very depressing, right? If you want a vampire not to be seen in a mirror (everybody knows that real vampires are invisible in the mirrors, right?) but your render engine says that mirrors reflect everything, vampires included, you'll be forced to use post production to make your effect. Now SuperFly will make your mirror shader acts like real mirrors do indeed, BUT you can get around that and make your mirror not reflect something, or reflects it with

another color, or without shadow, or... excuse me, I'm getting excited, SuperFly is way more than a true PBR, it lets you stay Artists.

Finally my explanation has been longer than expected, but if you get till this point, it was not that terrible, huh?

I will try in the following pages to explain some of the shaders I made. They are real easy to understand, trust me. Basically, one node with several settings, labeled in a way to catch it fast! Some I created while playing with Poser 11 during the Beta phase or after, others I was inspired by the guys at [BLENDER ARTISTS](#) or [BLENDER GURU](#) or elsewhere, doing some changes and improving it while keeping it simple and intuitive. Seriously guys, try to play, with the SuperFly nodes or my shaders, mix them, make some experiments, and most important, **have fun!!** It's the best way to understand and to do great materials!

Now the best way to understand SuperFly is to use it. So let's go!



General Settings

You'll find several settings you can change in these shaders. They are globally the same you can find with the Cycles nodes in the Poser 11 material room.

Diffuse Color

Controls the overall color of the shader. Generally an image map can be plugged here.

Glossy Color

Controls the glossy/reflection color (in SuperFly it's the same). Generally white, can be colored to create an effect.

Roughness

It can refer to the diffuse, reflection or transparency properties. Roughness can turn diffuse surfaces more light absorbing, reflections more blurry, and glass more foggy.



Diffuse Roughness 0

Diffuse Roughness 0.5

Diffuse Roughness 1

(also known as Oren-Nayar shading)



Glossy Roughness 0



Glossy Roughness 0,1



Glossy Roughness 0,5



Glass shader Roughness 0



Glass shader Roughness 0,02



Glass shader Roughness 0,2

Coat Color

Controls the color of the extra layer of coating (i.e. secondary glossy). Can sometimes go with a Coat Roughness setting that works the same as Roughness.

Brightness

Controls how much glossy the shader is. Works like Specular_Value in FireFly root node.

IOR

Stands for Index Of Refraction. There are 2 kinds of IOR:

- If for Glass, it controls the amount of refraction to occur.
- If for other shader like plastic or metal, it is used with Fresnel to control the reflectivity change depending on the surface / viewing angle.



IOR 1,35

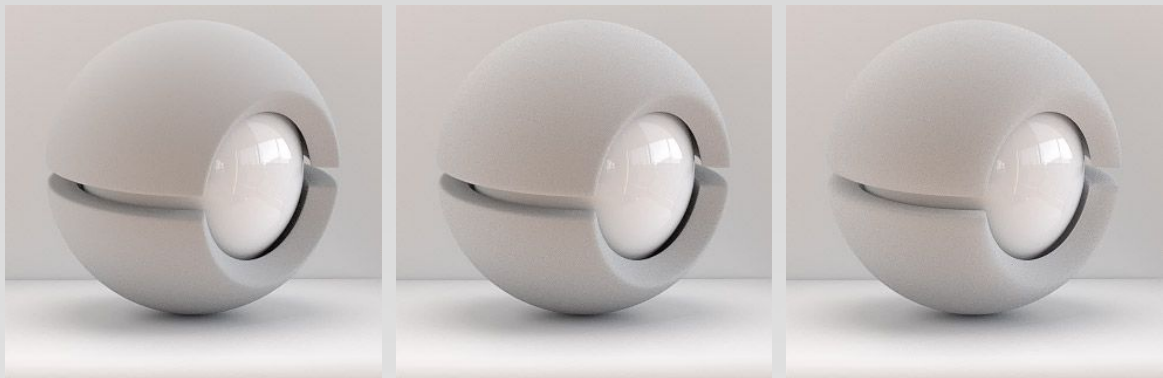
IOR 1,45

IOR 1,85

SSS

Stands for Subsurface Scattering. It simulates the scattering of light beneath the surface of an object.

- SSS Color controls the color of the effect.
- SSS Scale controls how far the light goes through the surface.



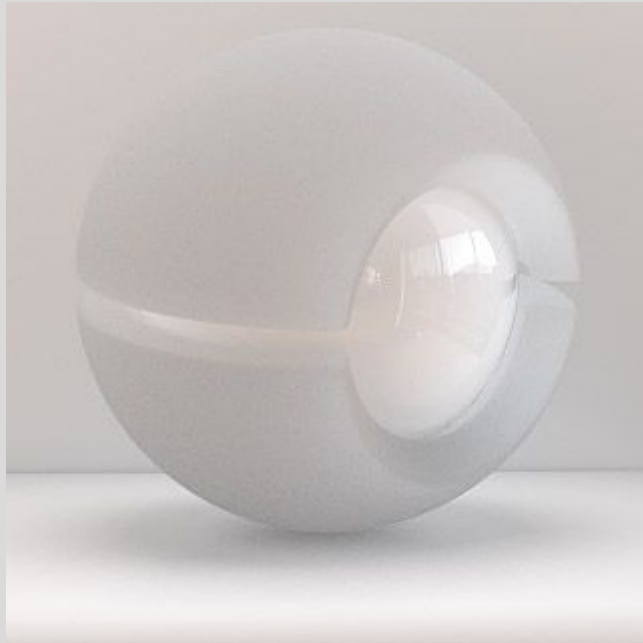
SSS Scale 0,0

SSS Scale 0,01

SSS Scale 0,05

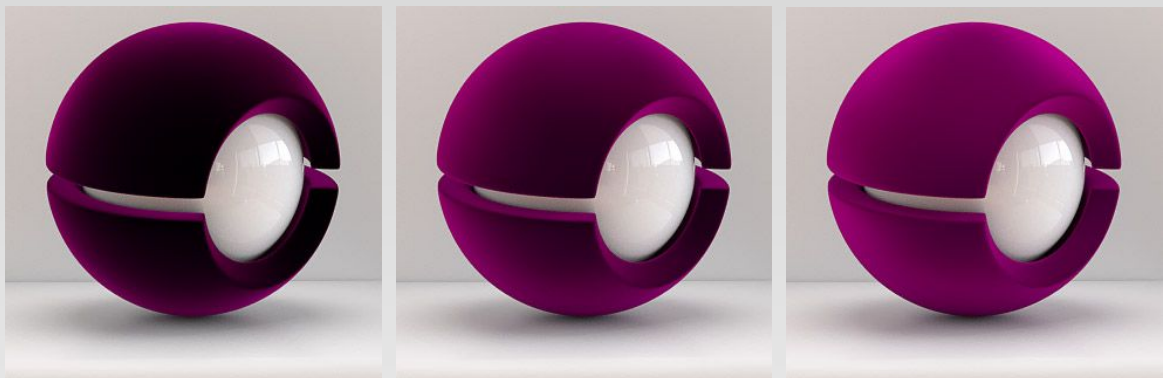
Translucent

Lets light passes through it and bounces back to light up the front faces. So take care to have enough Transmission Bounces in your render settings, otherwise you will have a dark shader, or completely black if no Transmission Bounce at all.



Velvet

Like real velvet fabric, for clothing and fabric. It gives the effect of thin hair all over the shader that bend light accordingly.



Velvet Sigma 0,35

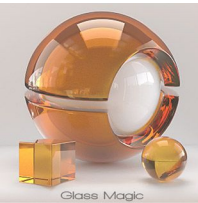
Velvet Sigma 0,65

Velvet Sigma 1

Map Projection

Image map based shaders have an alternative for objects without UV unwrapping, they are called "Whatever Name NoUV". With these shaders you will find new parameters :

- Map Projection : projection to use to map the texture.



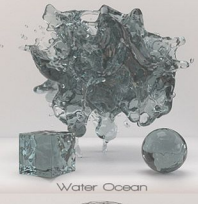
Glass Magic



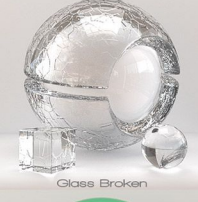
Painted Wood 05



Glass Mirror



Water Ocean



Glass Broken



Plastic Translucent



Ice 02



Concrete Polished 03



Leaf



Plastic Scratched 02



Satin Bump

- *Box* will map the image to the 6 sides of a virtual box, based on the normal, using XY, YZ and XYZ coordinates depending on the side.
- *Flat* will use the XY coordinates for mapping.
- *Sphere* will map the image to the sphere using Z axis as central.
- *Tube* will map the tube to the sphere using Z axis as central.

- Mapping : it's where you can offset, rotate or scale your image map.

- Texture Coordinate : here you can change the type of coordinate to use for the texture. There is a bunch of different options, I will cover the 2 main ones.

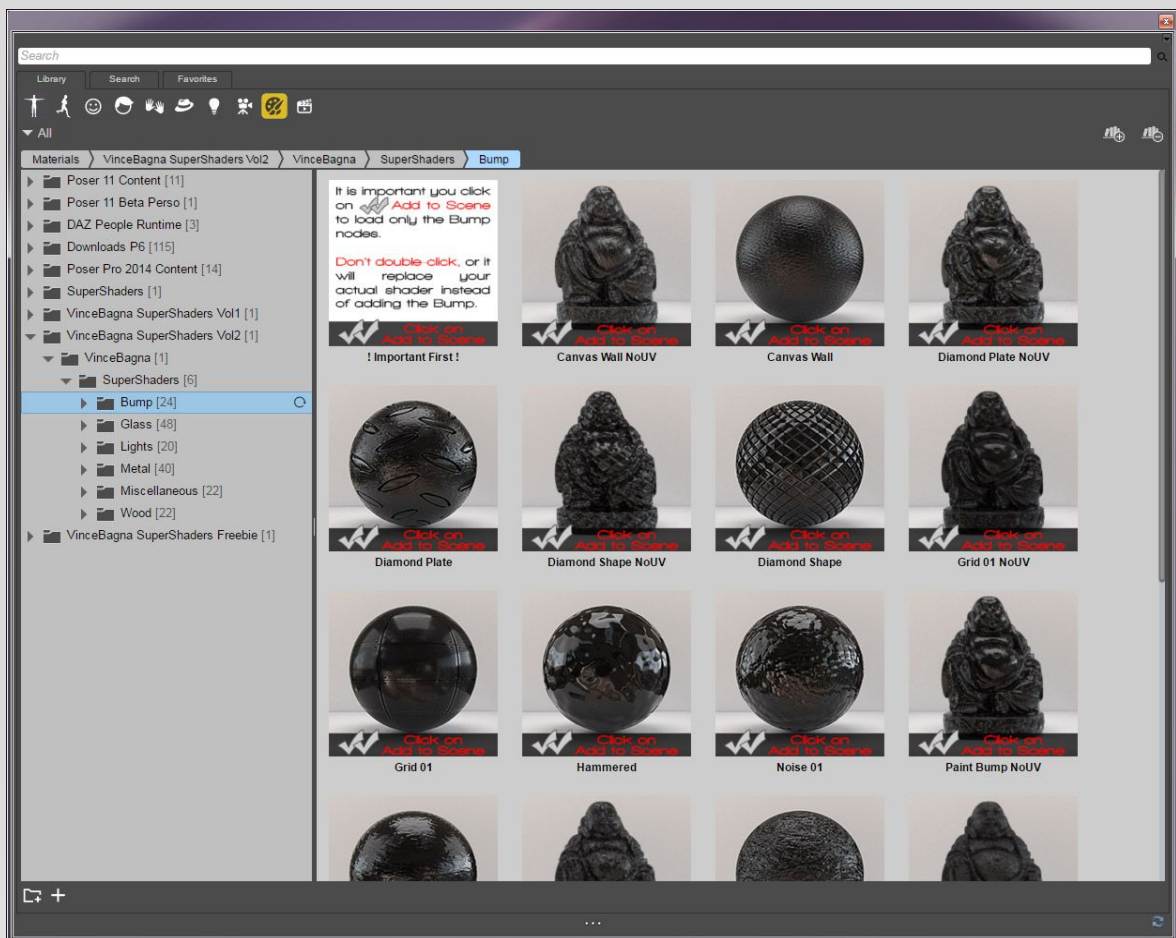
- *Generated* : automatically-generated texture coordinates from the vertex positions of the mesh without deformation. In short, for non UV unwrapped objects.

- *UV* : to use the UV map of the object.

Using Box Projection with Generated Texture Coordinate for non UV unwrapped object is far to be perfect, mostly with patterned image map. Though you can quickly texture non unwrapped objects easily, even with the other Map Projection options.

Bump

You will find a folder called **Bump** with several bump nodes to add to your shader,



You will probably try first to load them like any other material files, by double clicking on it. **MISTAKE!** Haven't you seen that first thumbnail called **! Important First !** ? It's there to tell you there is something important about the Bumps there,, obviously ;)

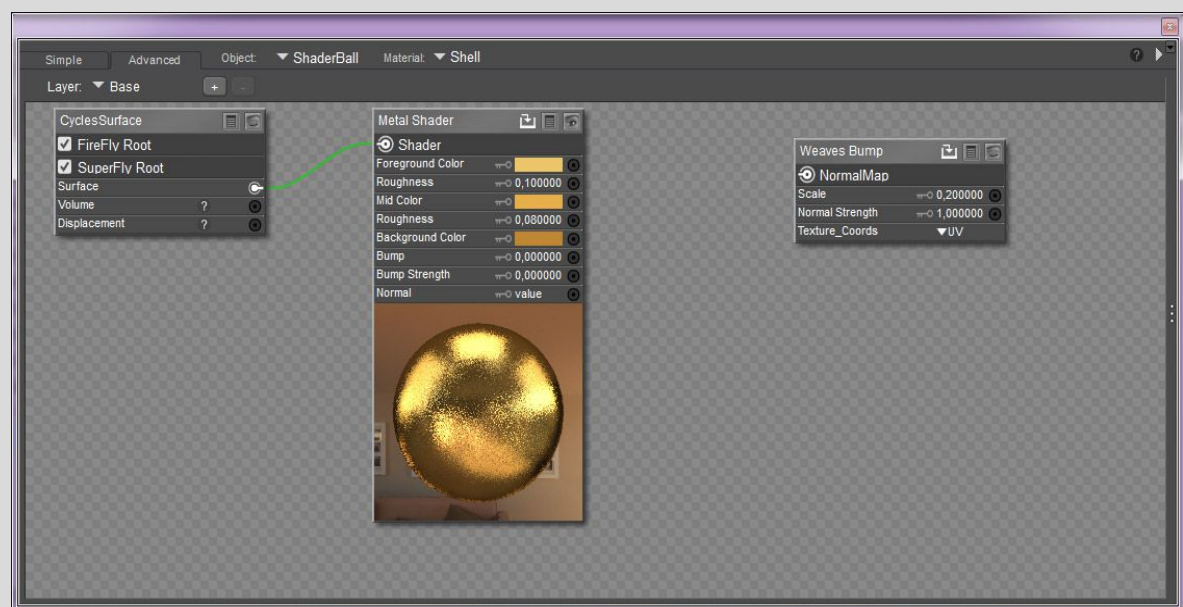


It is important you click on **Add to Scene** to load only the Bump nodes.

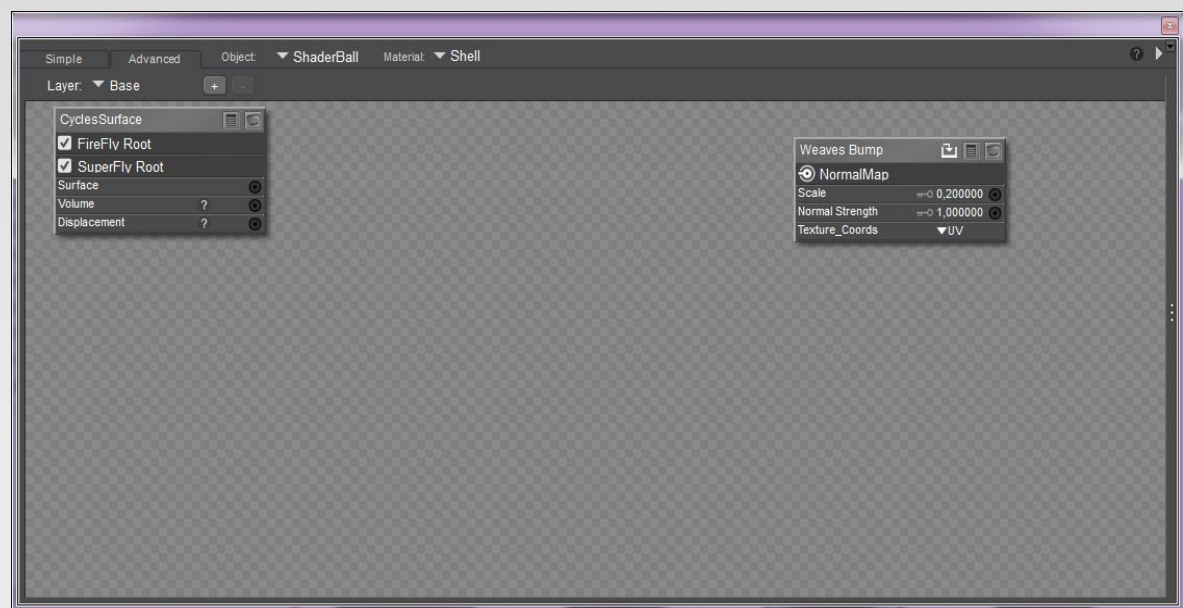
Don't double-click, or it will replace your actual shader instead of adding the Bump.

Click on Add to Scene

And those cute little double check marks on the bump icons ? Don't tell me I have made them for nothing. They are there to tell you to load the Bumps by first selecting the one you want, then click on the double check mark at the bottom of your library to **add** them in your material room, and to **not replace** the shader you have there,



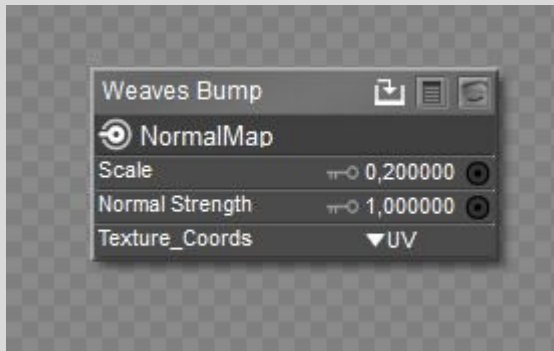
Add a Bump by clicking on the double check mark, then plug it into the bump input if it's a bump map node, or the normal input if it's a normal map node.



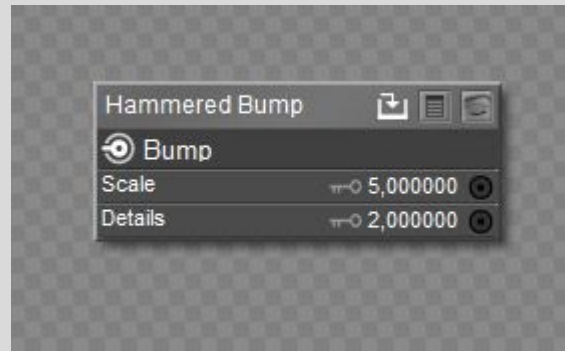
Load the Bump by double clicking on its thumbnail, bad bad bad :)

Keep calm if you already double click faster than light on it, don't forget CTRL+Z is your God *giggle*

Now, you will probably note that some Bump nodes are NormalMap, and some others are BumpMap.



Next to the node output, see Normal Map.

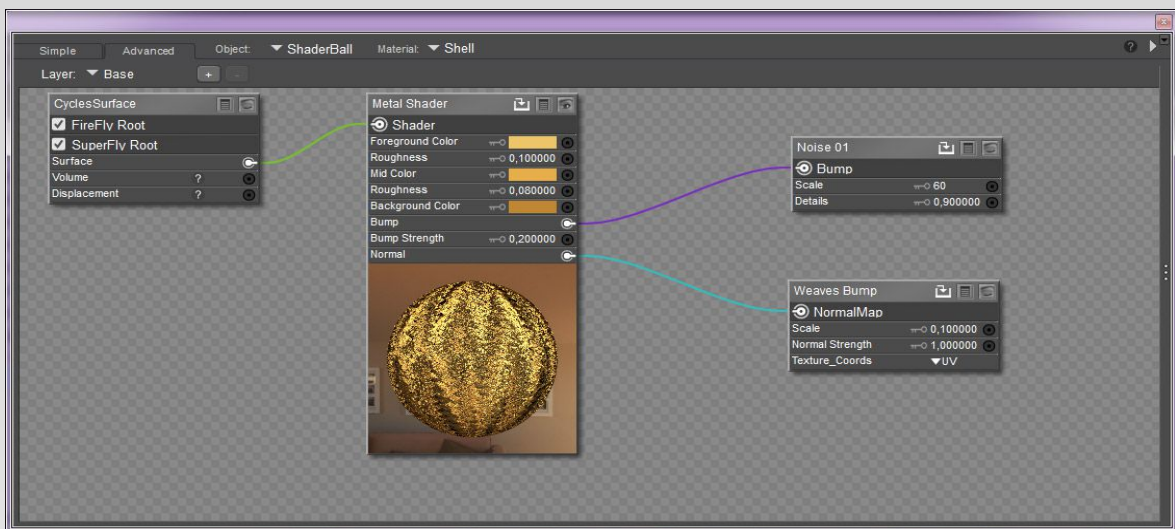


And here, see BumpMap.

Plug the NormalMap output into the Normal input of the shader node, and set the strength with the Normal Strength slider of the Bump node.

Plug the BumpMap output into the Bump input of the shader node, and control the strength with the Bump Strength slider of the shader node.

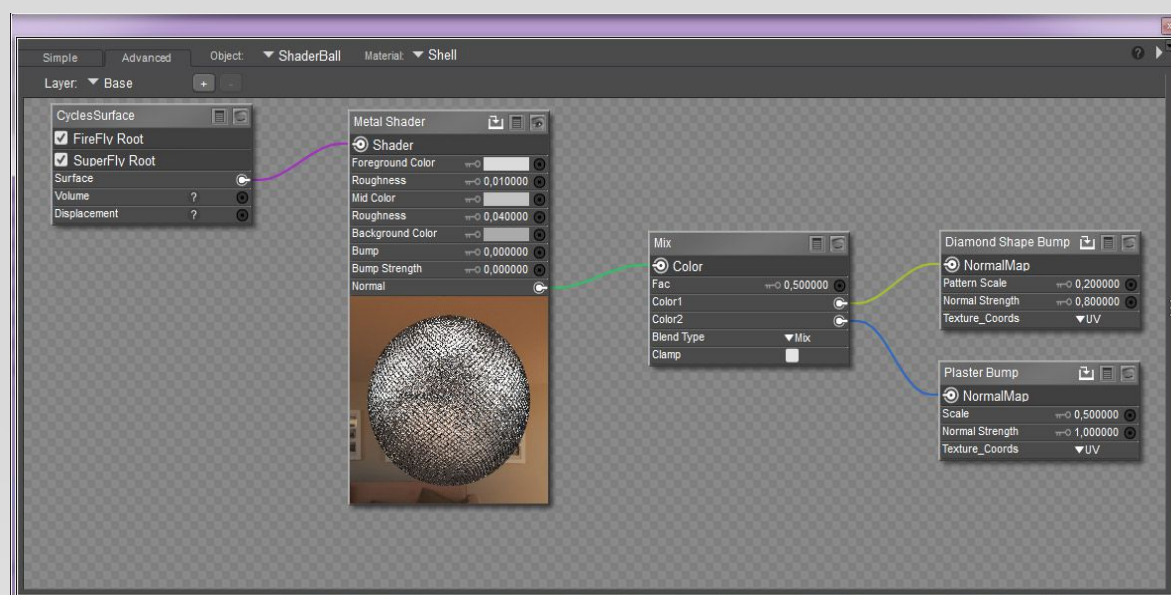
Hey, you can even use a NormalMap node AND a BumpMap node at the same time.



It can give great results!



Or even mix 2 NormalMap nodes or 2 BumpMap nodes together with a Color Mix node!



A lot of possibilities and so much fun!



Glass Magic



Painted Wood 05



Glass Mirror



Water Ocean



Glass Broken



Plastic Translucent



Ice 02



Concrete Polished 03



Leaf



Plastic Scratched 02



Satin Bump



Glass

Glass was something rather tricky to get right with FireFly. Now it's real easy with SuperFly!! ^-^

ArchViz Glass

Useful for building glass that should let light go through. It gives you great reflections too :)

Glass Basic

This shader already takes care of the glass shadow issue the Cycles render engine has. So with this shader you get nice transparency and great shadow/caustics.



Glass Coated

Glass Coated has additional settings as refraction and glossy/reflection are separated.



Refraction Roughness 0,1

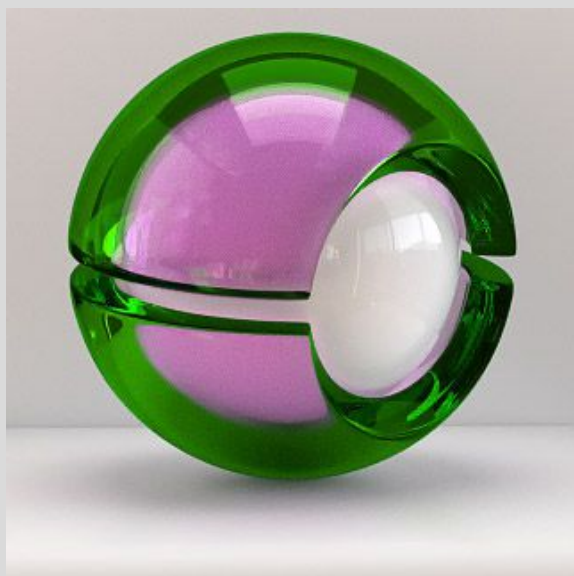


Glossy Roughness 0,1

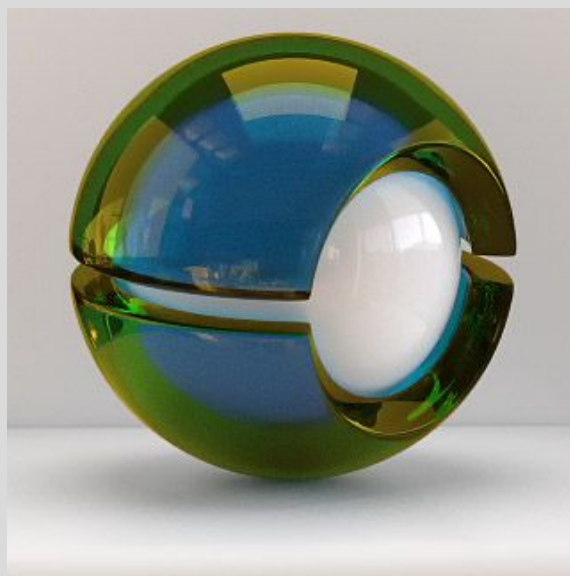


Both Refraction and Glossy Roughness 0,1

You can also set the colors separately for Refraction and Glossy.



Pink Refraction and green Glossy



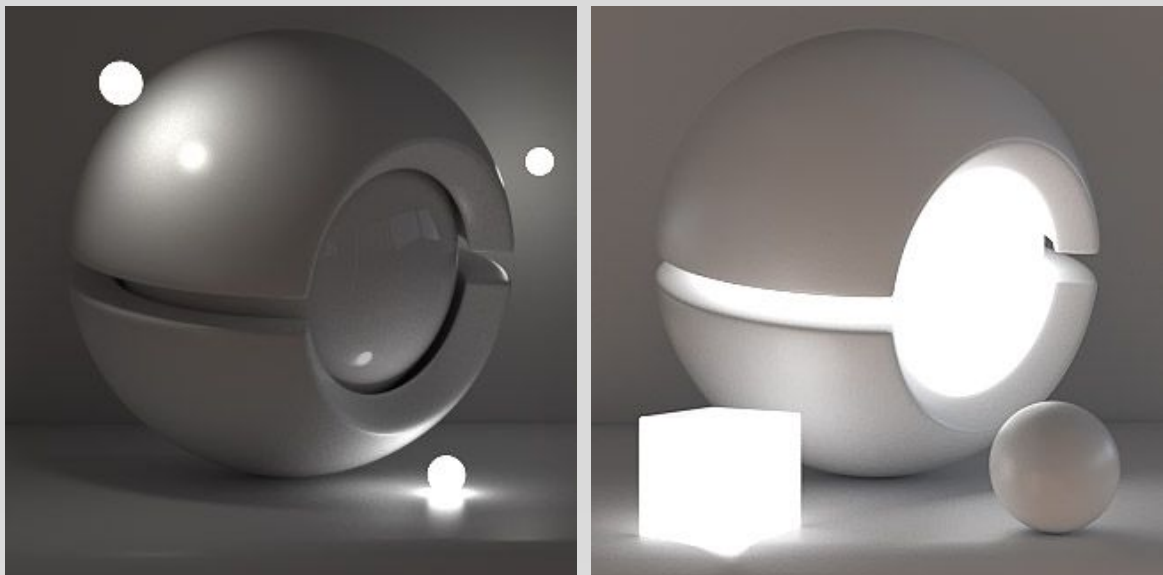
Blue Refraction and yellow Glossy

Lights

There are 2 kinds of shaders in the Lights folder :

- classic shaders you apply to a surface
- and shaders for an Environment Sphere or for the Poser Background to use for lighting.

Classic shaders are to apply to a surface like any other shader, but this time it will give you LIGHT!



The HDRI shaders are to be used either with the Poser Background, or with an Environment Sphere.

I would recommend you to use the Poser Background as the lighting with use more efficiently the HDRI dynamic range and will produce better shadows,

The HDRI shaders 1 Map are to be used when you have only 1 HDRI map (too easy, no?).

The HDRI shaders 2 Maps are when you have 1 map for lighting and 1 map for background (basically 1 HDRI for lighting and 1 JPG for background).

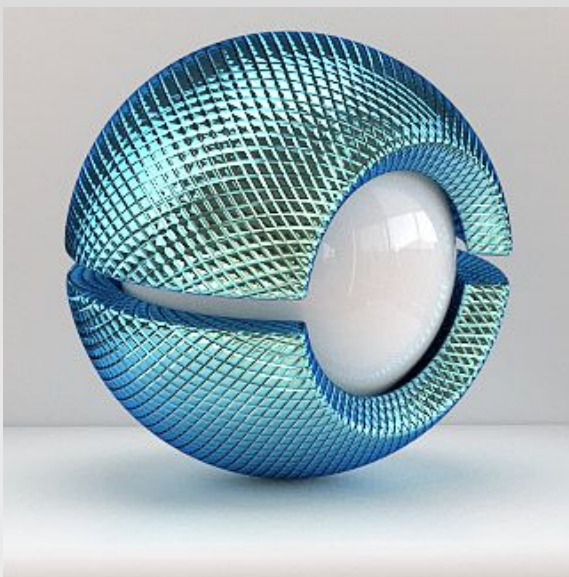
The HDRI shaders 3 Maps are when you have 1 map for environment lighting, 1 map for reflection and 1 map for background.

Metal

Metal shaders have a lot of parameters you can tweak to alter the look of the shaders. You can go from a high polished silver to a rough metal in just a couple of clicks.



Also don't forget you can play with the shaders by adding some bumps :)



Miscellaneous

Here are all the shaders that don't fall into the other categories.

So you get some FX shaders like Plasma, Hologram or Ghost, some specialties like Dust And Dirt, Fake AO and Color Blend, and also some Toon Shaders.



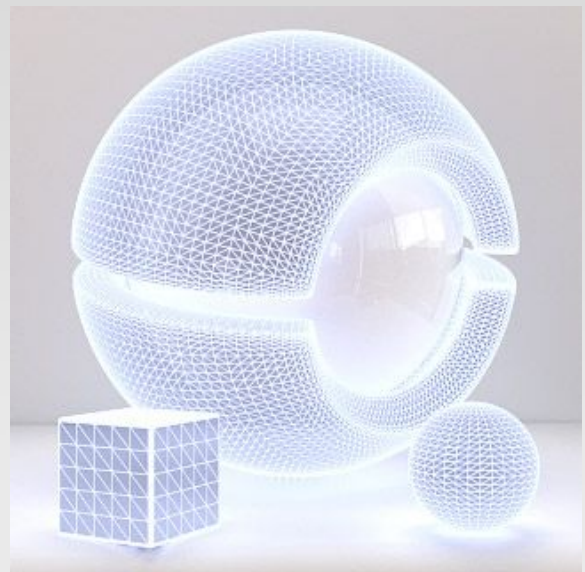
A neat one: Dust And Dirt... not that neat finally.



Shader Blend, try with other shaders for more fun ^^



Silhouette, simple, effective.



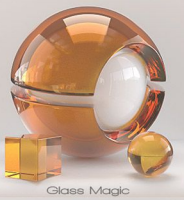
Hologram, choose color and line thickness.

Wood

Here you will find some wood shaders. All are map based. And each shader has a painted version too. And you know what? You can choose the paint color! ^^



Wood 05. Pretty, no?



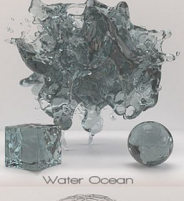
Glass Magic



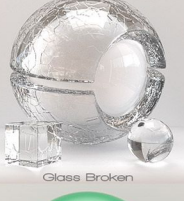
Painted Wood 05



Glass Mirror



Water Ocean



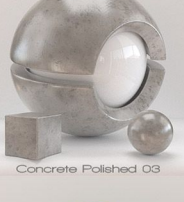
Glass Broken



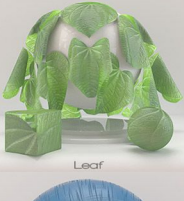
Plastic Translucent



Ice 02



Concrete Polished 03



Leaf



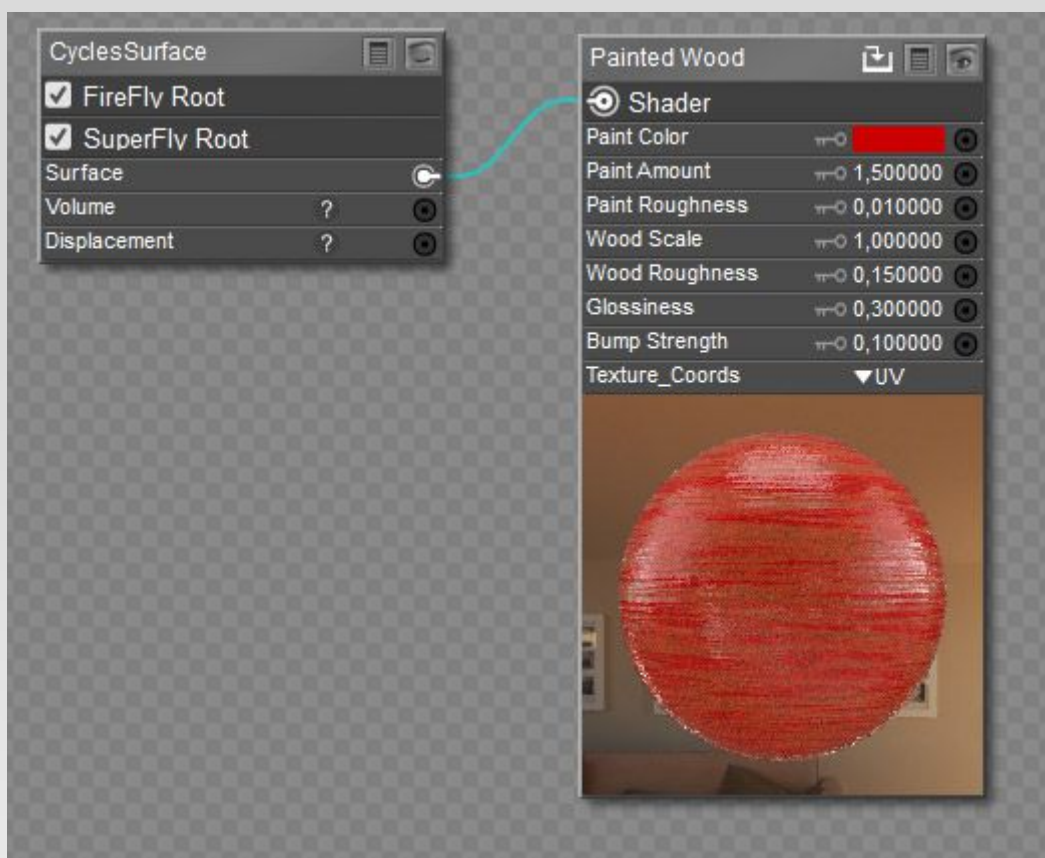
Plastic Scratched 02



Satin Bump



A mix of different Painted Wood shaders.



You have control over the paint color, amount and roughness, the wood scale and roughness, and the overall glossiness and bump strength. You can also plug a texture map into the paint color.



Notes

As these shaders use the Cycles Root, they are **only for SuperFly**. They won't work with FireFly.

Some shaders will take more time to render than others. This is due to the very nature of the shaders themselves, i.e. shaders with glass, translucency or SSS need more time to get rid of the noise while rendering.

Shaders are very dependent of the scene lighting. In fact, to get best results, you should always have something to reflect, i.e. some props around your scene, or easier : a HDRI on an environment sphere surrounding your scene, or plugged into Poser Background ticking the Cast_Light box. It will then be used for lighting.

In any case, **reflective shaders need something to reflect.**

Don't hesitate to mix shaders using the MixClosure node. You will get interesting results! ^^

The most important thing is : **HAVE FUN !!!**

Special thanks to Dead Reckoning for his precious help and advice, and for having suggested some of the SuperShaders. Thank you Dead! ^^