Contents

Introduction

Its all about...Light!

Blender

Blender is a fantastic, free, open-source 3D modelling package that enables you to create games, animations, architectural visualisations, physics animation and much, much more!

Blender delivers out-of-the-box its internal rendering engine, which will be replaced by the already released (since 2.61) Cycles engine. It is an unbiased engine directly built into Blender as an addon and uses path tracing. It is still under development.

Indigo

So why consider Indigo?
Indigo is an unbiased, physically based and photorealistic renderer which simulates the physics of light.

It boasts a physical camera model, a great materials system and realistically computes complex lighting situations.

Realism, to put it all in one word.

This tutorial was written to help understand how Blender and Indigo work together to create photorealistic renderings.

Prerequisites

A basic knowledge of Blender is a must.
Although, when possible, this tutorial will give a short explanation of the how and whats of working with Blender, you will need a good understanding of Modelling (scaling, extruding, normals,...), Lighting (lamp types, light setups, lighting techniques,...), Texturing (UV-Unwrapping, Materials, Material types,...) and the like.

Please note:This is a tutorial for working with Blender and Indigo: it is -not- a *Blender tutorial*.

We will cover some of these important steps, but please take this hint straight from the beginning, as it will save a lot of frustration and confusion later on.

Lets get started!

Indigo is a stand-alone application (it can be used in a network: a master and n-slaves can share the work, but this wont be covered here). It has a Windows, Apple and Linux version in 32 and 64bit versions.

This means the actual work is done -outside- of Blender: a addon has been developed that seamlessly integrates into Blender when working with your scenes and models, and then can prepare and export right into the format Indigo needs to render your project.

Installing Indigo

The Addon, which is included in the Indigo setup, can be downloaded at www.indigorenderer.com/blender.

Instructions on how to install Indigo and the Blender Exporter can be found at:

Camera!

Materials!

An introduction

Materials are used to describe an objects surface, or the object as a whole. Light reacts very differently on different materials: a tablecloth reflects / absorbs light in a different way than a polished wood table. Materials like glass, wood, metal all have characteristics that can be stored in a material to be used on a 3D object.

The material library is a great place to have a look at already existing materials and how they "tick". Simple, colored materials may only contain attributes defined in the ISL Language of Indigo, others are packed IGM or PIGM files (which are a set of compressed files) that include the textures or texture maps needed for that material (more to that later). The IGM or PIGM files can be opened with a compressor /decompressing program like 7Zip.

Material types

Different material types are defined to describe the kind of characterstics an object should have. These are:

Diffuse
Phong
Specular
Oren-Nayar
Glossy Transparent
Diffuse Transmitter
Blend
Exit Portal
Null

Material Attributes

Each material type has certain attributes that can be assigned to it. The different material attributes that Indigo uses are described here:

www.indigorenderer.com/documentation/manual/indigo-scenes/materials.

Here is a list of material attributes:
Albedo
Bump
Displacement
Exponent
Base Emission
Emission
Absorption Layer Transmittance
Layer

Albedo

Albedo is a basic "color" material. If you set an objects color to RGB colors 0.0 for all colors, it will be black, any other combination gives different colors that are reflected of that surface.

Bump

Bump mapping lets you create an object in Blender and then use a material that gives that object a "look" as if it had more complicated geometry. (have a look at en.wikipedia.org/wiki/Bump_mapping).
Lets say you create a plain sphere in Blender. You can create a black and white image (the bump map) that has a "noisy" array of black dots on a white background. You can draw a larger, black dot in the center.
When using this bump map on your sphere (in the materials settings), your sphere will be rendered and it will use this bump map to create the illusion of real sophisticated geometry (lots of dimples and depressions on the surface).

Displacement

A Displacement map is a texture that actually effects geometry when applied to an object. It uses a texture or height map (it interprets the white and black areas of the texture as high or low points) that then changes the surface of your object.

Exponent

As described in the Indigo Manual, the exponent controls the roughness of a surface (makes it dull or shiny).

Base Emission

Giving Your object this material lets it emit light (in units of Lumens).

Emission

This parameter is used to brightness varies over the surface of your object.

Absorption Layer Transmittance

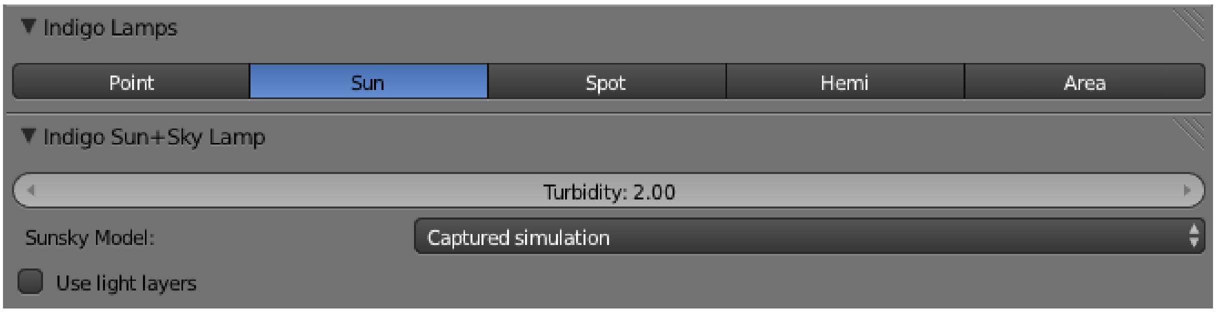
This can be a bit difficult: this defines how a specific layer absorbs light, without changing the medium s (your object) specular (shiny) absorption proberties (like a stained glass window: the "outside" layer of the glass has a certain reflectivity or absorption, a different layer inside the glass having other absorptioin properties).^

Layer

Layers, as the name says, lets you split the rendered image into layers, and being able to control how much each layer individually influences the whole image. Useful when using light layers.

Lights!

Only two types of lights (if you exempt materials emitting light) are supported with Blender in combination with Indigo: **Sun** and **Hemi**.


▼ Indigo Lamps

▼ Indigo Sun+Sky Lamp

Turbidity: 2.00

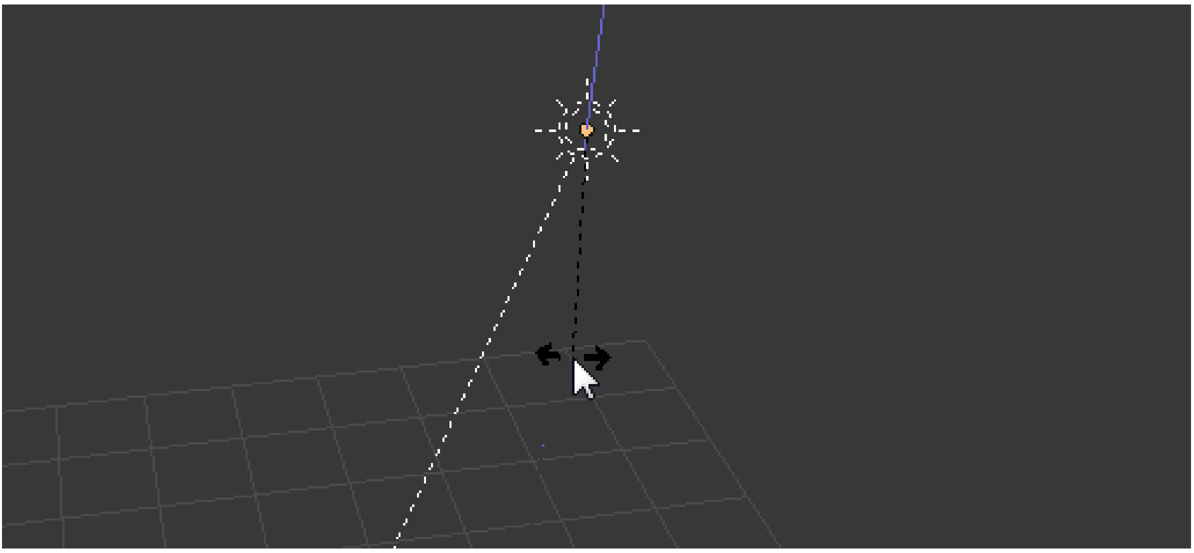
Sunsky Model: Captured simulation

□ Use light layers

By clicking on the light object you can switch between the two types in the lamp properties.

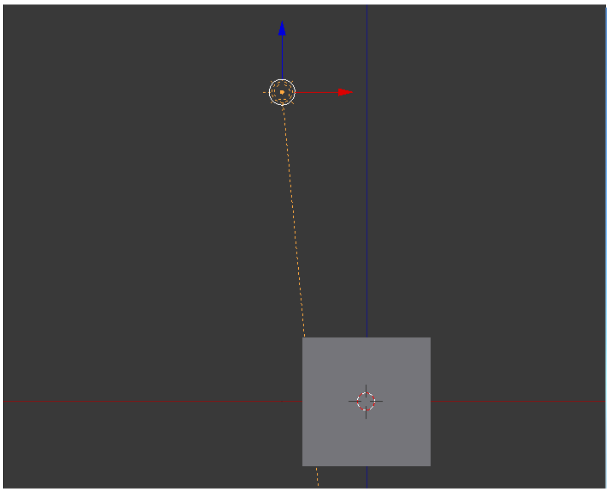
Sun

When using the sun lamp, you can influence the lights direction by changing the lamps orientation.

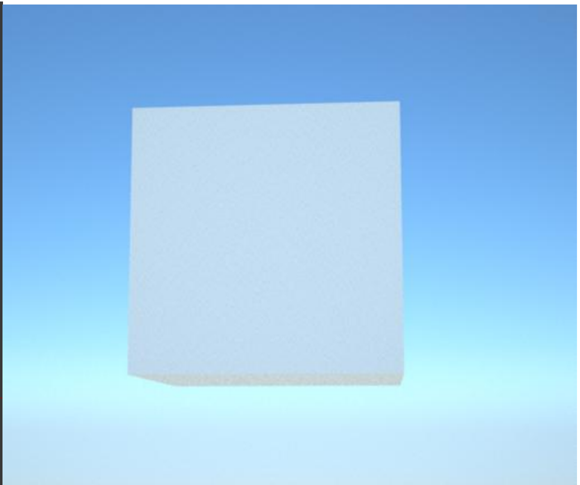


To change the “time of day”, so to say, the angle of the sun is important. Create a very steep sun, and you will have a scene as during midday.

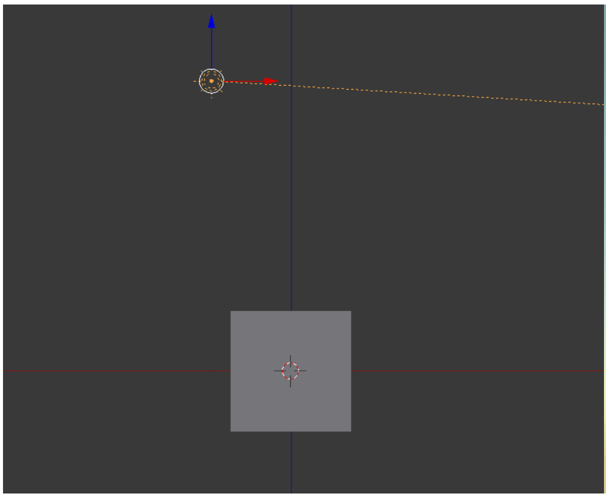
Here we have some 2 examples:



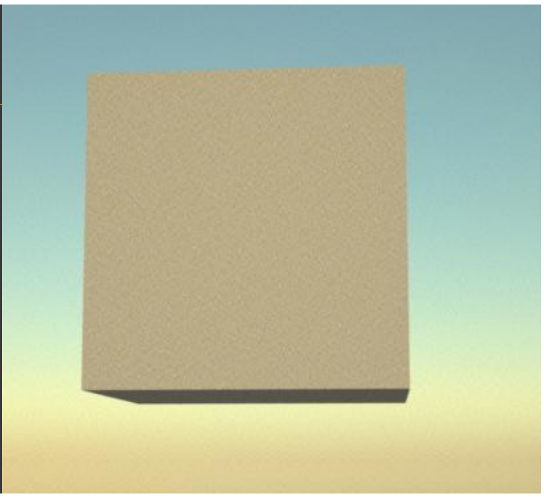
“High Noon”



The Render



„Evening Sky“



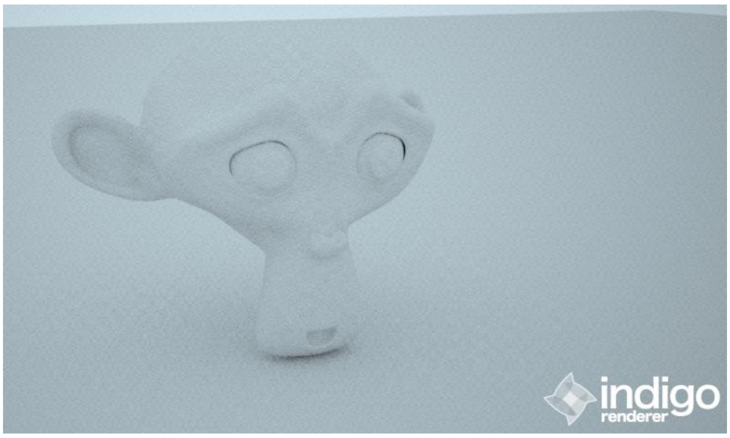
The Render

Hemi

The Hemi lamp has various settings.

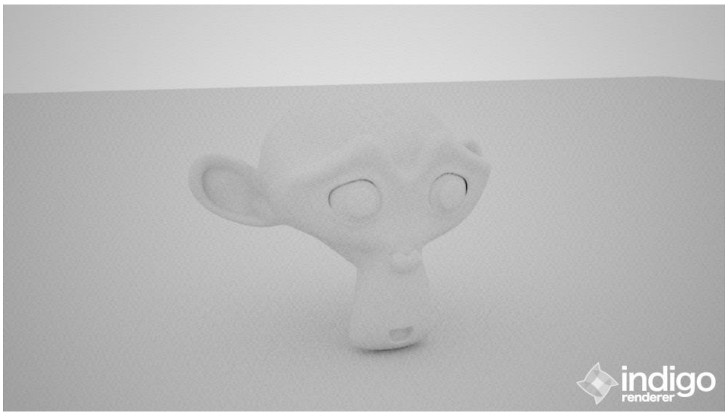
Blackbody

This gives off light based on the temperature. Measured in Kelvin.



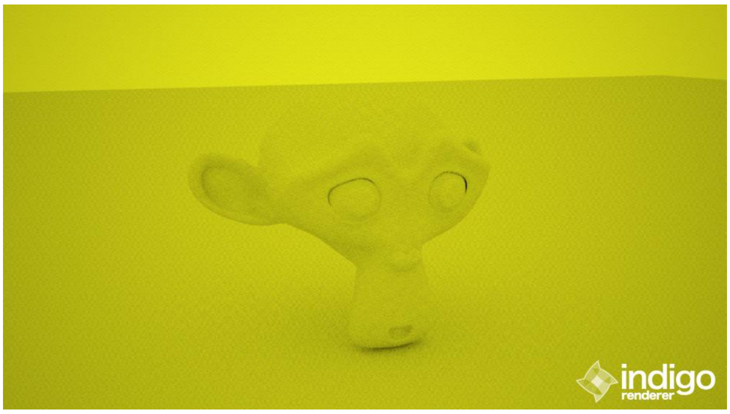
Uniform

A white light with intensity based on value given.



RGB

The color of the light can be chosen via color picker



Mesh Emitters

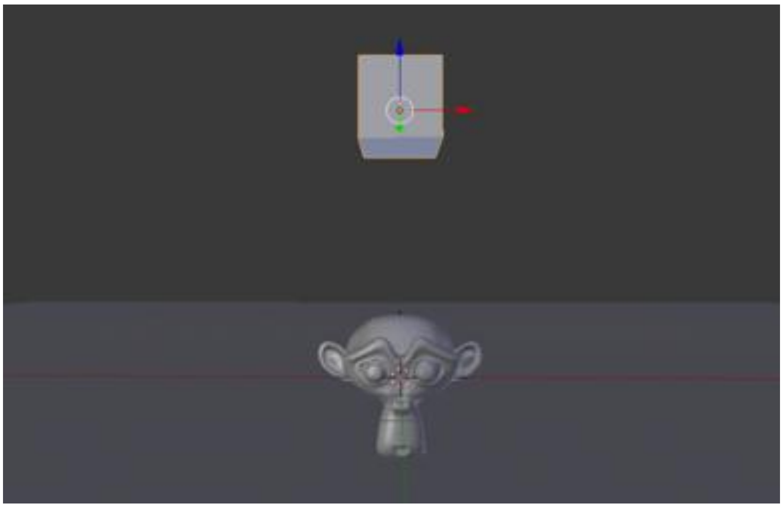
Another type of using light in Blender and Indigo is to use a mesh and use it to distribute it.

In this case, we would use the “emit” property of its material.

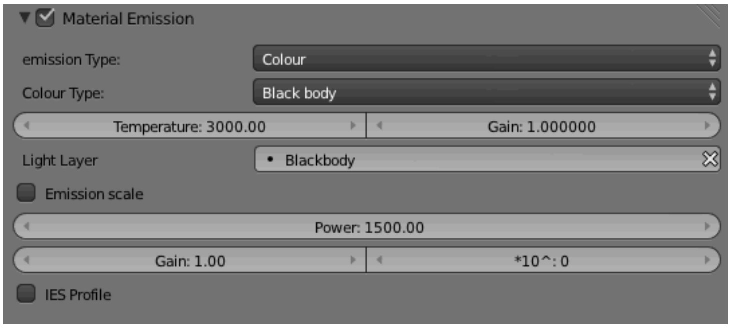
Emission Types

Black Body

In this scenario, a cube was added as a “blackbody” emitter.



This cube received a basic diffuse material with a “Black Body” Color type (this gives off Light based on the temperature. Measured in Kelvin) in the “Material Emissions” properties:



▼ ☑ Material Emission

emission Type: Colour

Colour Type: Black body

Temperature: 3000.00 | Gain: 1.000000

Light Layer • Blackbody

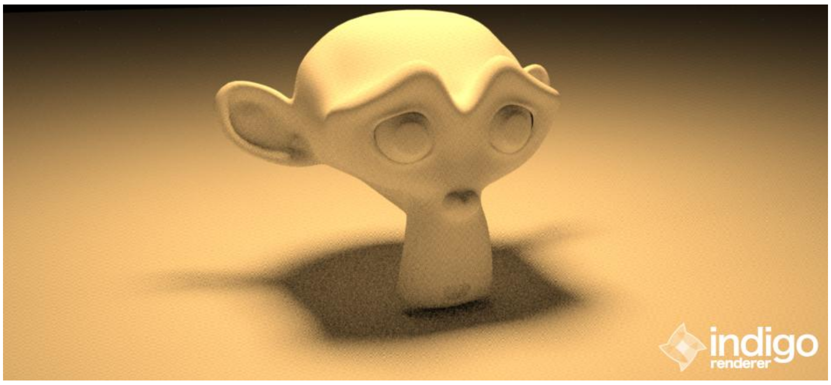
□ Emission scale

Power: 1500.00

Gain: 1.00 | *10^: 0

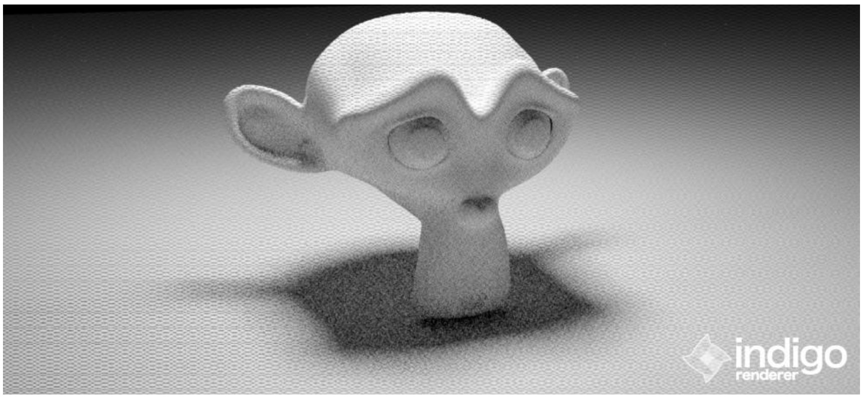
□ IES Profile

The result:



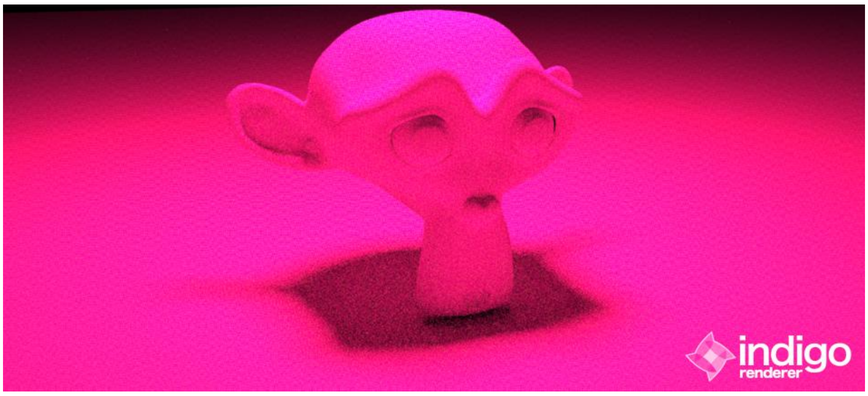
Uniform

In this scenario, the same cube has a basic diffuse material with a "Uniform" Color type (A white light with intensity based on value given) in the "Material Emissions" properties:



RGB

In this scenario, the same cube has a basic diffuse material with a "RGB" Color type (the color of the light can be chosen via color picker) in the "Material Emissions" properties:



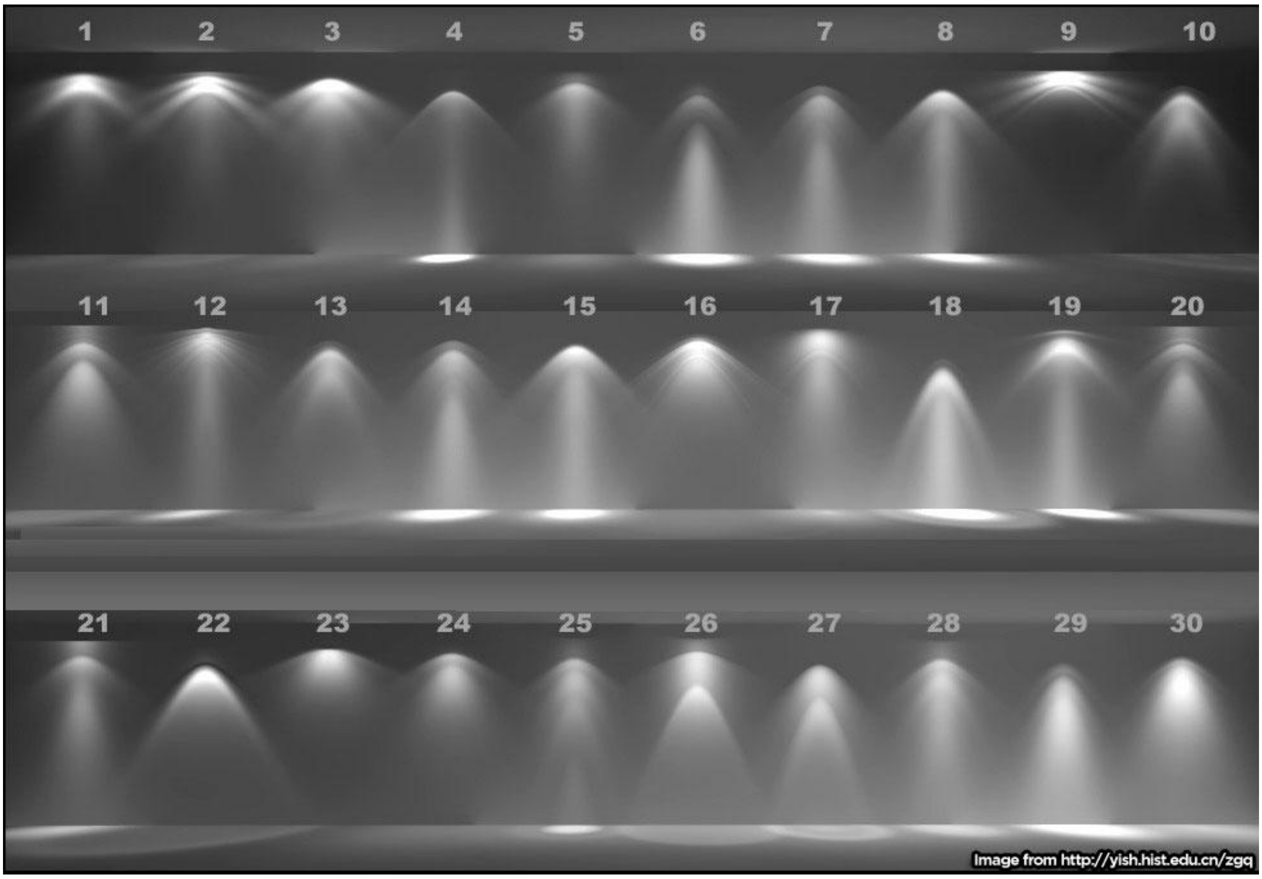
IES Profiles

IES stands for the Illuminating Engineering Society, which has defined a file format for describing the distribution of light from a light source.

Using only a small, simple emitter such as a single quad, an IES profile will shape the distribution of light emitted from it to match that of a much more complicated light fixture.

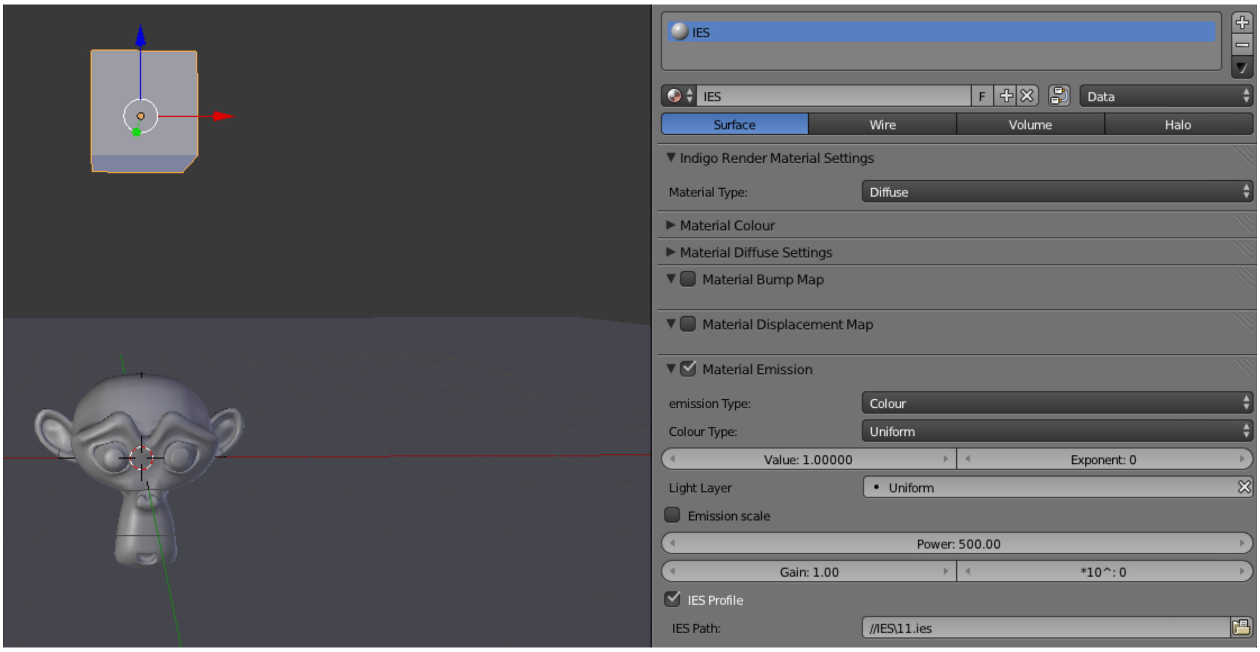
While Indigo is capable of creating real refractions of an accurately modelled light fixture to create this effect, it is far easier to use an IES profile, and the result is much the same. Many manufactures provide IES files for their lights, and it is a great way to add realism to your scene. [1]

IES Profiles overview



In this example, we have a mesh with a emitting material. This material has an IES Profile assigned to it:

[1] http://www.indigorenderer.com/documentation/manual/techniques/ies-lights



Indigo Render Material Settings

Material Type: Diffuse

Material Colour

Material Diffuse Settings

☐ Material Bump Map

☐ Material Displacement Map

☑ Material Emission

emission Type: Colour

Colour Type: Uniform

Value: 1.00000 Exponent: 0

Light Layer • Uniform

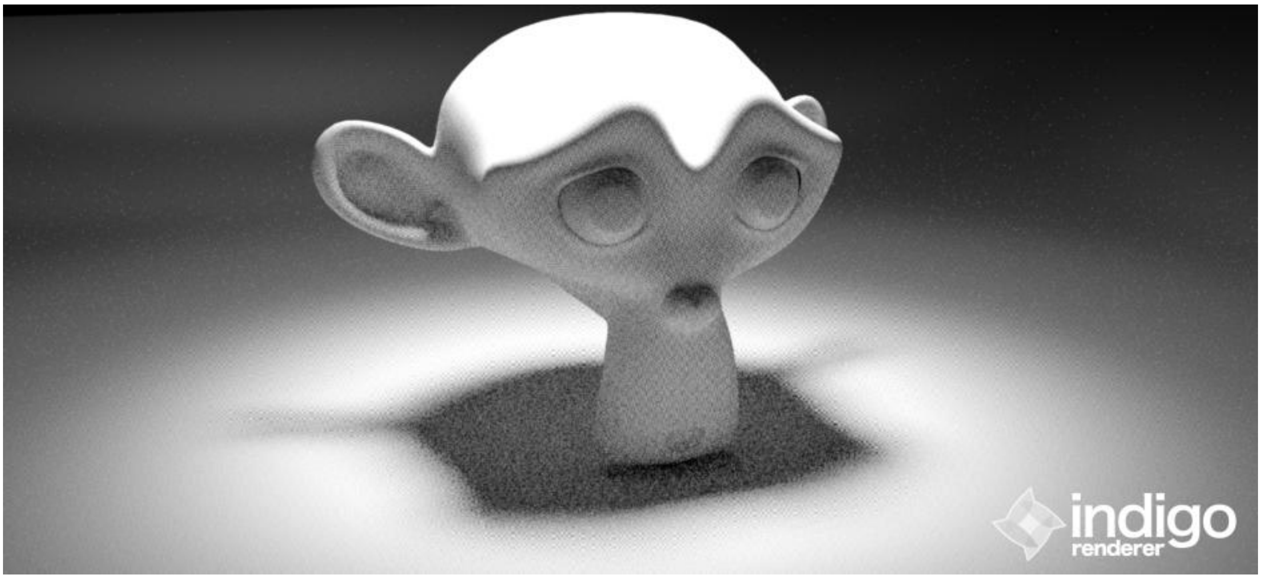
☐ Emission scale

Power: 500.00

Gain: 1.00 *10^: 0

☑ IES Profile

IES Path: //IES\11.ies



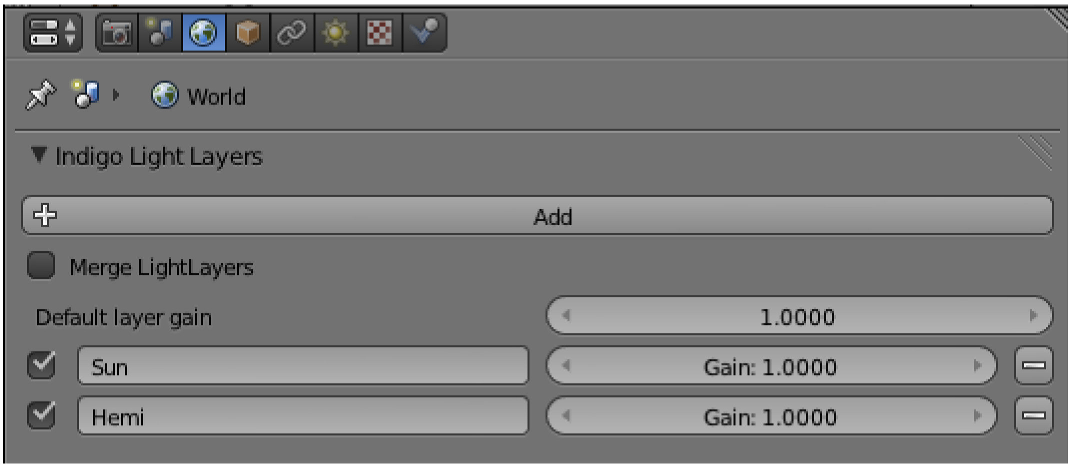
As you can see, the IES Profile shows its specific light settings.

Light Layers

You can create light layers in Blender for use with Rendering with Indigo.

This is very practical: you can set up various light setups, assign each light to a certain layer and when you are ready to render, you can either “mix” these layers together and adjust the influence of each one, or create completely different lighting scenarios (a room with sunshine from a sun on the outside and the room only lit by lamps on the inside) and tell Blender / Indigo which light layer to use.

In the World options of Blender is where the light layers are created. You can add or delete layers here, and adjust the influence of each layer.



World

▼ Indigo Light Layers

Add

□ Merge LightLayers

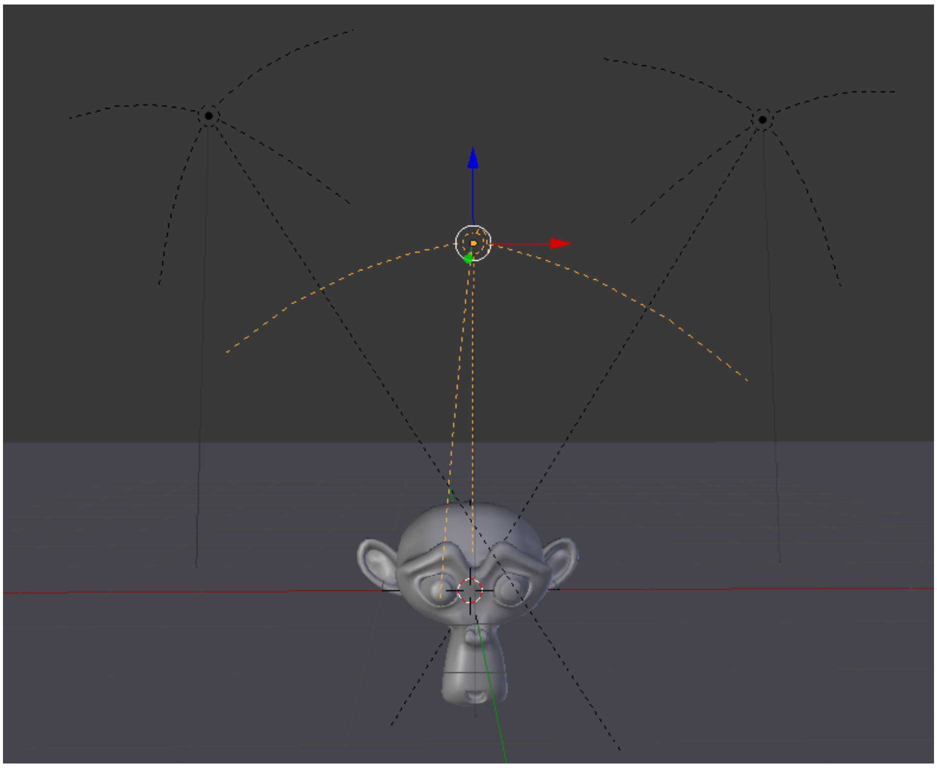
Default layer gain: 1.0000

☑ Sun — Gain: 1.0000

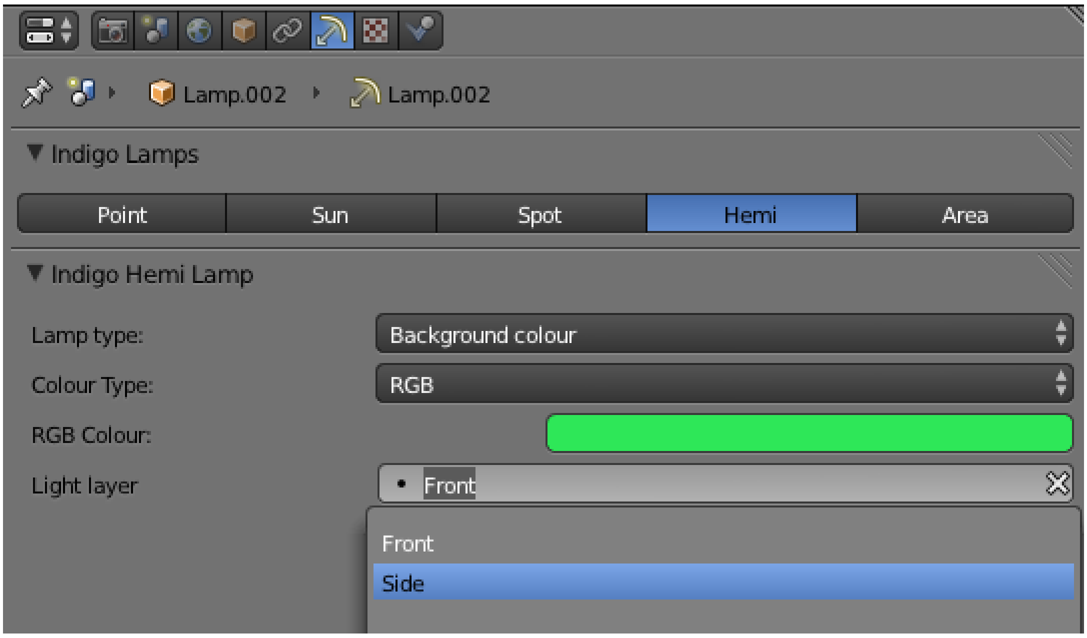
☑ Hemi — Gain: 1.0000

Case in point

In this scenario, we have 3 lamps: a hemi lamp in front, and two hemi lamps left and right:



When selecting a lamp, you can assign it to a light layer that you added in the world panel.



Lamp.002 ▸ Lamp.002

▼ Indigo Lamps

▼ Indigo Hemi Lamp

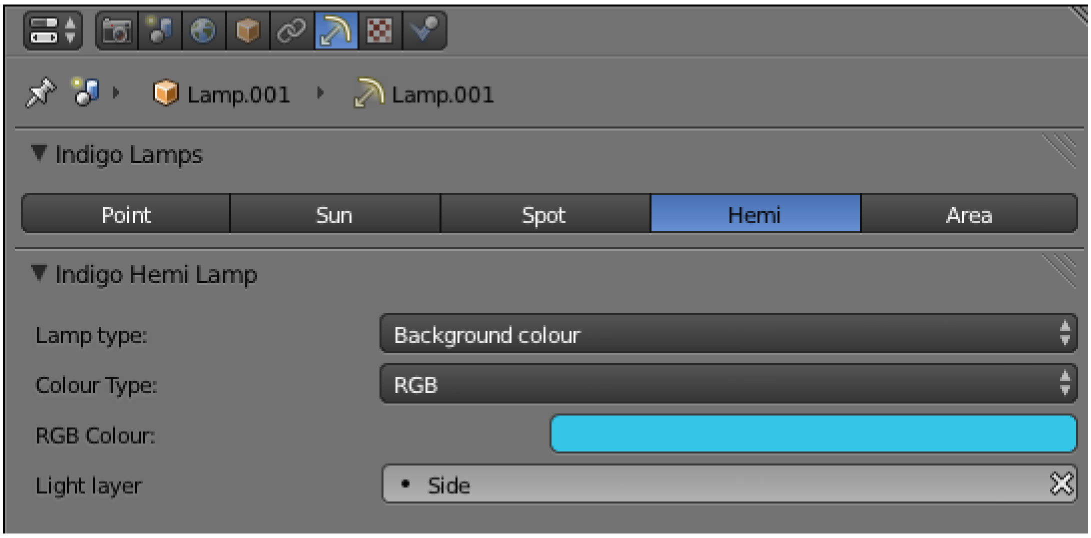
Lamp type: Background colour

Colour Type: RGB

RGB Colour: 🟩

Light layer • Front

Picture 1 - Assigning the Front Hemi Lamp the "Front" Light Layer



Lamp.001 ▸ Lamp.001

▼ Indigo Lamps

▼ Indigo Hemi Lamp

Lamp type: Background colour

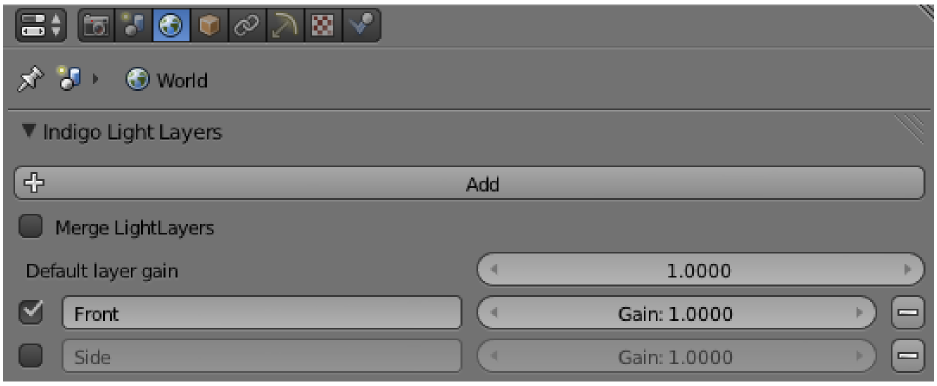
Colour Type: RGB

RGB Colour: 🟦

Light layer • Side

Picture 2 - Assigning the Left Hemi Lamp to the "Side" Light Layer

If we want to render only a certain light layer, deactivate the other layer(s) in the world parameters:



World

▼ Indigo Light Layers

□ Merge LightLayers

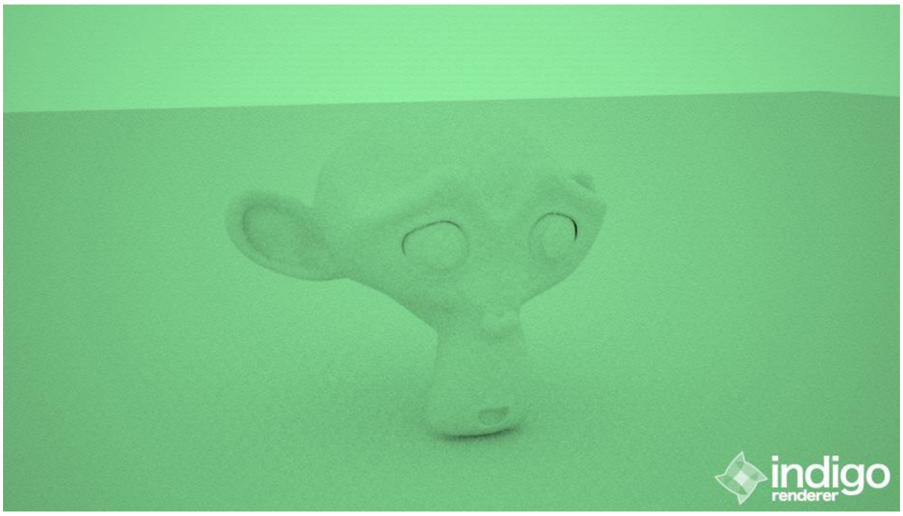
Default layer gain 1.0000

☑ Front Gain: 1.0000

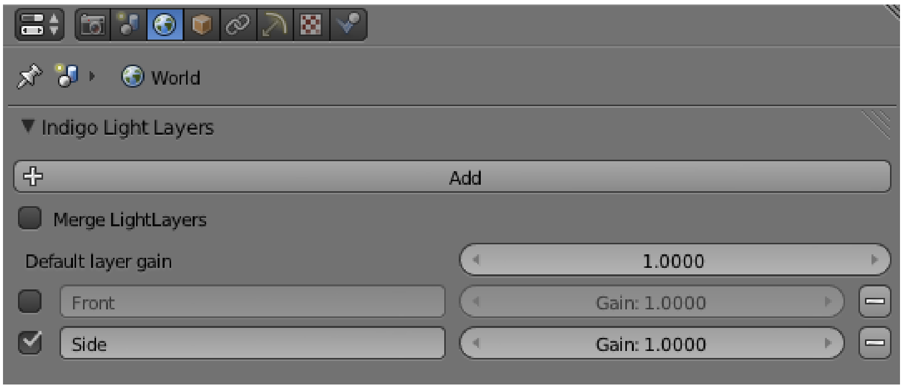
□ Side Gain: 1.0000

Our front lamp is green, so if we only check the “front” light layer in the world settings:

We get this:



Our side lamps are blue, so if we only check the side lights layer:


World

▼ Indigo Light Layers

Add

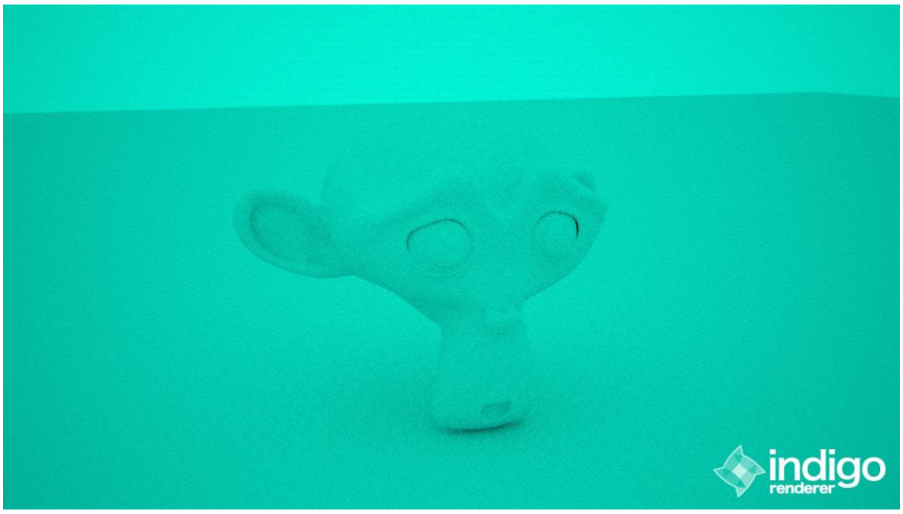
□ Merge LightLayers

Default layer gain 1.0000

□ Front Gain: 1.0000

☑ Side Gain: 1.0000

we get this:



Another use would be a night / day scenario, to be able to switch between the two light setups with no hassle.

indigo
renderer

Picture 3 - An example from indigorenderer.com

Exit Portals

Creating a simple room with Blender and Indigo
As you can see here, this is a simple room created in Blender and rendered in Indigo.

This room is a closed object: it has two windows with exit portals (more to that later) and one sun. The walls of the room are a cube (closed), with the two windows cut out witht the knife tool and then deleting those faces.

Each window, to increase render performance and "help" the computation of light have a so-call exit portal. This is a plane with the normals facing inward and the properties of this plane being an "exit portal".

Important: the room must have no "gaps"! The exit portal must face "inwards" towards the room.

Then position the sun as desired.

Basic Texturing

Here is a basic introduction to texturing with Blender and Indigo.

The steps were mentioned in the forum: here is an update with screenshots.

Viewport setup

In the following scene 3 areas are of interest (of course your scene can be set up as you please):

1. The default cube (here in edit mode)
2. Another window with the UV Editor Window
3. Our Texture properties.

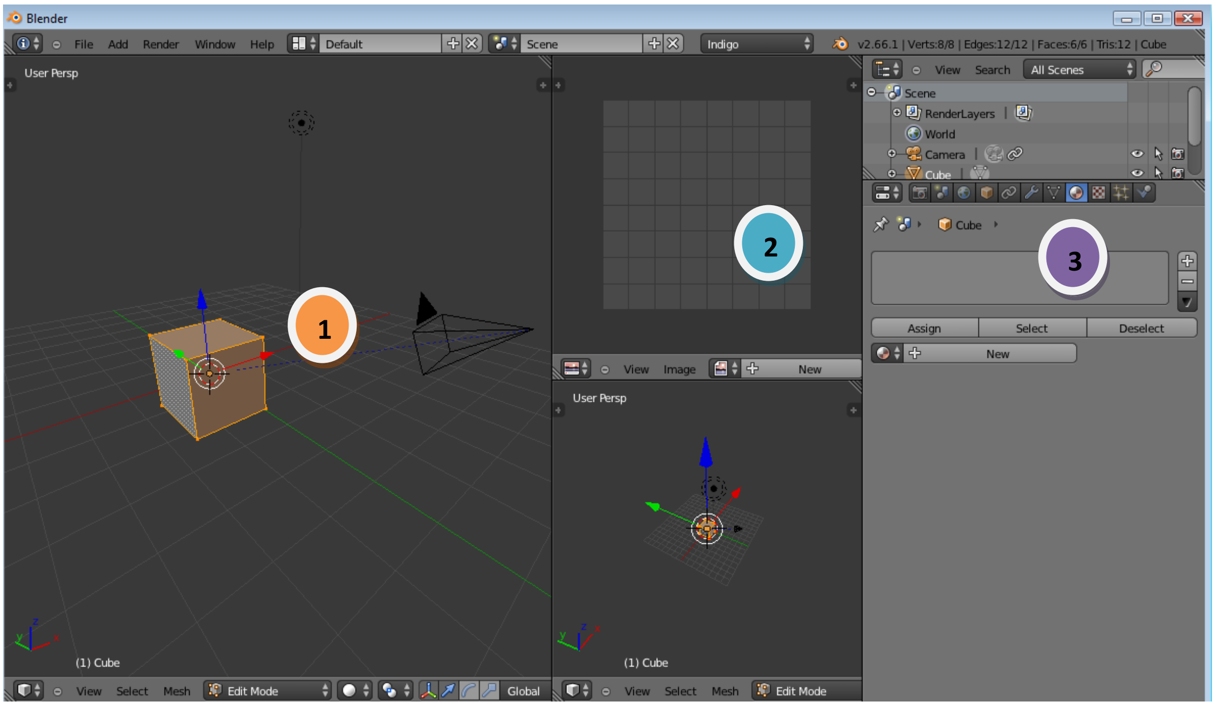
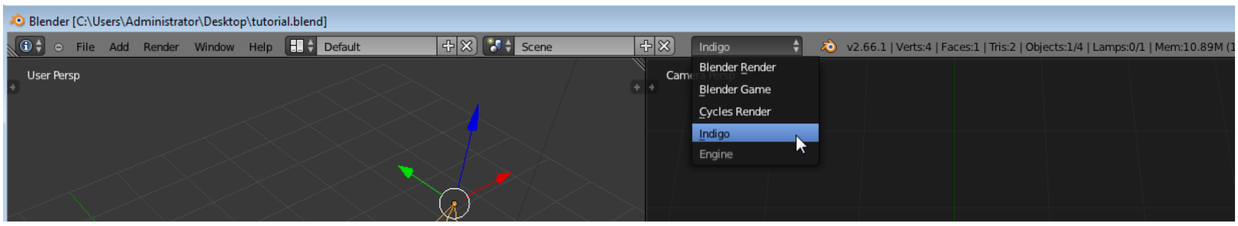


Figure 1 - A viewport setup with 3 main areas of interest

Setting Indigo as default Renderer

First, make sure that Indigo is set as default Renderer. In the Render dropdown box, select Indigo.



Blender [C:\Users\Administrator\Desktop\tutorial.blend]
File
Add
Render
Window
Help
Default
Scene
Indigo
v2.66.1 | Verts:4 | Faces:1 | Tris:2 | Objects:1/4 | Lamps:0/1 | Mem:10.89M (1
User Persp
Cam
Blender Render
Blender Game
Cycles Render
Indigo
Engine

Figure 2 - Selecting Indigo as Render Engine

The Wraps

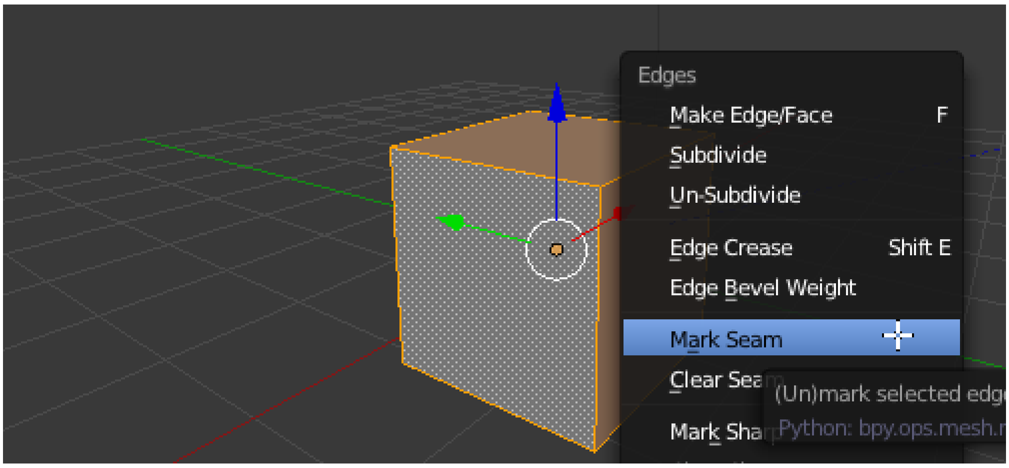
First of all, we want to be able to texture our object properly: to do that, we need to ***unwrap*** it.

Definition of unwrapping.

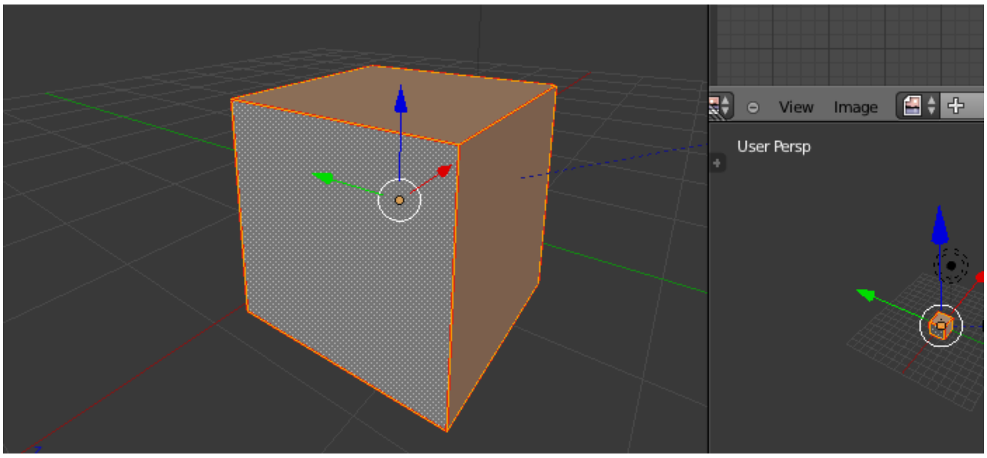
Why do we need to unwrap objects? To put it simple: Blender / Indigo needs to know how to "display" a texture onto our object. Unwrapping is essentially "unfolding" our 3D object onto a two-dimensional plane, and from there we can lay our texture on corresponding sides of our object. Imagine taking a cardboard box, cutting all necessary glued parts apart and laying it flat on the ground.

To help Blender unwrap our object the best way, we have to mark ***seams***. The seams on our figurative cardboard box are the parts that hold the different parts of the box together when folded back together.

1. To start, click on the default cube and hit TAB to enter edit mode.
2. Select the edges you want marked as seams and press CTRL+E and mark as seam:



If you have a look at the cube now, you will see that all edges you marked as seems are red:



3. Now you can press "A" to select your whole object and then press "U" and select "Unwrap" from the menu. Now you see the UV map on the right upper panel:

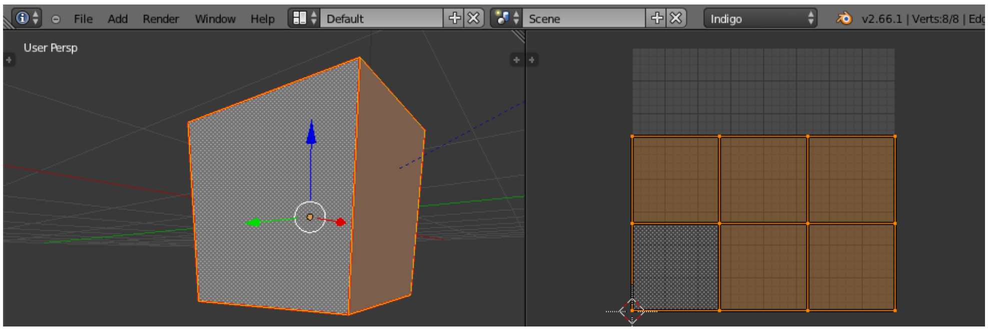
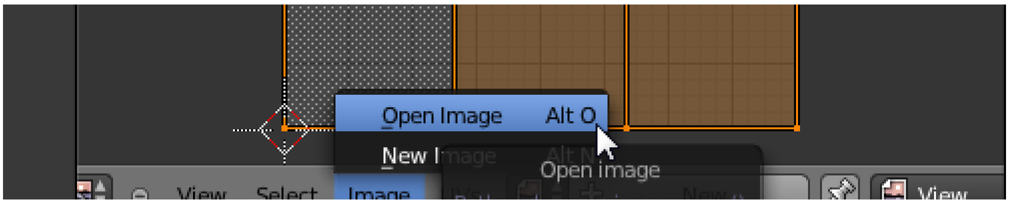


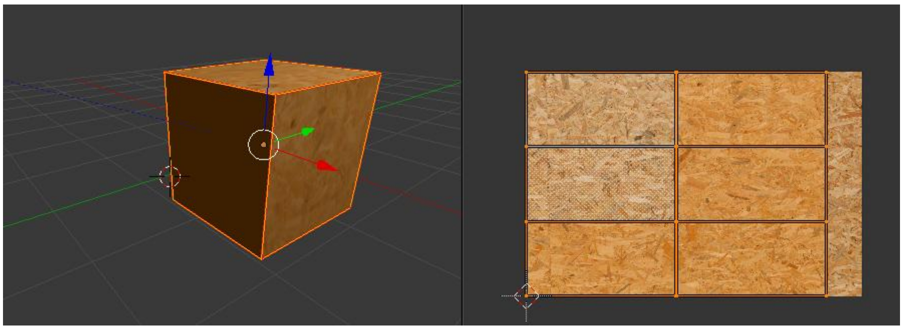
Figure 3 - All the "unwrapped" parts of our box.

Texturing the image

Now you can open an image (jpg, PGM, tif, etc.) with "ALT+O" or by using the GUI:

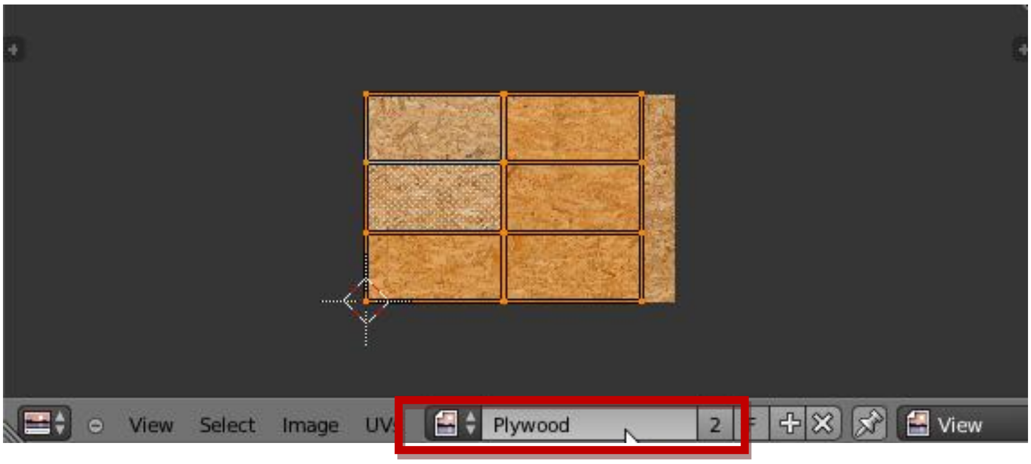


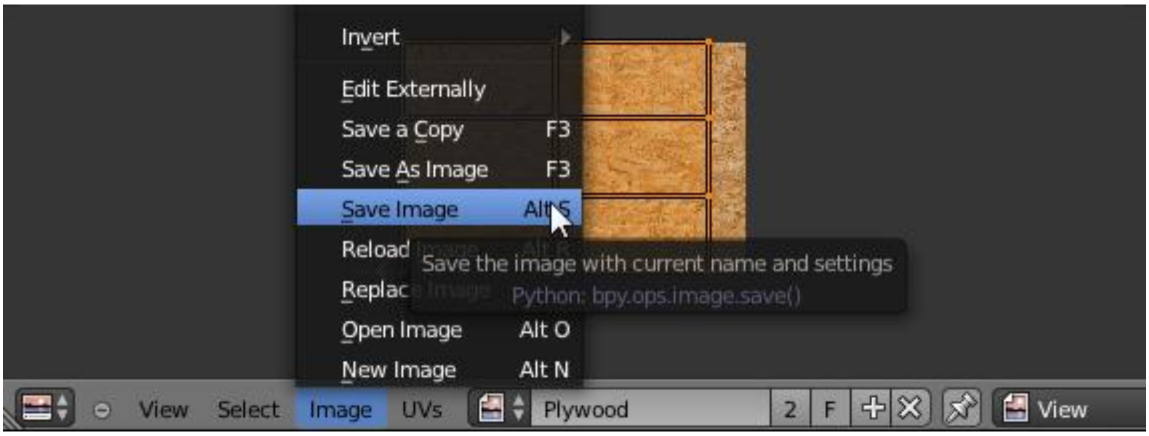
And once opened it will show up "under" your UV mapped mesh on the upper right in the UV Editor.



You can now move, stretch, scale your UV map to fit your needs.

Give the texture a meaningful name and save it (with ALT+S or use the User Interface), as we will need it later:





Save the image with current name and settings

Python: bpy.ops.image.save()

By the way: if your object doesn’t show up ***textured*** in the viewport:

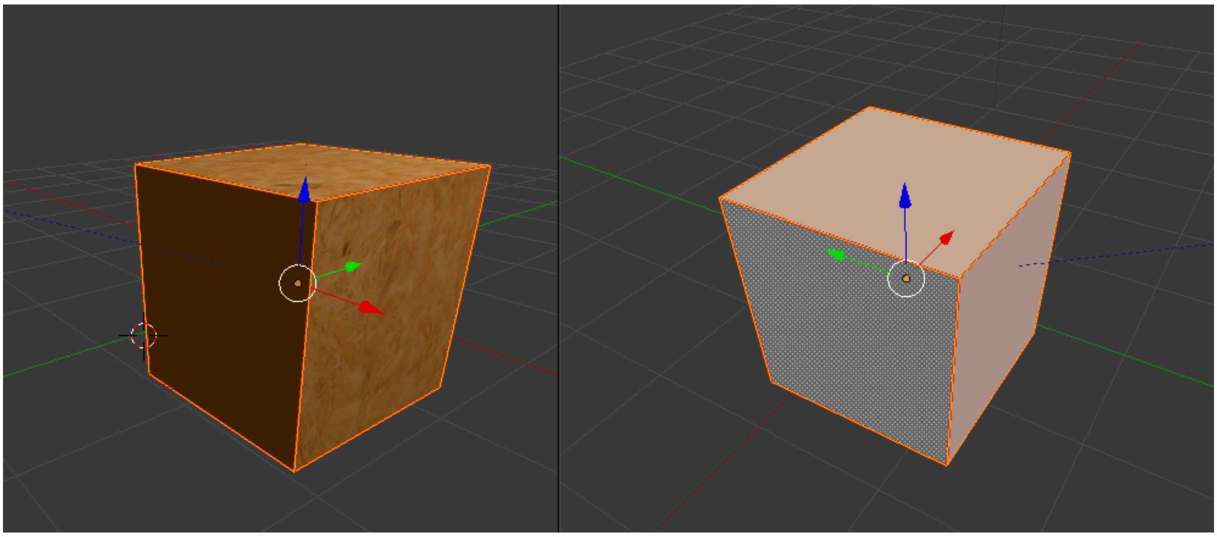
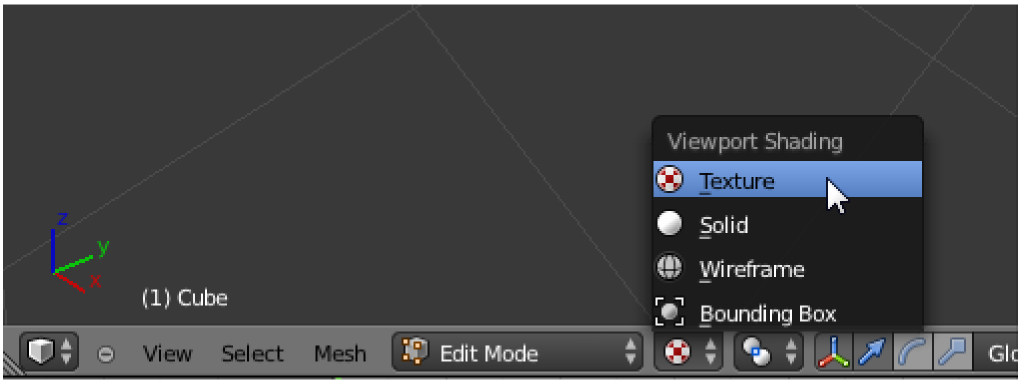
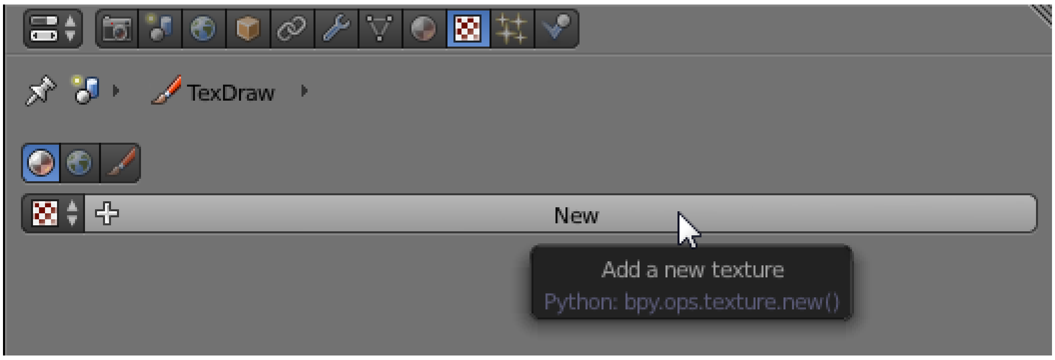


Figure 4 - Textured via Untextured in Blender Viewport

change the viewing mode to “Texture” or press “ALT+Z” in your viewport.

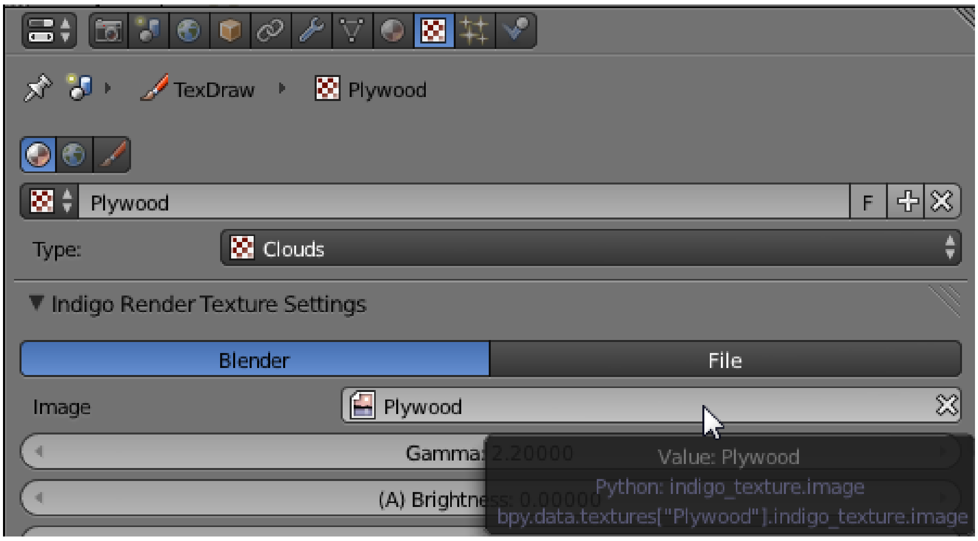


Now, in the "Properties" panel click on the "Texture" button. Create a new one by clicking on the "+" button.



Add a new texture
Python: bpy.ops.texture.new()

Click into the "image" selection field. The image that you used earlier in the UV editor should be presented to you for selection. Go ahead and select it. Give it a meaningful name and press ENTER. Otherwise, your settings might be lost once you leave the texture area.



Plywood

Type: Clouds

Image: Plywood

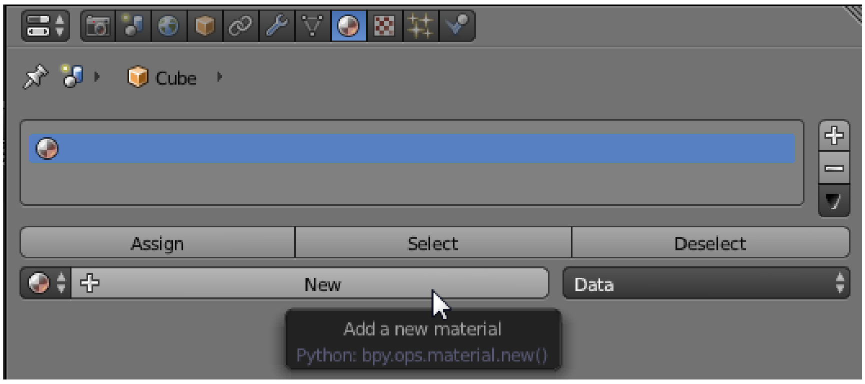
Gamma: 2.20000

(A) Brightness: 0.00000

Value: Plywood
Python: indigo_texture.image
bpy.data.textures["Plywood"].indigo_texture.image

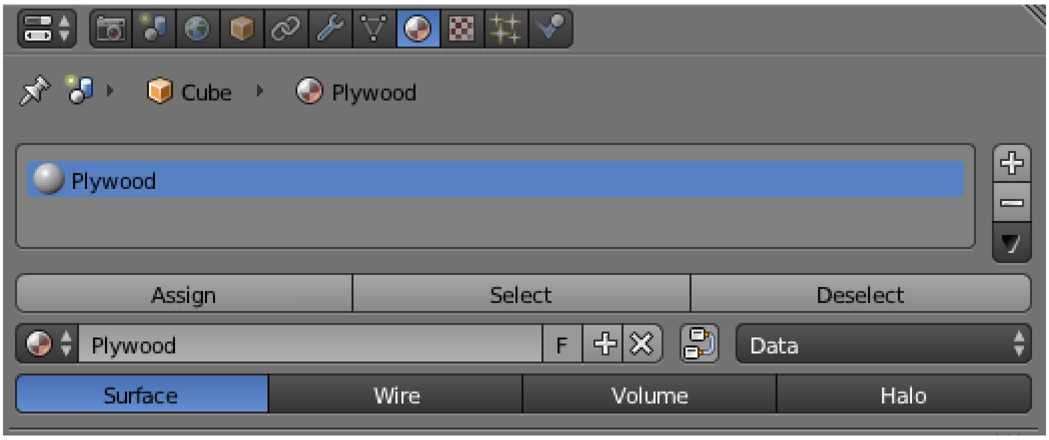
The Material

In the "Properties" panel click on the "Material" button. Add a new one.

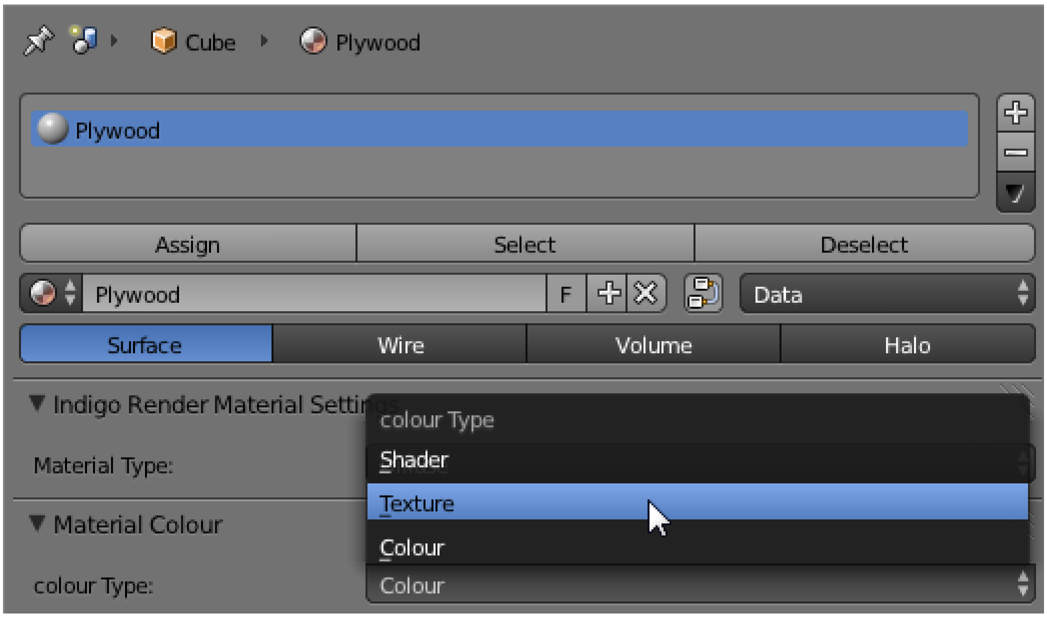


Add a new material
Python: bpy.ops.material.new()

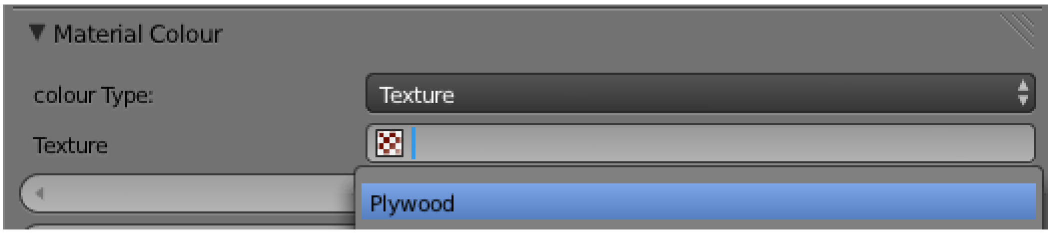
Change the default name of the texture to something meaningful



Change "Material Color" from "Color" to Texture".

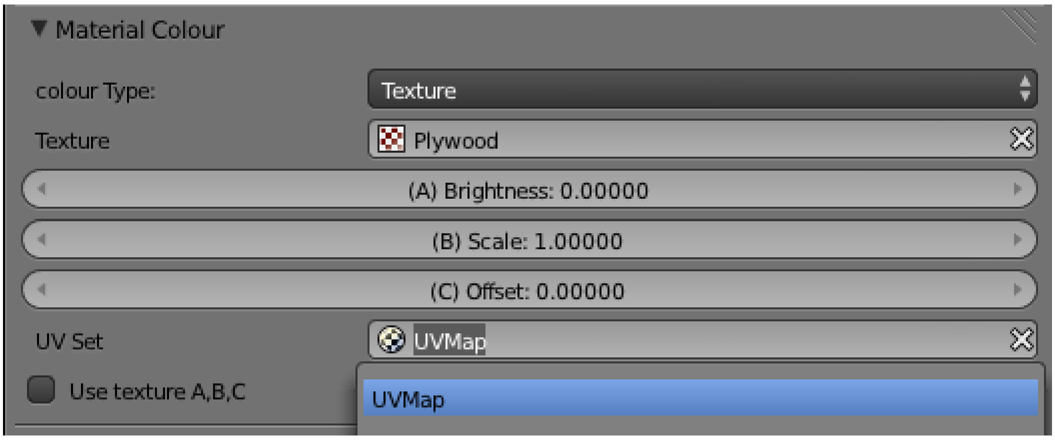


Click into the "Texture" selection field and you should be presented with at least the texture that you just created in the steps before. Select it.



Setting the UV Map

Select the UV Set we have created when unwrapping our object.



▼ Material Colour

colour Type: Texture

Texture: Plywood

(A) Brightness: 0.00000

(B) Scale: 1.00000

(C) Offset: 0.00000

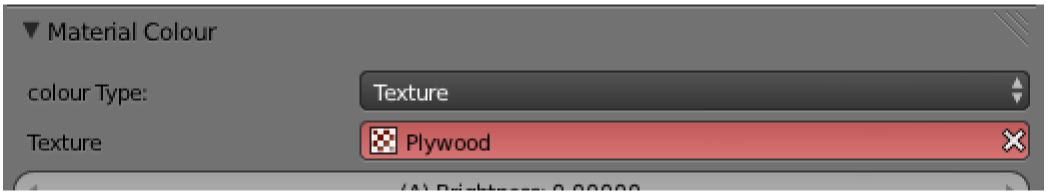
UV Set: UVMap

□ Use texture A,B,C

UVMap

Note: changing any names in the texture, material or uv areas will “break” any references and will have to be set anew in the other corresponding areas.

In the following example, we changed the name of our texture. The materials panel will then mark the error.



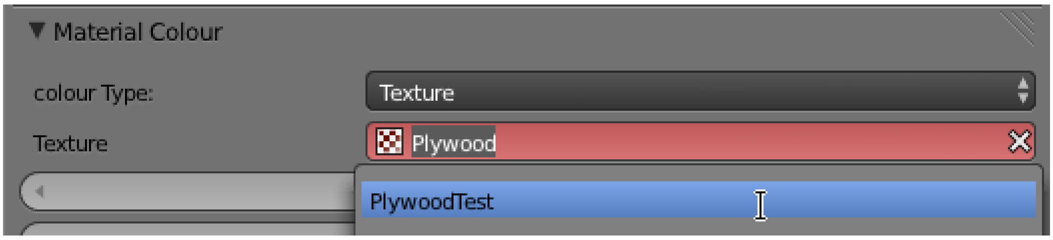
▼ Material Colour

colour Type: Texture

Texture: Plywood

(A) Brightness: 0.00000

Reselecting the right texture (click in the texture box) will correct this problem.



▼ Material Colour

colour Type: Texture

Texture: Plywood

PlywoodTest

SAVEYOURWORK. Blender can still crash unexpectedly from time to time!

Go into the "render" panel and click on"Image"

Hit F12 or render (make sure that Indigo is set as default renderer!) and now your cube should render with the applied texture.

Using a Material from the Indigo Online Library

www.indigorenderer.com has a large library of materials ready for use.
Check it out: http://www.indigorenderer.com/materials/

Here is a little preview of what materials in the library look like.
They are sorted in categories like Ceramics,Glass, Leather, Metals, Plastics, Stone, Textile, Vegetation, Wood and more.

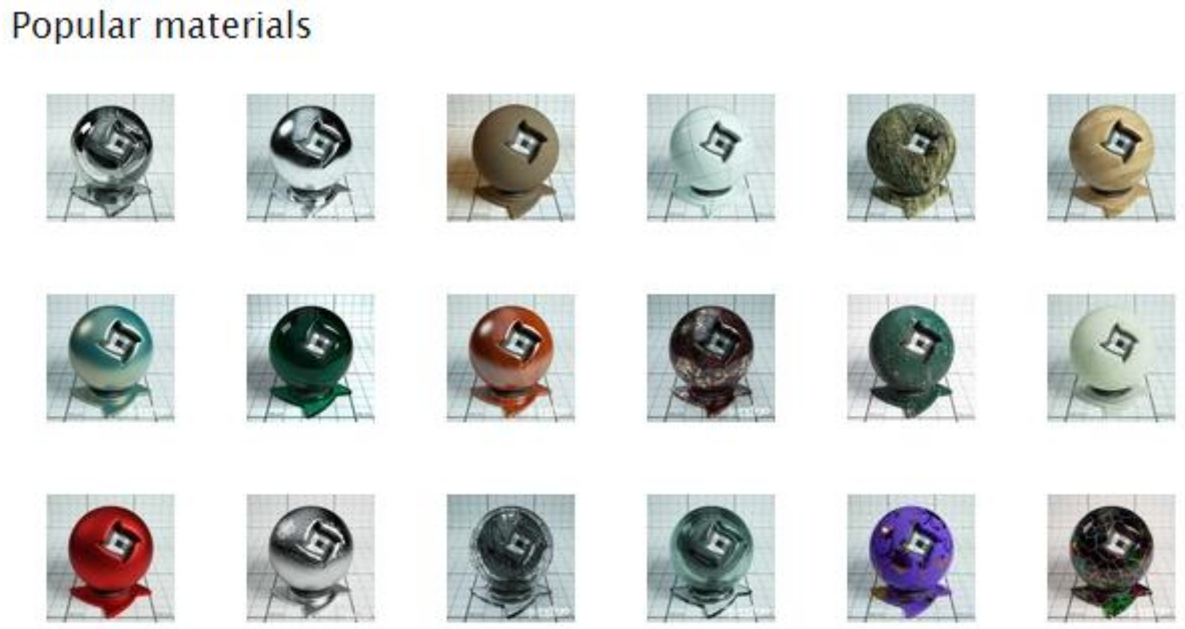
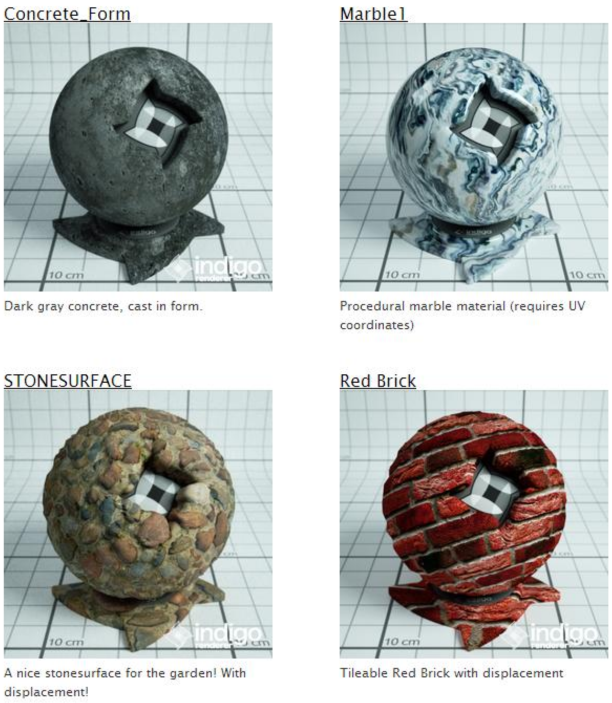

Popular materials

Figure 5 - A look at various materials from metal, plastic, wood and the like


Concrete_Form

Dark gray concrete, cast in form.

Marble1

Procedural marble material (requires UV coordinates)

STONESURFACE

A nice stonesurface for the garden! With displacement!

Red Brick

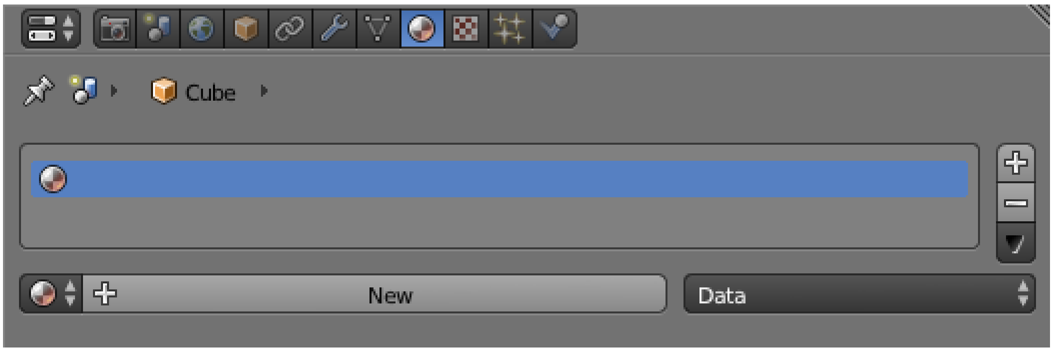
Tileable Red Brick with displacement

Figure 6 – example stone materials

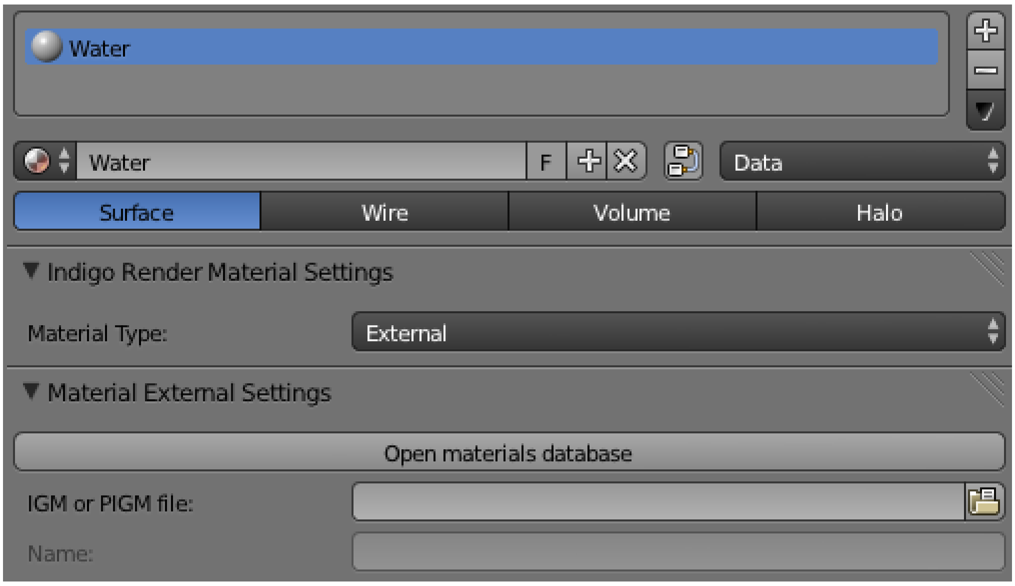
Using an external Material

You can use an external material in Blender with the following steps:

Create a new material in Blender for your object and give it a meaningful name.



Change the material type to “external” in the Material Type area.

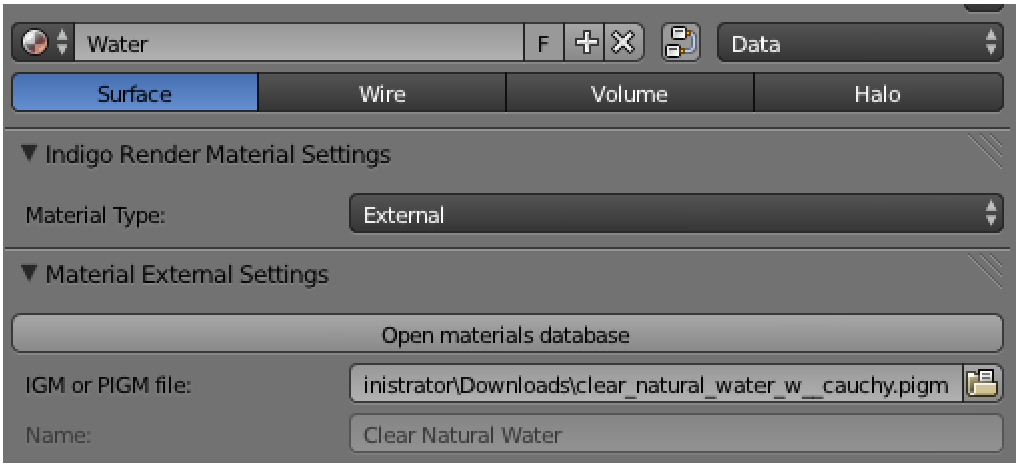


Material Type: External

IGM or PIGM file:

Name:

You can now either browse the online library for a material to download, or pick one already saved locally.



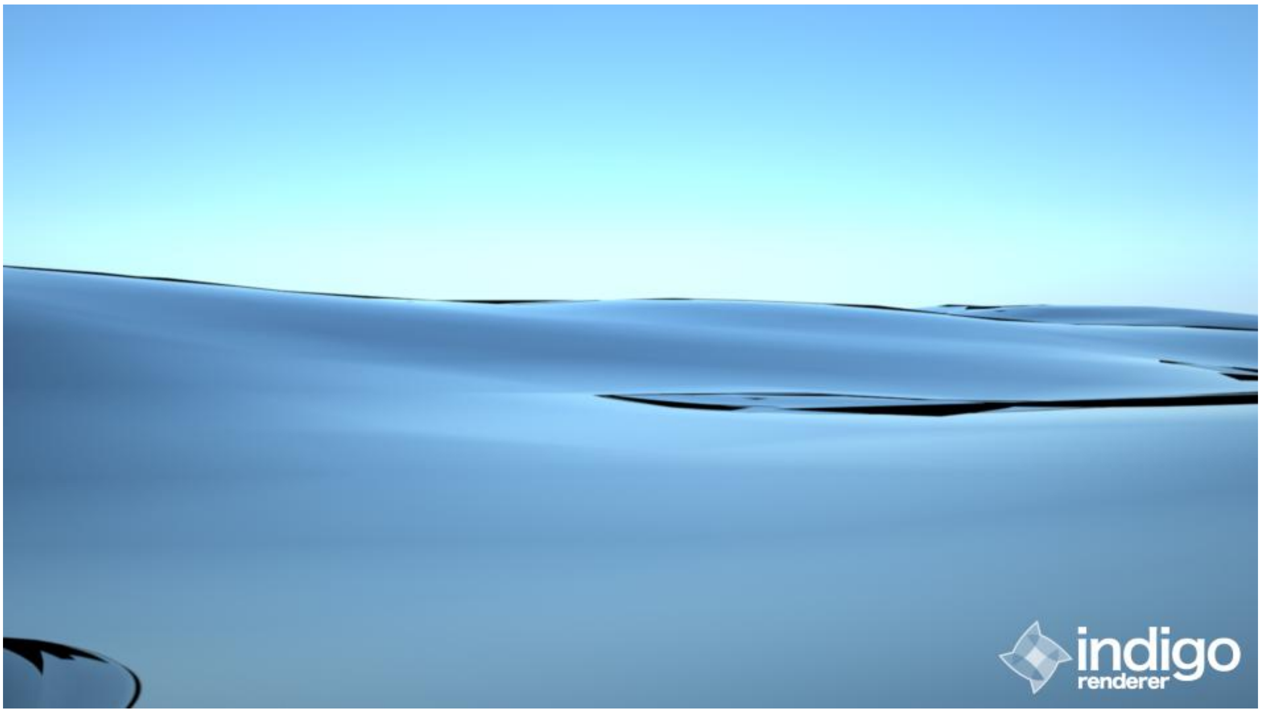
Water

Material Type: External

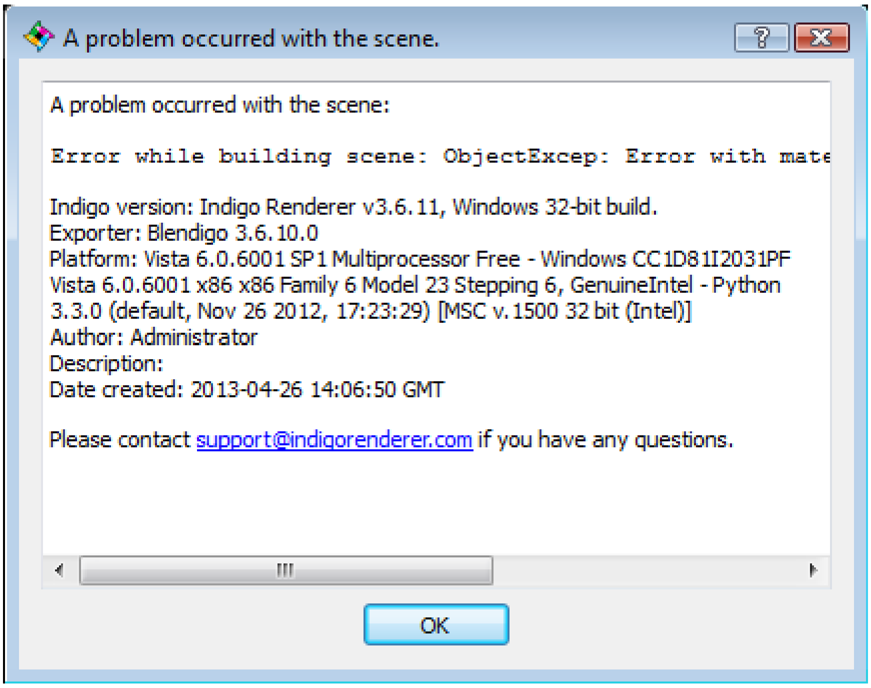
IGM or PIGM file: inistrator\Downloads\clear_natural_water_w__cauchy.pigm

Name: Clear Natural Water

Your object will then be rendered with this selected material.



Important notes: Make sure your object in Blender is UV unwrapped! If it isn't, then Indigo will present a similar error to this one:



A problem occurred with the scene:

```
Error while building scene: ObjectExcep: Error with mate
```

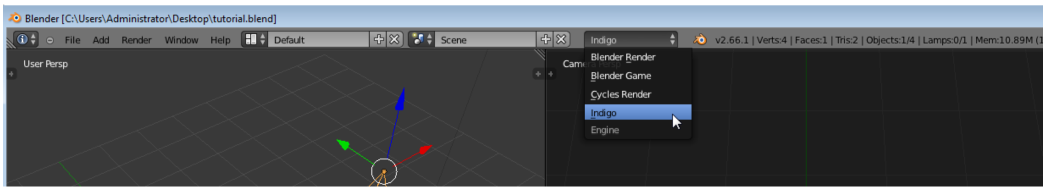
Indigo version: Indigo Renderer v3.6.11, Windows 32-bit build.
Exporter: Blendigo 3.6.10.0
Platform: Vista 6.0.6001 SP1 Multiprocessor Free - Windows CC1D81I2031PF Vista 6.0.6001 x86 x86 Family 6 Model 23 Stepping 6, GenuineIntel - Python 3.3.0 (default, Nov 26 2012, 17:23:29) [MSC v.1500 32 bit (Intel)]
Author: Administrator
Description:
Date created: 2013-04-26 14:06:50 GMT

Please contact support@indigorenderer.com if you have any questions.

Advanced Texturing in Blender and Indigo

In this part, we want to use different kind of texture maps in Blender for rendering in Indigo.

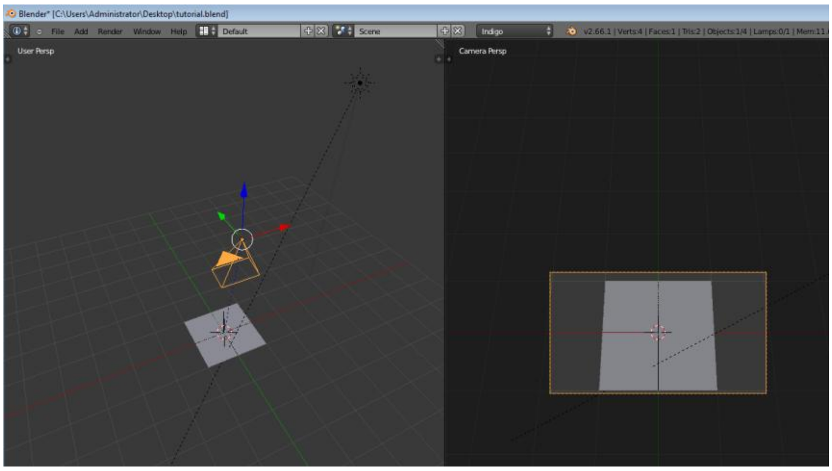
First, make sure that Indigo is set as Renderer.


Blender [C:\Users\Administrator\Desktop\tutorial.blend]
File
Add
Render
Window
Help
Default
Scene
Indigo
v2.66.1 | Verts:4 | Faces:1 | Tris:2 | Objects:1/4 | Lamps:0/1 | Mem:10.89M (1
User Persp
Blender Render
Blender Game
Cycles Render
Indigo
Engine

Picture 4 - Selecting Indigo as Render Engine

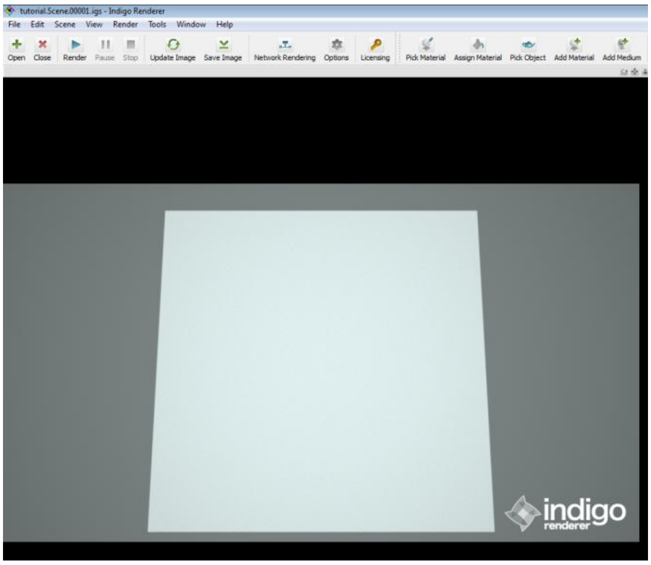
Sample Scene

For this example I have set a –very- simple setup for a texturing example. There is a plane, a sun and an empty, which the camera is parented to.


Blender* [C:\Users\Administrator\Desktop\tutorial.blend]
File
Add
Render
Window
Help
Default
Scene
Indigo
v2.66.1 | Verts:4 | Faces:1 | Tris:2 | Objects:1/4 | Lamps:0/1 | Mem:11.
User Persp
Camera Persp

Picture 5 - Simple overview of our setup

Now if we just hit „Render“ (F12) we will see our plane with no texture, just lit by our sun.



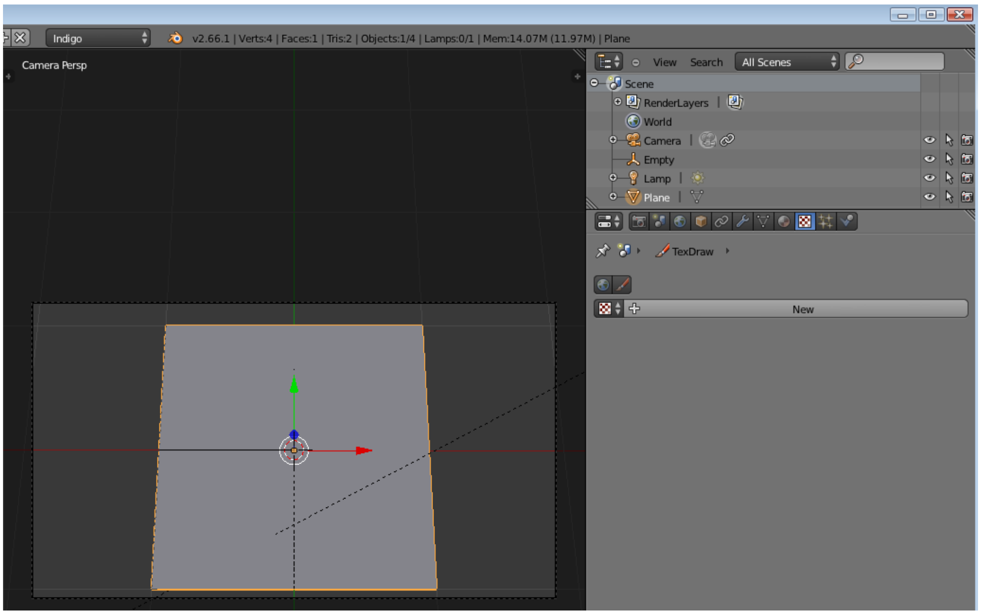
Picture 6 - The Render of our plane

Now, step by step we are going to texture this plane.

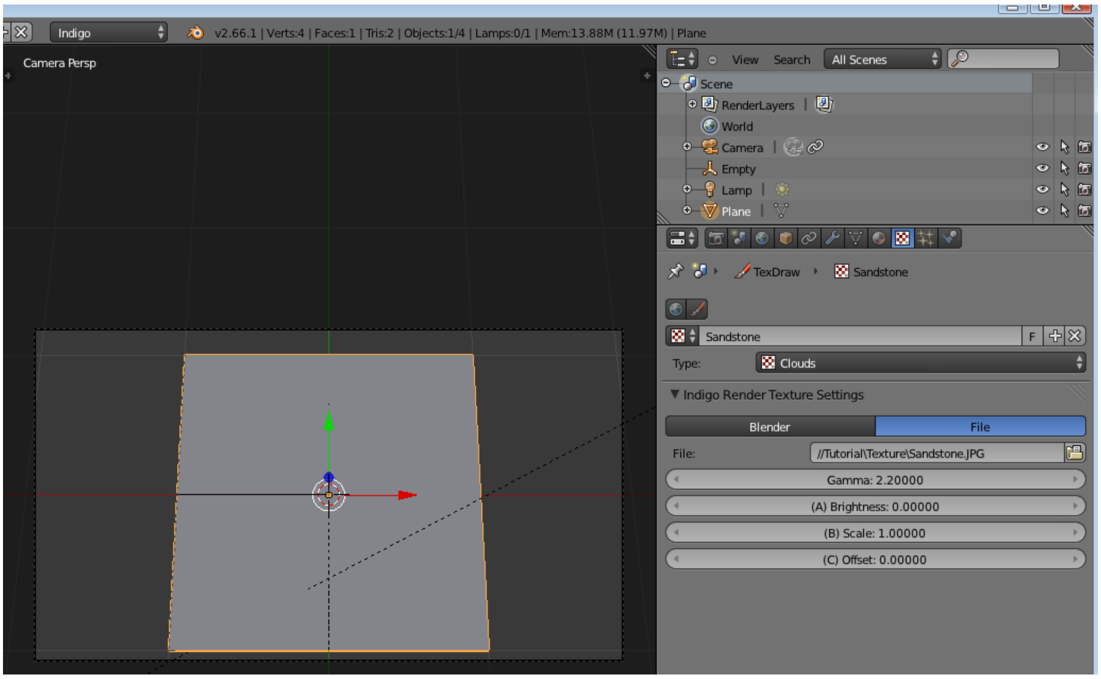
Texturing

Lets swivel our attention to the „Textures“ panel. We are going to add a texture.

1. Select the plane.
2. Select "New" on the textures panel.
3. Select "File", as we will select a file from the file browser and select the picture you want to use as a texture.
4. Give the Texture a meaningful name, as that will help us in the future.


Indigo
v2.66.1 | Verts:4 | Faces:1 | Tris:2 | Objects:1/4 | Lamps:0/1 | Mem:14.07M (11.97M) | Plane
Camera Persp
View
Search
All Scenes
Scene
RenderLayers
World
Camera
Empty
Lamp
Plane
TexDraw
New

Picture 7 - The empty Textures Panel


Indigo
v2.66.1 | Verts:4 | Faces:1 | Tris:2 | Objects:1/4 | Lamps:0/1 | Mem:13.88M (11.97M) | Plane
Camera Persp
View
Search
All Scenes
Scene
RenderLayers
World
Camera
Empty
Lamp
Plane
TexDraw
Sandstone
Sandstone
F
▼ Indigo Render Texture Settings
Blender
File

Type: Clouds

File: //Tutorial\Texture\Sandstone.JPG

Gamma: 2.20000

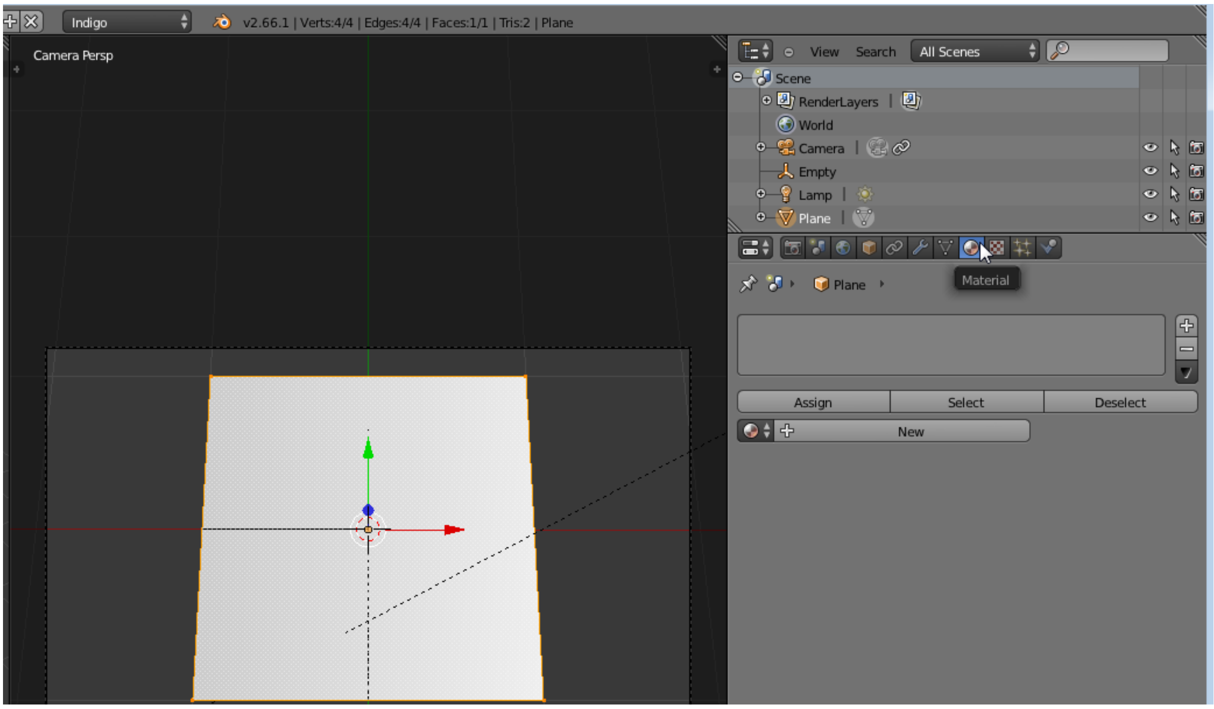
(A) Brightness: 0.00000

(B) Scale: 1.00000

(C) Offset: 0.00000

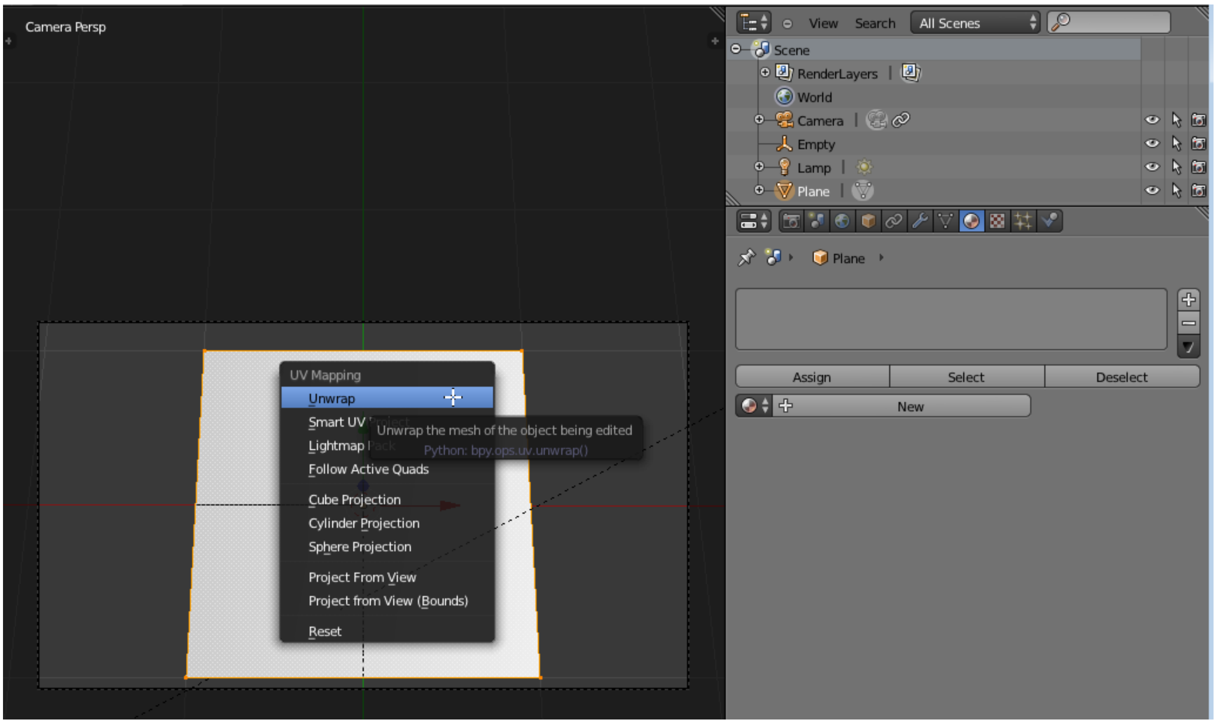
Picture 8 - The selected texture

Now lets move over to the Materials Panel.


v2.66.1 | Verts:4/4 | Edges:4/4 | Faces:1/1 | Tris:2 | Plane

Unwrapping

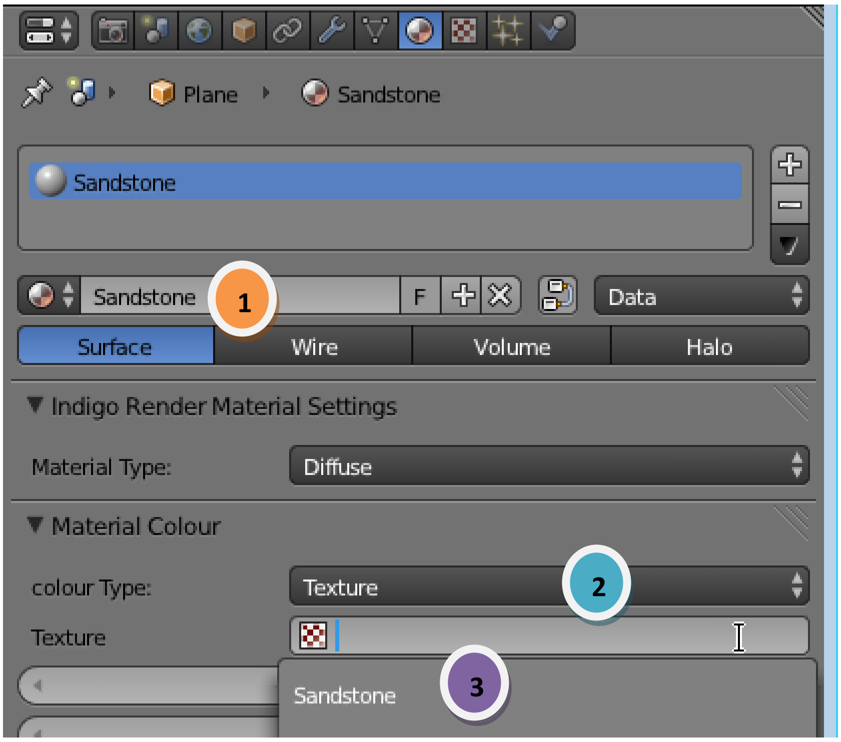
Now before we continue, we want to „UV Unwrap“ our plane.


Unwrap the mesh of the object being edited
Python: bpy.ops.uv.unwrap()

In Edit Mode (have the plane selected and press “TAB”), press “U” for “Unwrap”.

We want to keep things simple in this instance, so we just want to click “Unwrap” for now. After unwrapping, press TAB to exit Edit Mode.

Click “New”.



Plane › Sandstone

Sandstone

Material Type: Diffuse

colour Type: Texture

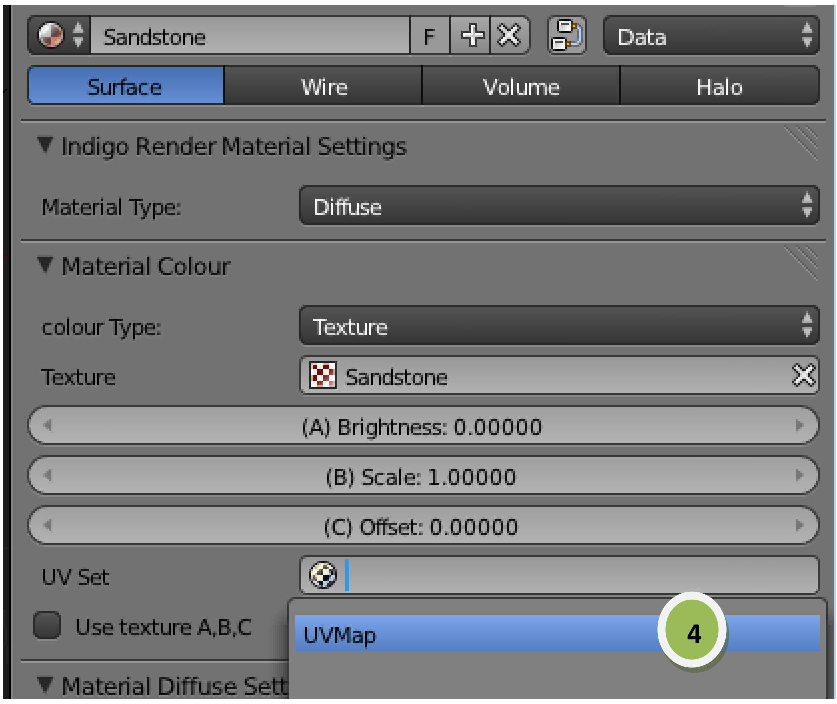
Texture

Sandstone

1. Give the material a name.

2. Leave the Material Type as it is for now.

3. Change the **colour Type** in the **Material Colour** section to “**Texture**” and choose the Texture You created in the dropdown menu.



Sandstone

Material Type: Diffuse

colour Type: Texture

Texture: Sandstone

(A) Brightness: 0.00000

(B) Scale: 1.00000

(C) Offset: 0.00000

UV Set

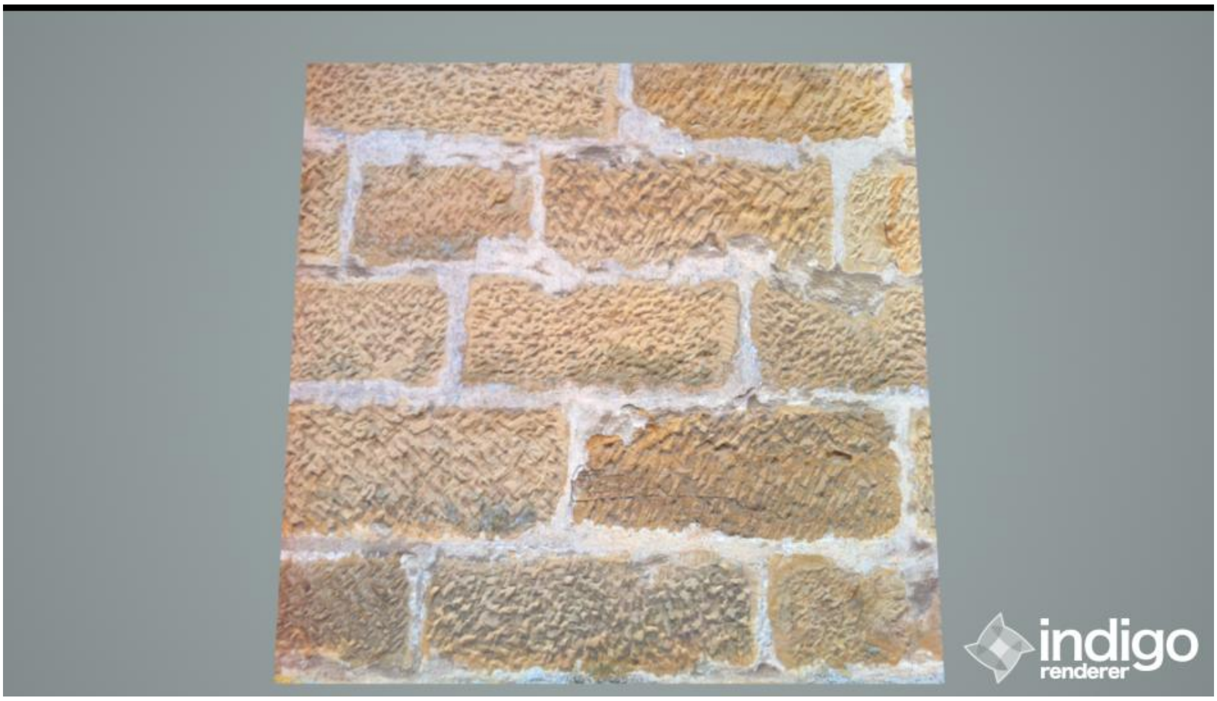
UVMap

□ Use texture A,B,C

4. Choose the UVMap we created when unwrapping our plane in the UV Set.

Rendering with plain texture.

Now we can hit Render again, and this is the result.



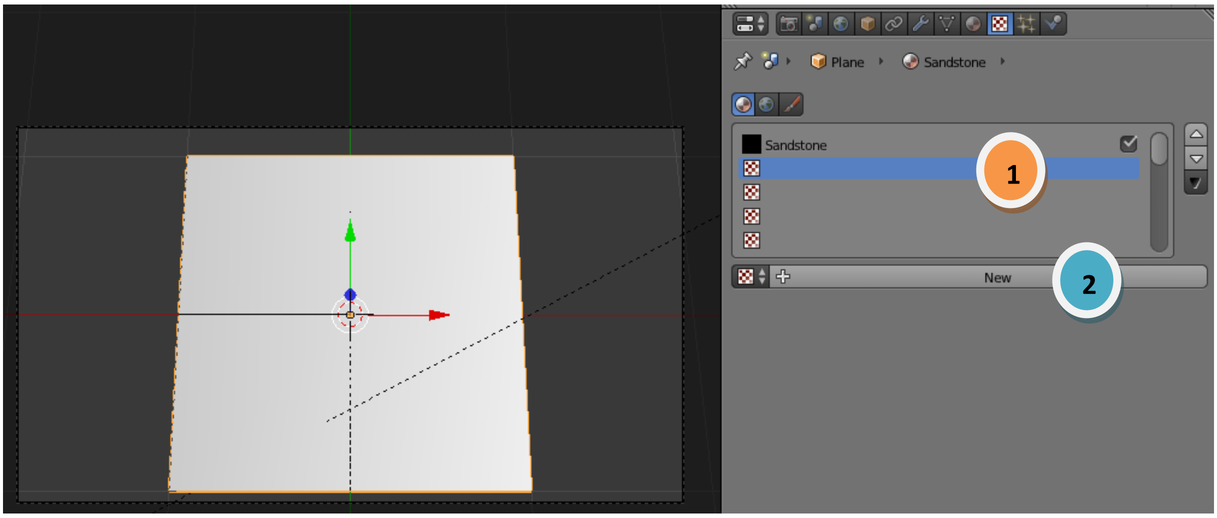
Are we happy? Yes and no. Yes, because we successfully gave this baby a texture. But this texture is plain old 2D on 3D. No ups, no downs, no shadows. That is NOT 3D! Our plane deserves better!

This is where we can use displacement mapping.

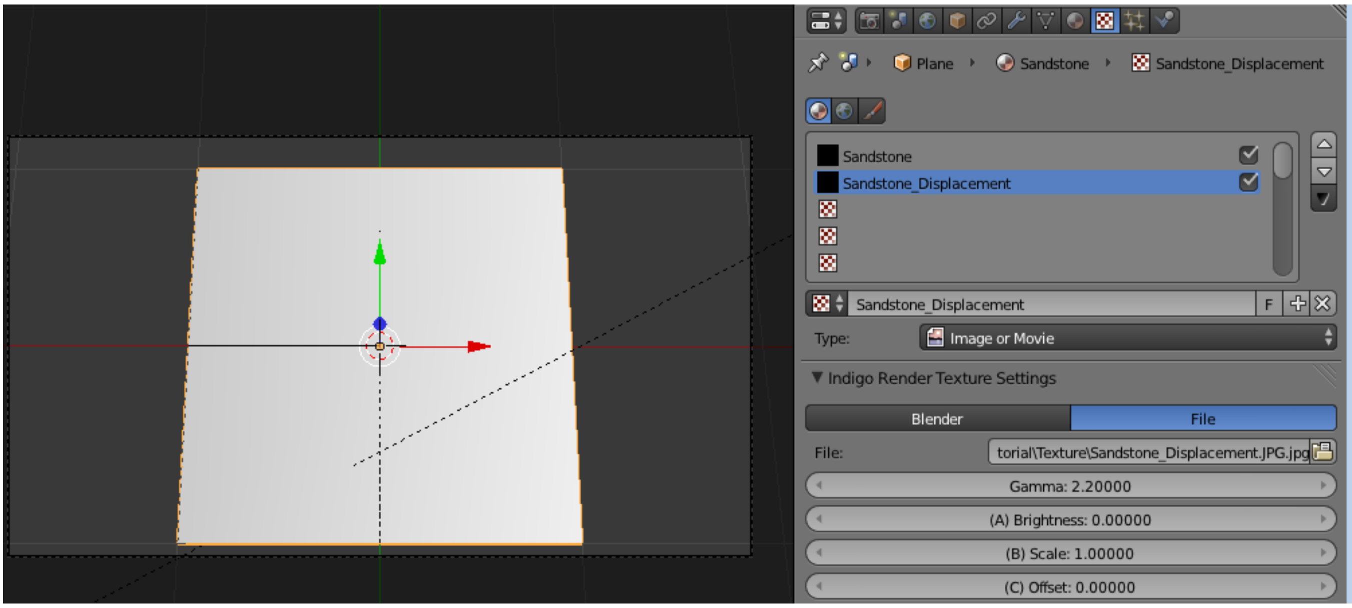
Displacement Mapping

Lets go back to the Texture Panel of our plane (make sure the plane is still selected).

1. Add a new texture (click on an empty slot first)



2. Repeat steps as when we added our first texture.



Plane ▸ Sandstone ▸ Sandstone_Displacement

☑ Sandstone

☑ Sandstone_Displacement

Sandstone_Displacement F

Type: Image or Movie

▼ Indigo Render Texture Settings

Blender | File

File: torial\Texture\Sandstone_Displacement.JPG.jpg

Gamma: 2.20000

(A) Brightness: 0.00000

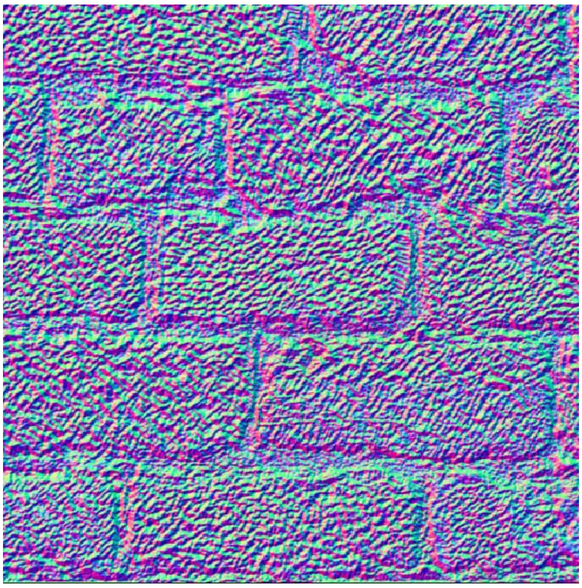
(B) Scale: 1.00000

(C) Offset: 0.00000

Check the “Material Bump Map” in the Materials section.

Choose the “Texture” as Bumpmap Type and the UVMap in the UV Set options.

This is a normal map I used for this example:



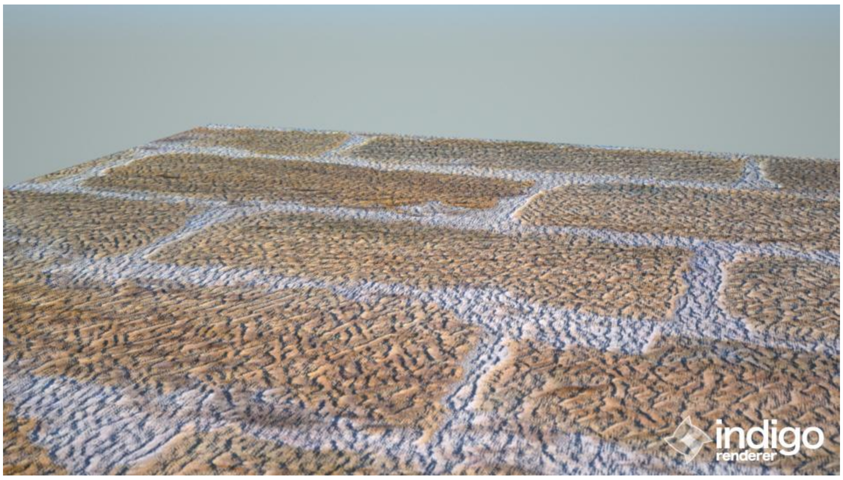
Definition of Normal and Bump Maps

The Blender Wiki defines Normal and Bump Maps this way: Normal Maps and Bump Maps both serve the same purpose: they simulate the impression of a detailed 3D surface, by modifying the shading as if the surface had lots of small angles, rather than being completely flat. Because it's just modifying the shading of each pixel, this will not cast any shadows and will not obstruct other objects. If the camera angle is too flat to the surface, you will notice that the surface is not really shaped.

Both *bump maps* and *normal maps* work by modifying the normal angle (the direction pointing perpendicular from a face), which influences how a pixel is shaded. Although the terms *normal map* and *bump map* are often used synonymously, there are certain differences.

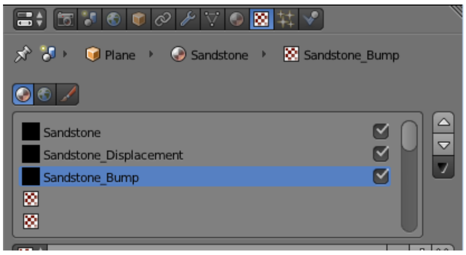
Render with Normal Map

Now, if we render again, we can see a difference: lots of details –without- adding any "real " geometry to our plane.



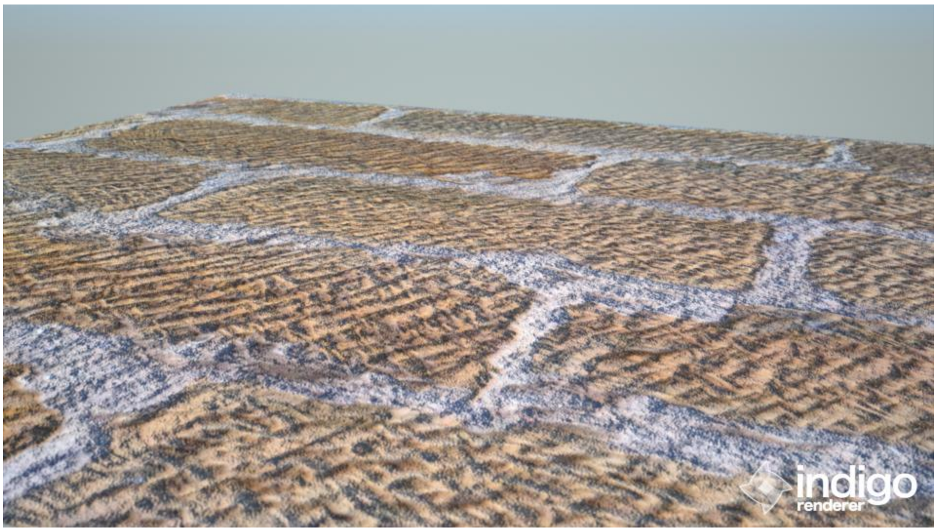
Render with Bump Map

Now, if we add a bump map (in the same manner as described before, just in the displacement map slot property of our object`s materials),



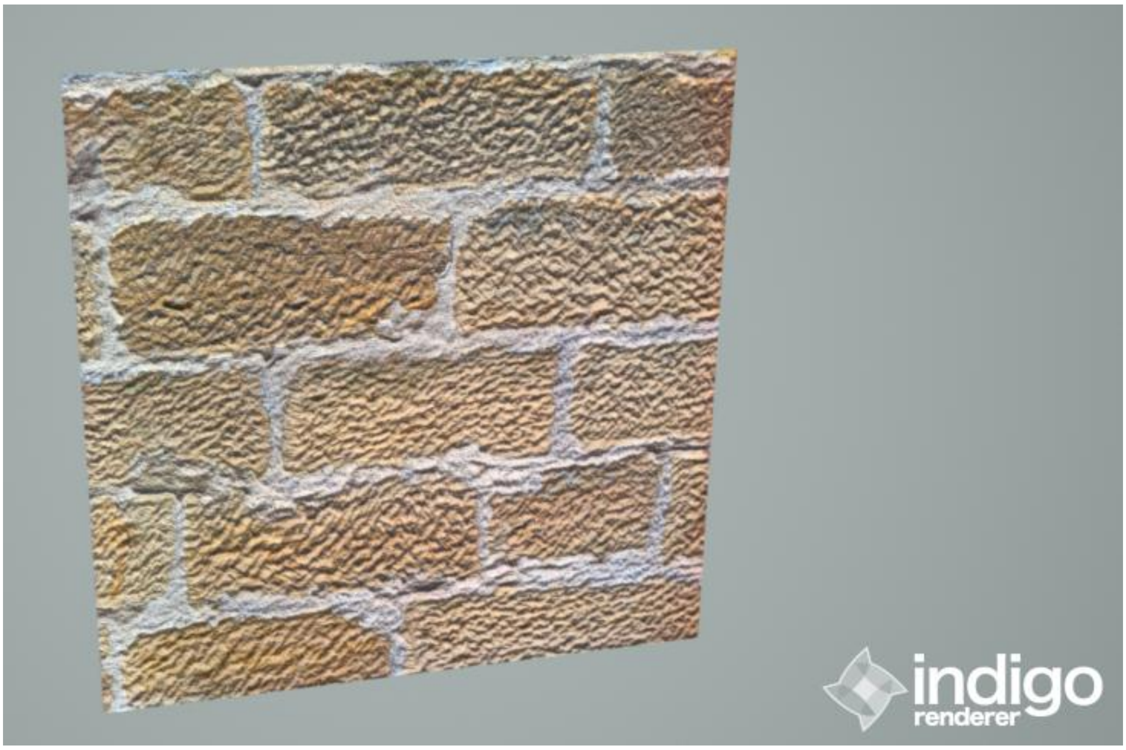
- Sandstone ☑
- Sandstone_Displacement ☑
- Sandstone_Bump ☑

we get this result:

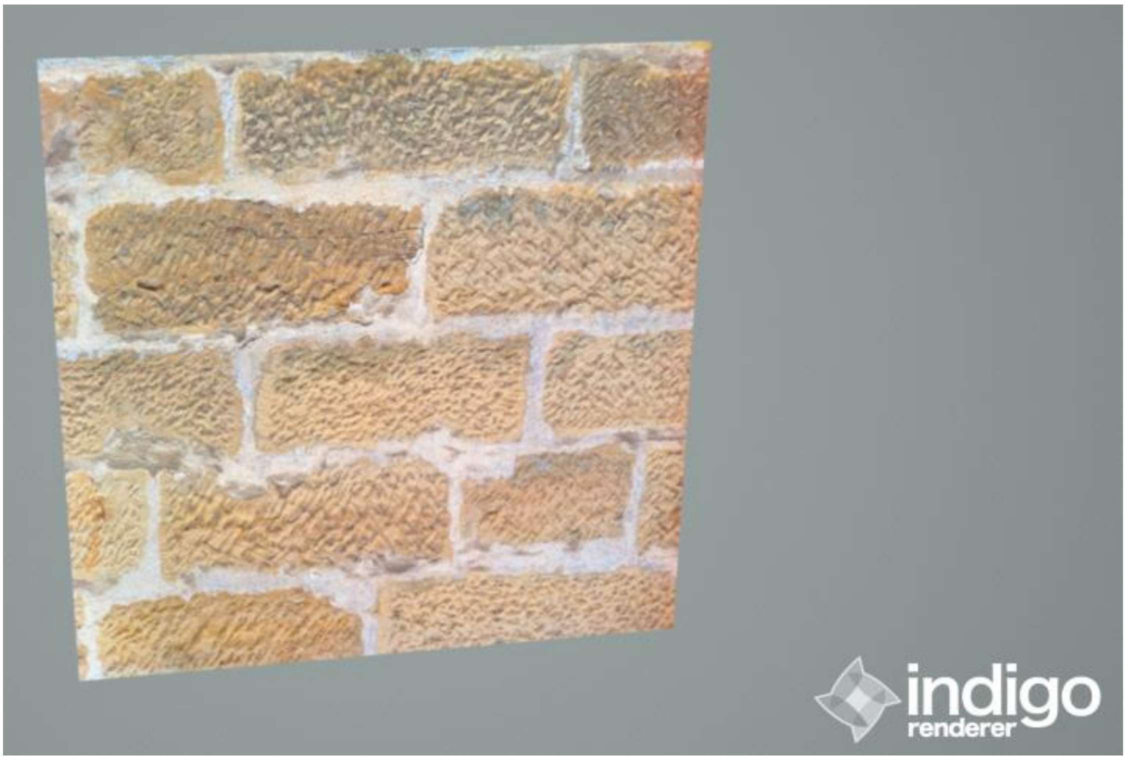


Comparison

Lets change our view to a front view: a comparison with and without our displacement and normal maps



Here we have a „natural“ look with just a plane and its texture maps applied.



Here we our plane with a flat texture and nothing else.