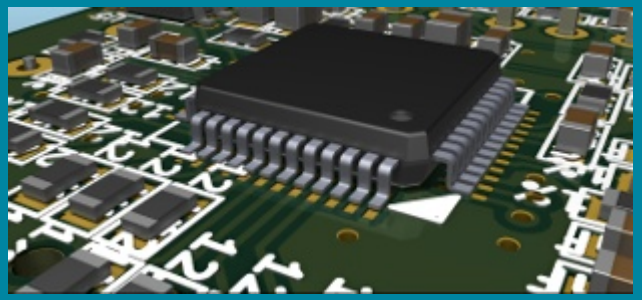


KiCad - 3D Viewer

Illumination model and materials (for 3D component model artists)

KiCad
http://kicad-pcb.org/

Mario Luzeiro
mrluzeiro@ua.pt

10 February 2016

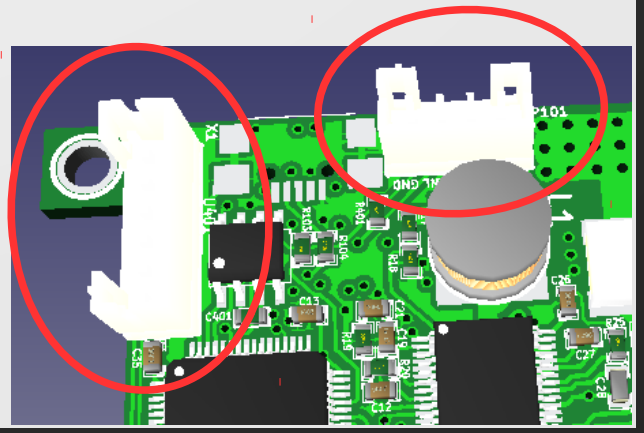
Propose and target audience

- Anyone that is interested in create 3D models for KiCad footprints.
- Learn how to use the most common material properties.
- Learn how to choose the best values for different material types.

- The propose of this guideline is to help artists to understand the materials properties hopping that we will get a consistency between different 3D model libraries for KiCad.

Background

- KiCad 3D-viewer started as a very simple 3D board and model visualization, supporting only a subset of the VRML 2 file format.
- On the first versions, almost only the diffuse property of the materials was used.
- Most of the artists that created models, some VRML exporters softwares, or online model databases, didn't add the other properties (the 'not so bad case'), others,
  add other properties with
  no good values (the 'worst case',
  eg: the emissive property).

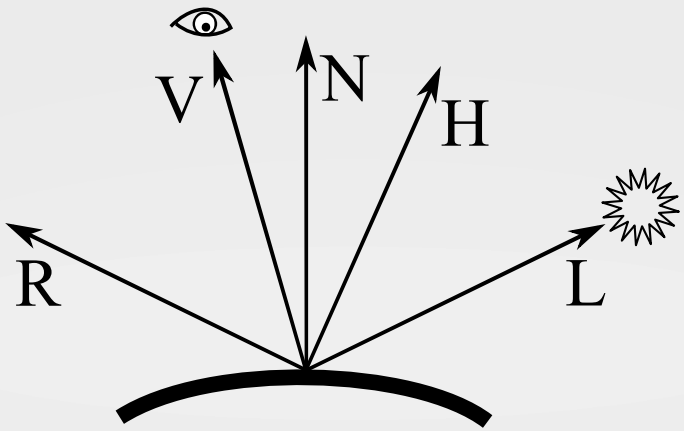


Background

- On the KiCad 4.0.1, it was added a feature in the 3D-Viewer to ignore the material properties. So it only select the diffuse color (this was a workarround to display bad material models).
- Since the 3D-Viewer is in continuous development, this materials will become more used. So in future, in order to get good results, we need to use models with good material properties assigned.

Scope

- This guidelines are related with illumination model

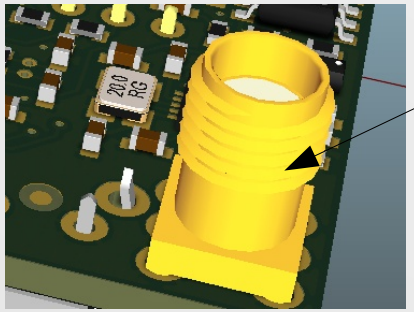


- It will not talk about textures or other image aspects.

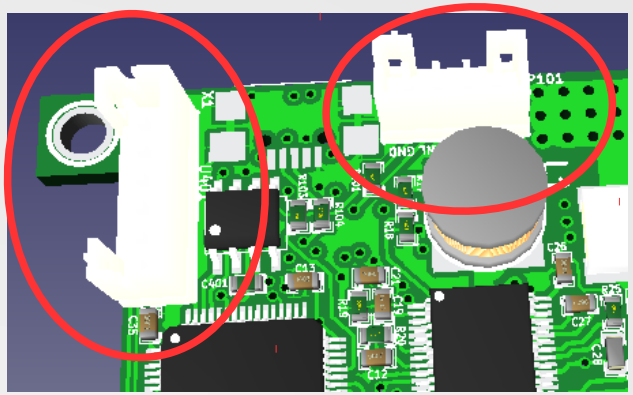
Common material issues

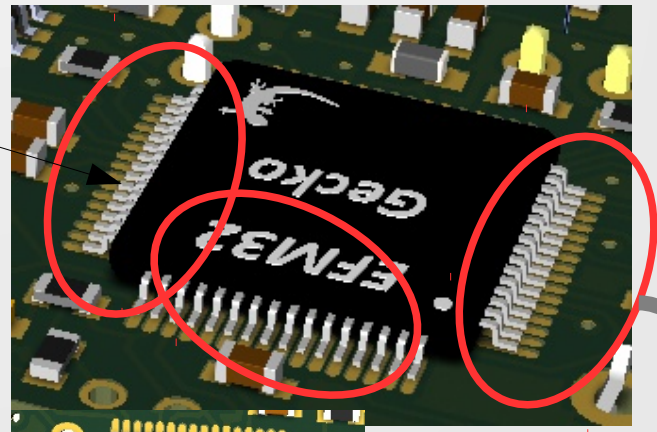
Common material issues found in the KiCad 3D model libraries:

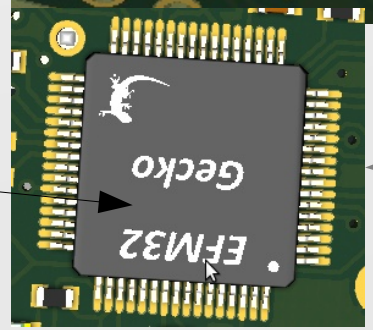
- Use the *emissive* property (shall not be used)
- *ambient* or *diffuse* too high (pins look always very bright)



- *shininess* too high (body will become too bright when reflect the light)

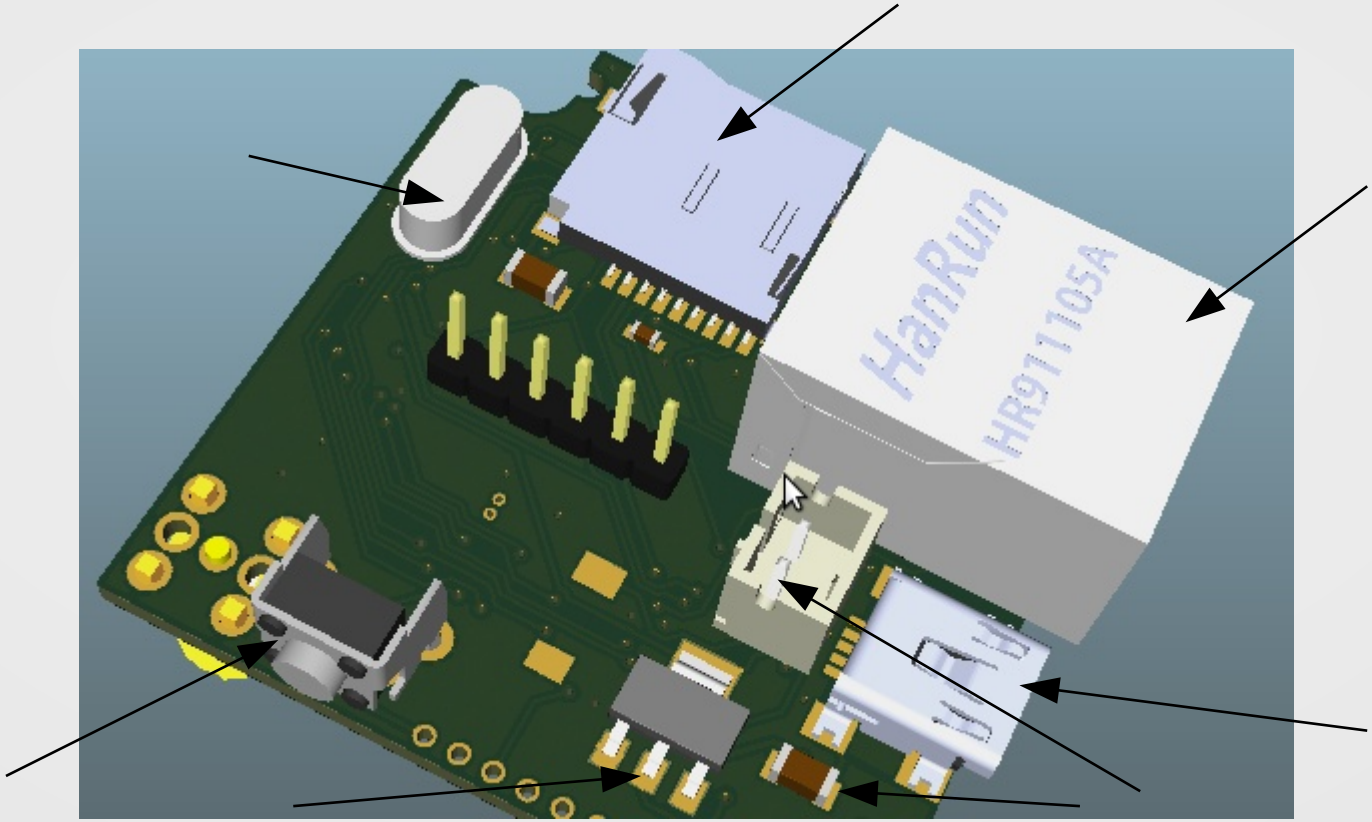






Common material issues

- Lack of consistency of adoption between different libraries causes weird differences:



All the materials indicated should be very similar but they look very different in color and shading.

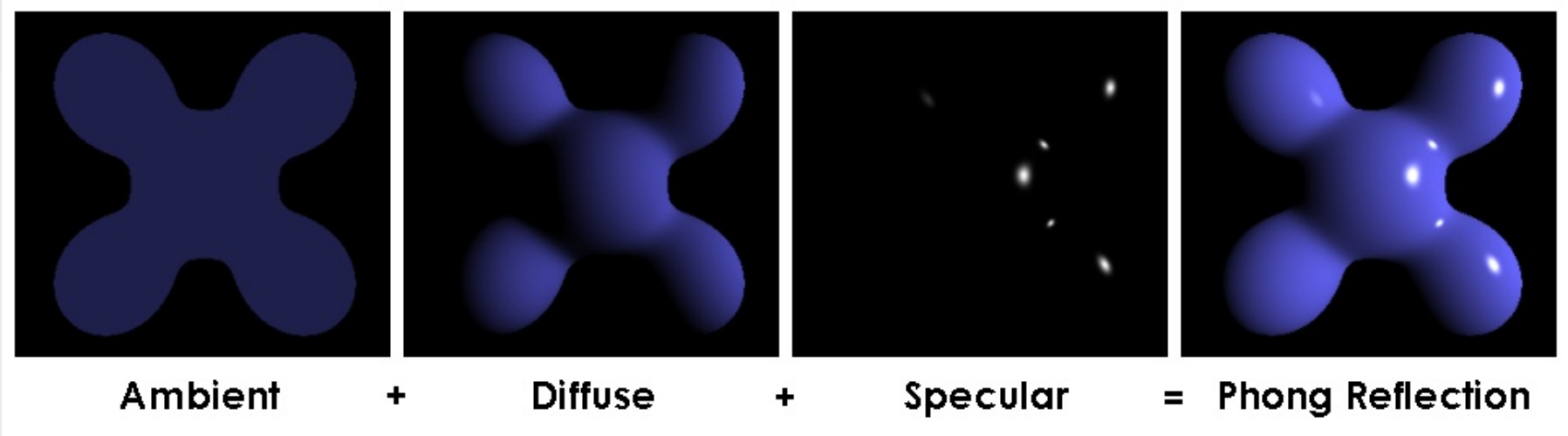
Blinn–Phong shading model

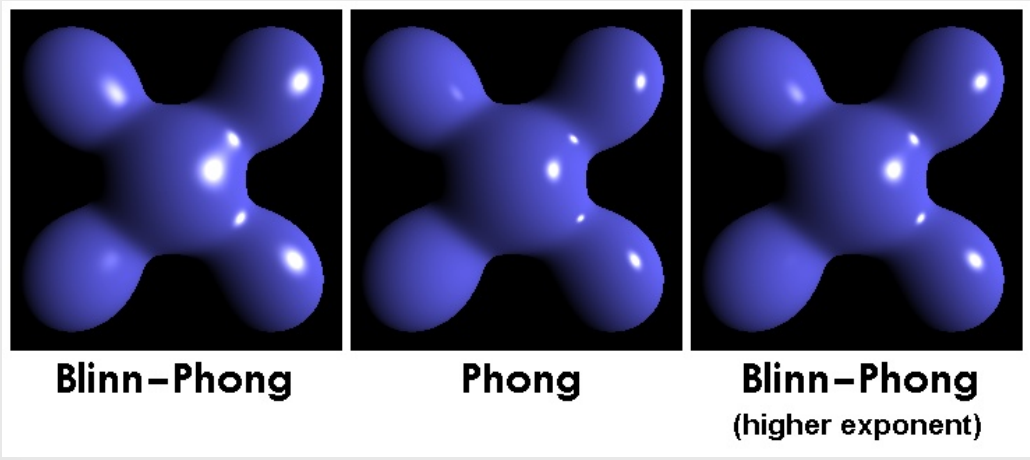
- KiCad implemented (initially) the 3D-viewer using the openGL; It also adopted (initially) VRML 2 file format for the 3D models.
- The VRML standard, proposes a shading model that is very similar to the one that openGL uses.
- They both use the most common material parameters:
  - Emissive, Ambient, Diffuse, Specular, Shininess and Transparency.

Blinn–Phong shading model at wikipedia

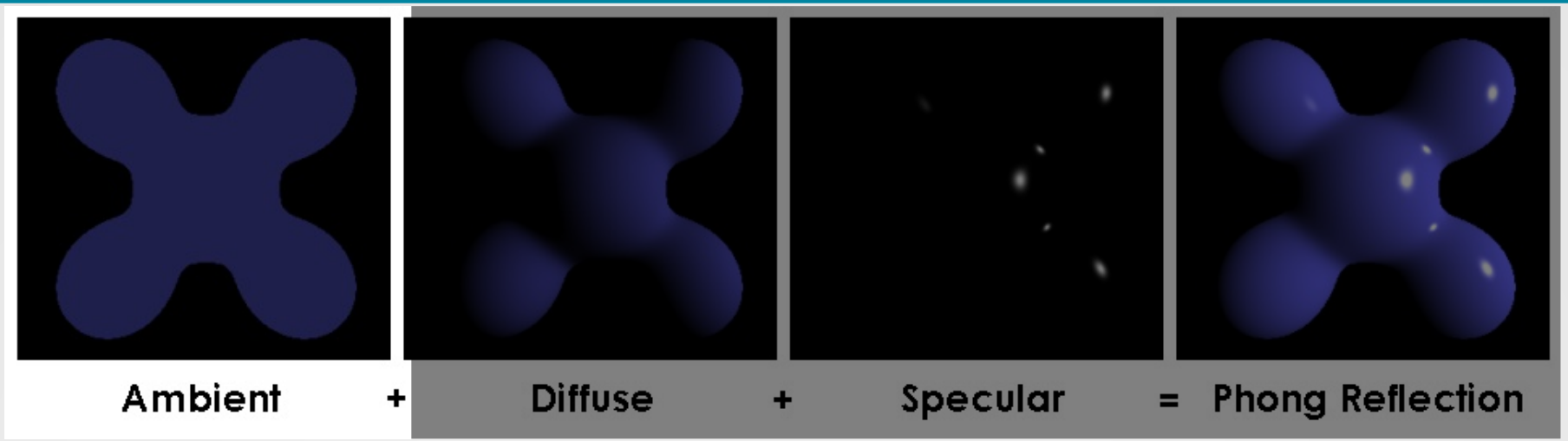
Blinn–Phong shading model

- The Blinn–Phong reflection model is a modification to the Phong reflection model



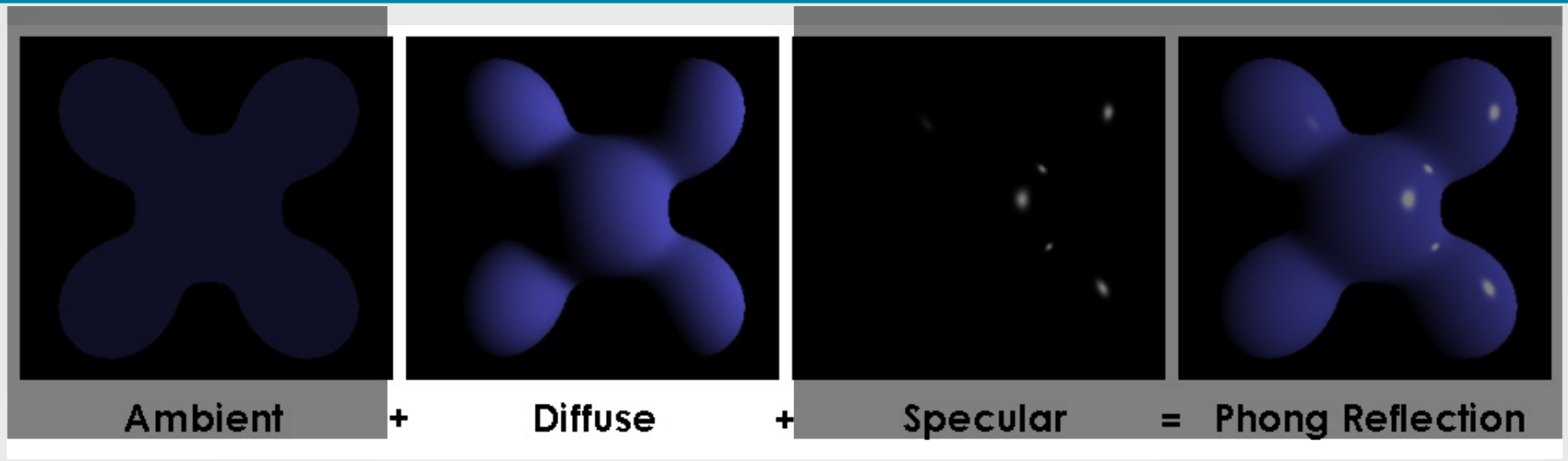


Blinn–Phong shading model - Ambient



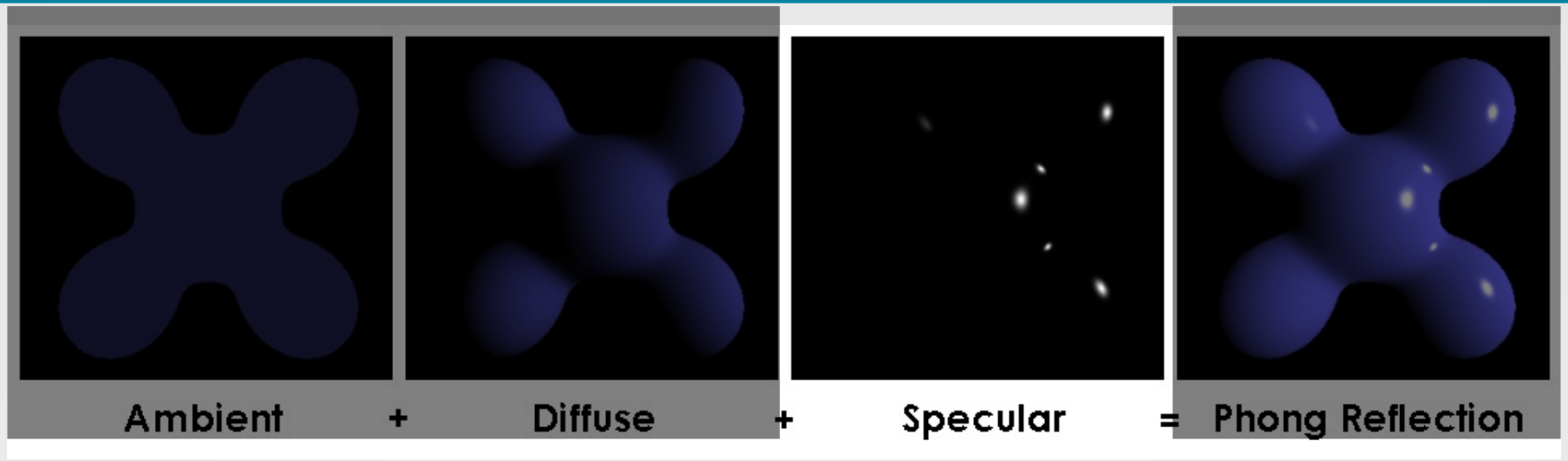
- **Ambient** – You can think that the ambient is the color (or factor) that will be added to the final render color when there is no light illuminating the model (I.e: the model is into shadow) but it is also added when there is light. So it work as a “bias factor”. You usually would like this value to be lower in intensity (but in the same hue color) as the diffuse.
  So you usually should expect values between 0.0 .. 0.25
- Note: Some formats (eg: VRML 1), define the ambient as an RGB color. Others (eg: VRML 2 and X3D) define the ambient as an intensity factor. (eg: 0.25) In this case it is a factor to be applied to the diffuse.

Blinn–Phong shading model - Diffuse



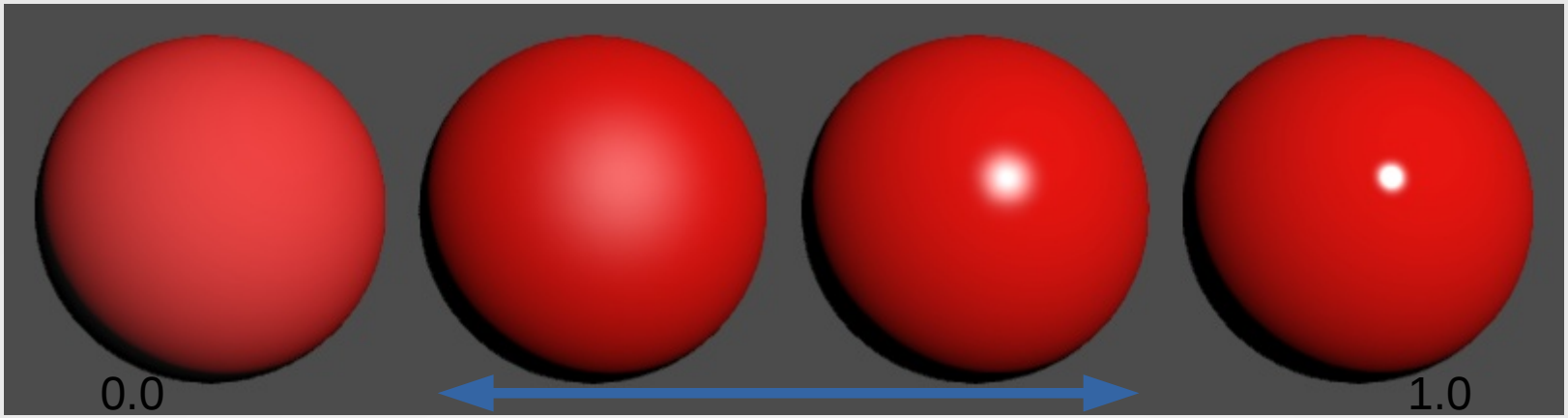
- **Diffuse** – It is the base color of the material. It is the most important one. You can think as it was in reality, the diffuse color would be the color of the material when you see it in a grey cloudy day, because it will light the object with the same light from the top hemisphere direction. You can have all ranges of RGB values (0.0, 0.0, 0.0 ... 1.0, 1.0, 1.0). But for this propose I suggest that you keep this at max around 0.8

Blinn–Phong shading model - Specular



- **Specular** – It is the color you see when the light is reflected into the object to you. This is a similar color as the diffuse. Some materials could reflect a color that is a bit different (in hue) than the diffuse color.

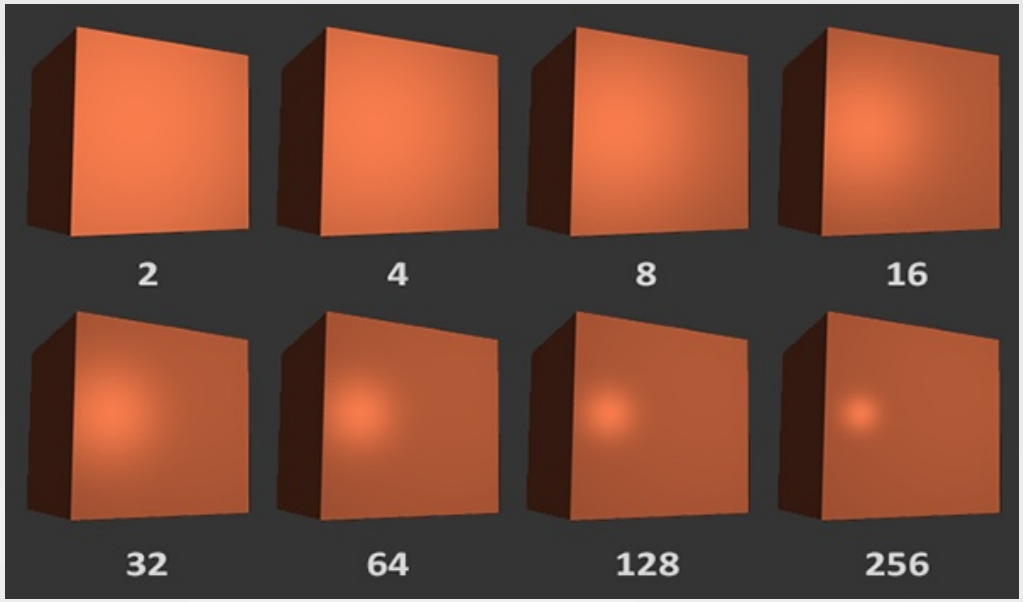
Blinn–Phong shading model - Shininess



(source)

- **Shininess** – The shininess factor is how much the light reflection is spread or concentrated in one spot. It is usually related to the reflection factor of the material. (some formats call it Hard factor)
- A value near zero will produce a smooth shine (left) and a high value (max 1.0) will produce very sharp spots hi-lights

Blinn–Phong shading model - Shininess



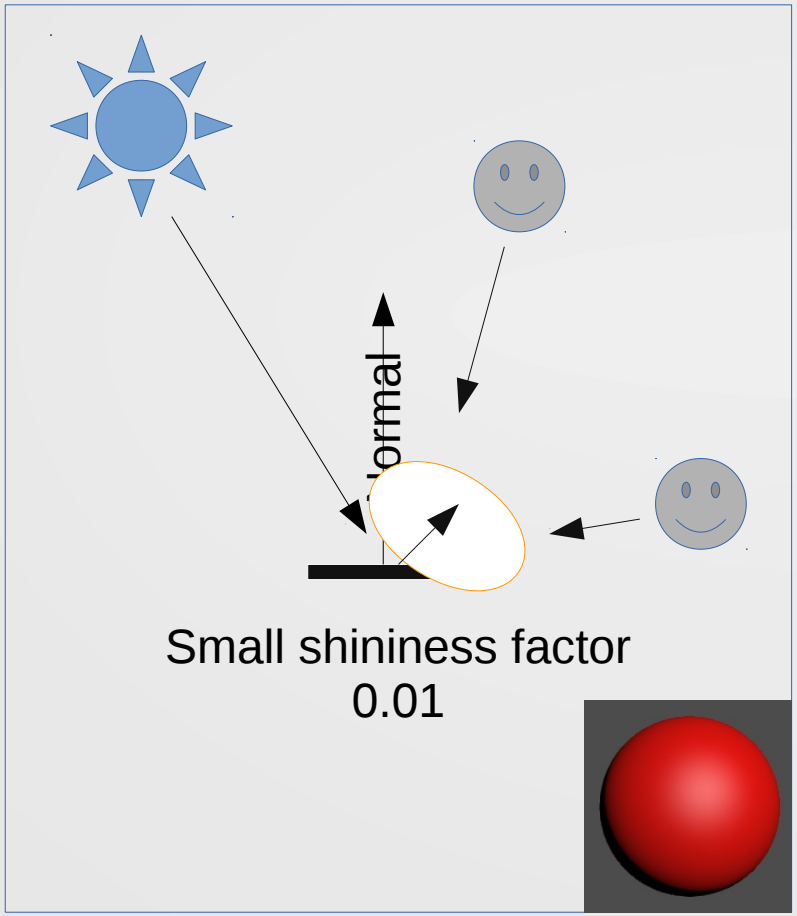
(source)

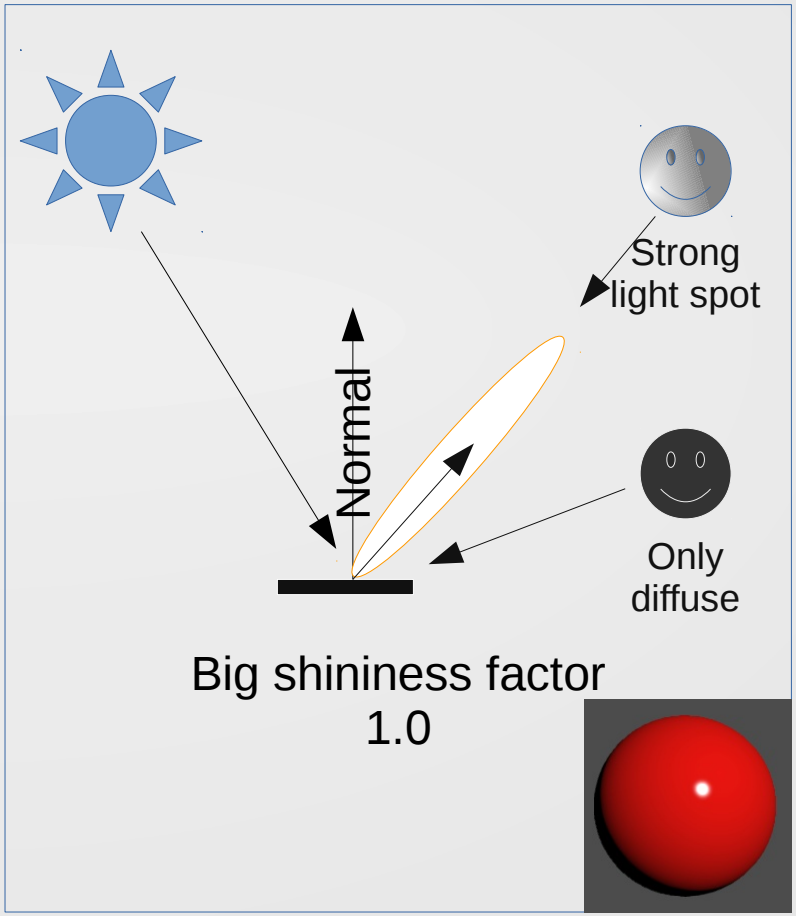
- Some formats consider other range of values for Shininess
  - OpenGL: 0 .. 128
  - VRML: 0.0 .. 1.0

  Please make sure you use the correct range for your file format.

Blinn–Phong shading model - Shininess

- Understanding the shininess





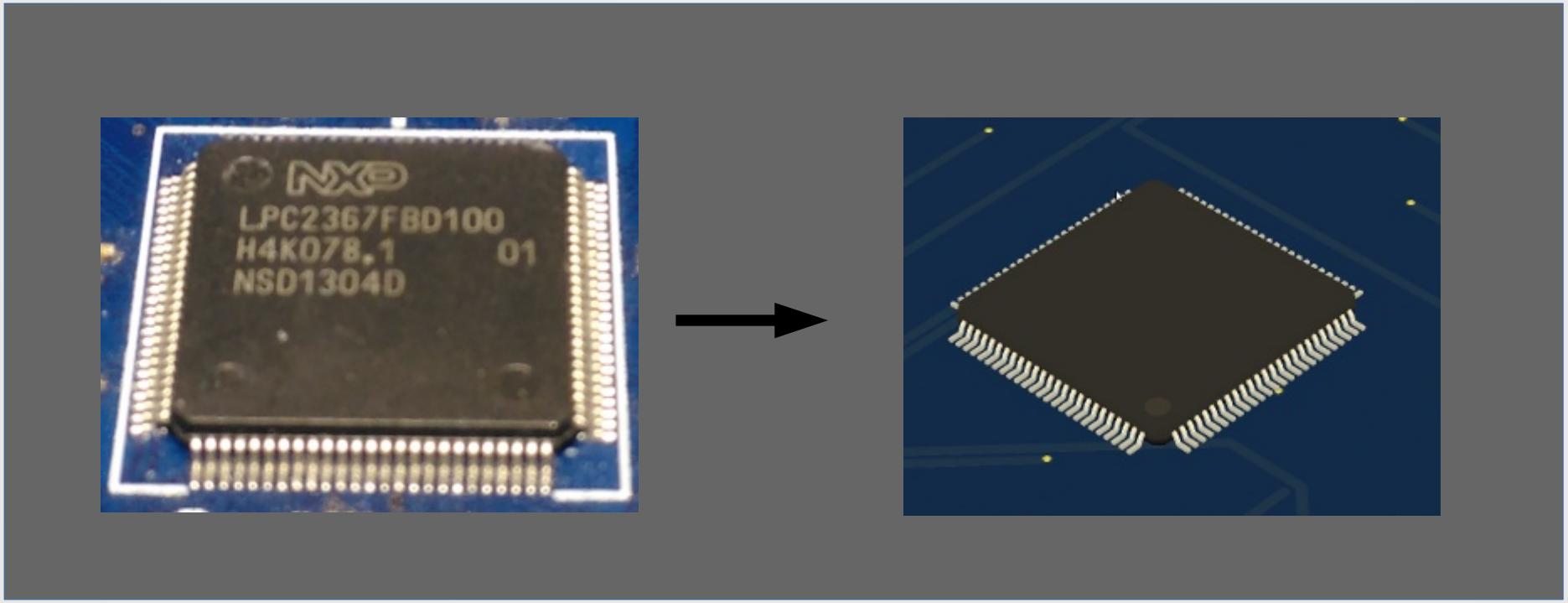
Emissive

- When to use Emissive?
- **Emissive** property, should only be used **if an element** of your model **produces light**.
- It shall only be assigned to the elements that are in fact producing light.
- E.g.: A bulb light filament (but not the glass), or to the LED element (but not the package or transparent plastic)

- I suggest (having in mind future use of Emissive) that you only assigned it to the real element that generate the light.
- Do not get tempted to add it to surrounding materials. Please use for that propose (with moderation) the Ambient property instead.

From photos to model materials

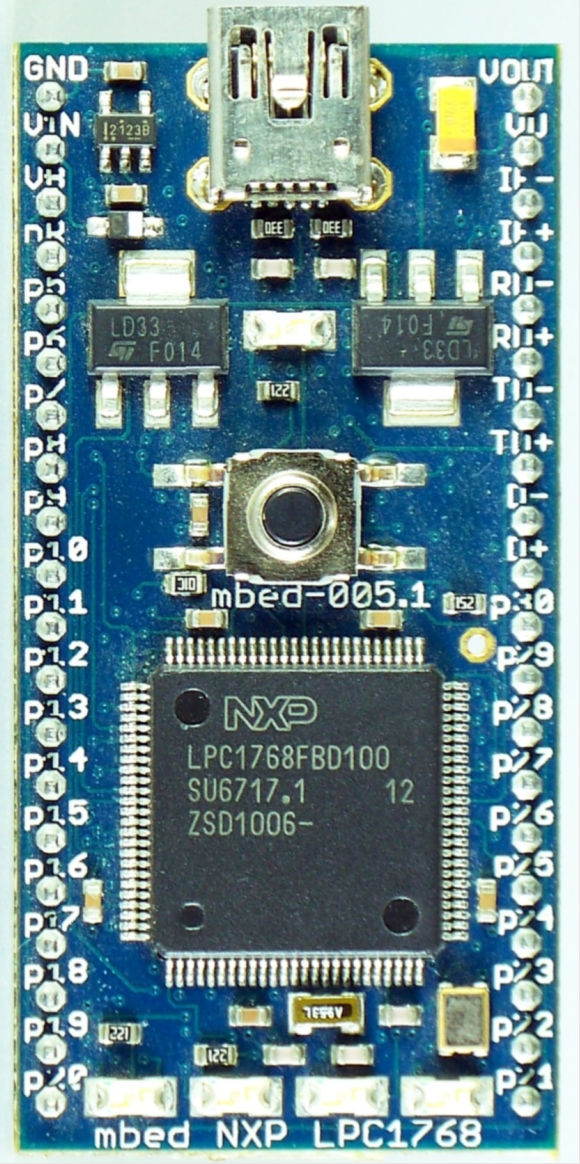
- We will learn now how can we take photos to components and get from it the material properties.



Bad photos

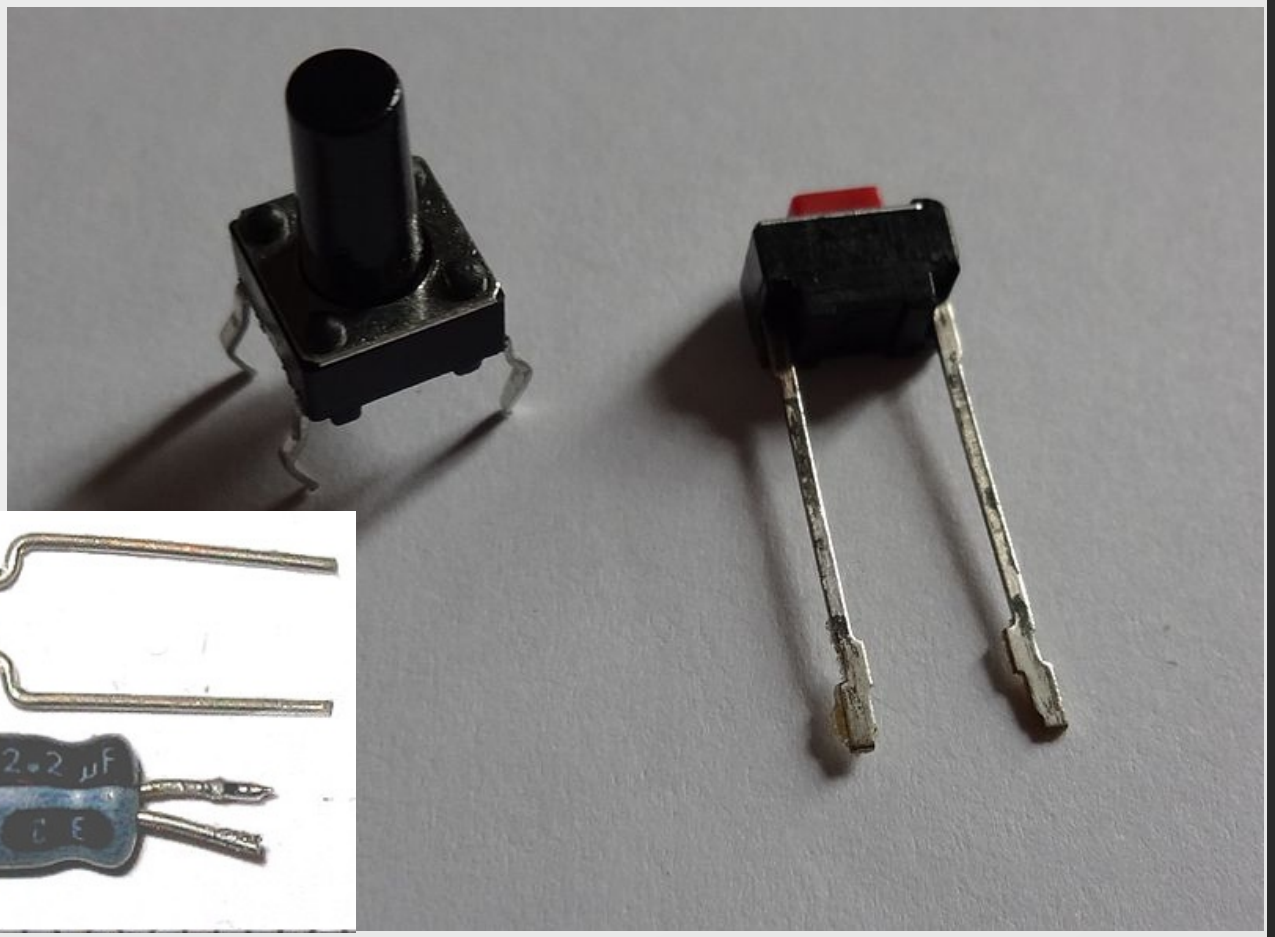
- DO NOT choose photos that:
  - Are very bright
  - Are very dark
  - Are artistic photos
  - Have low resolution details
  - The light is too diffuse and you don't understand the light direction.

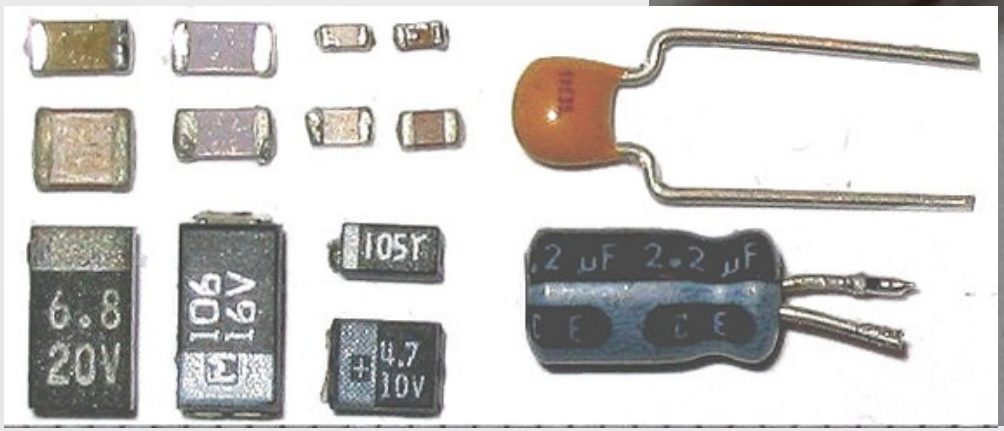
  - This example is a bit bright and there is a blue light. It is not good to pick materials properties.



Bad photos

- Washed (too light) or too dark (or too much contrast)



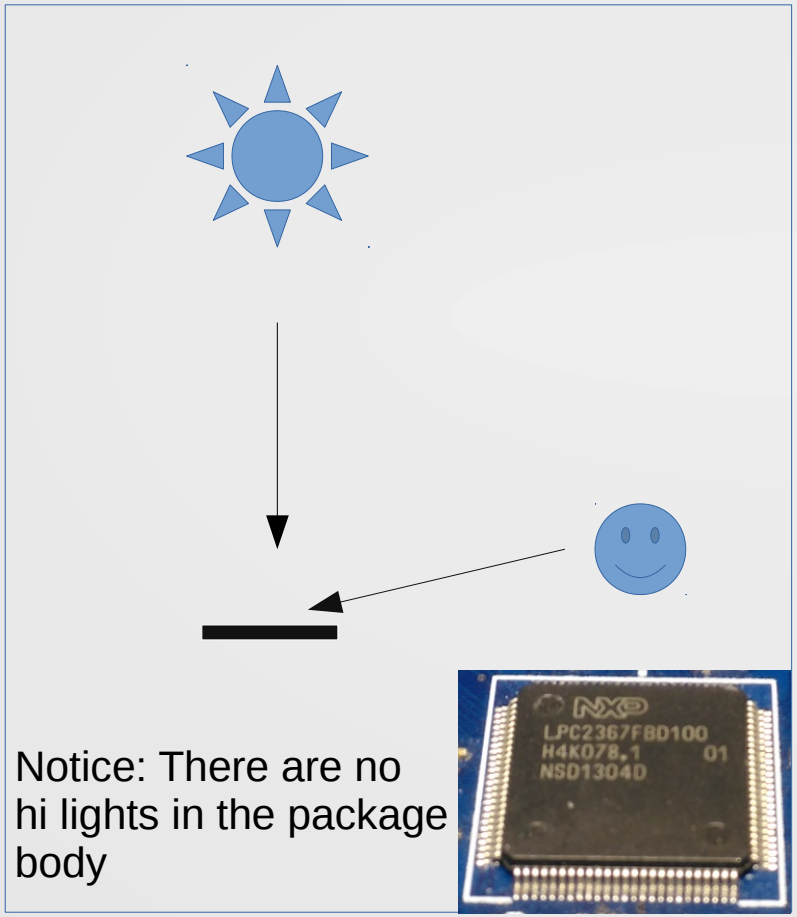


Create materials from photos

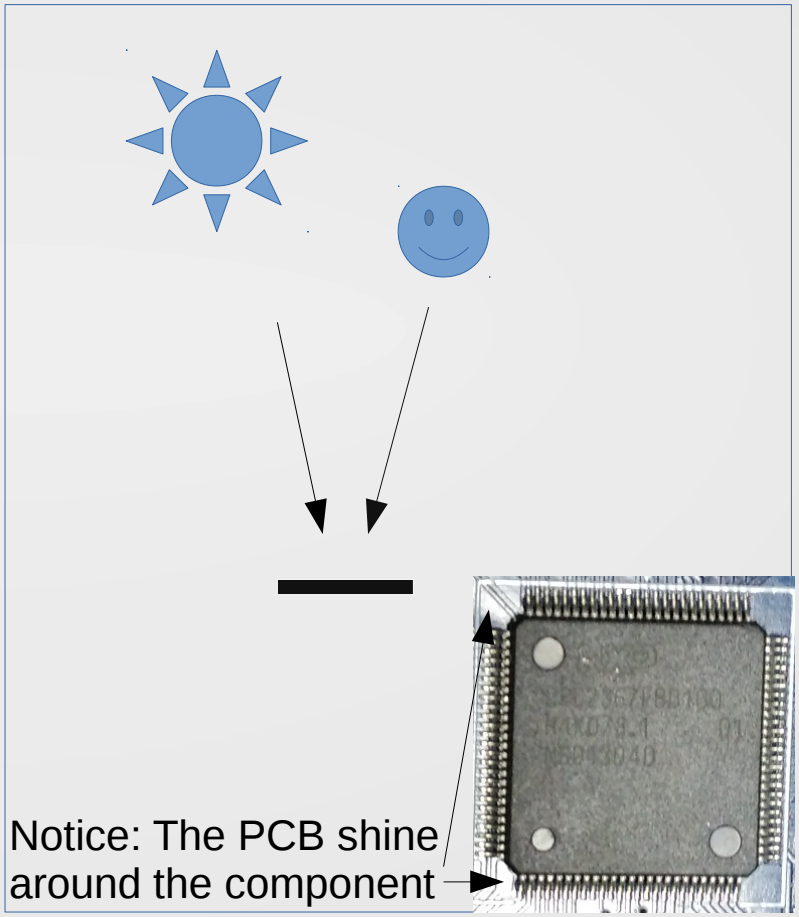
- Lets see how to get the material values for a TQFP100. You will need:
  - The component
  - A white fluorescent light (ideal small bulb 2 meters in the top in a ideal empty large room)
  - A camera (can be your phone camera if it is good). Ideally you use a camera that you can control the gains and exposition and try to keep it the same for different shots.

Create materials from photos

- Take two (or more) photos. In the flowing positions:



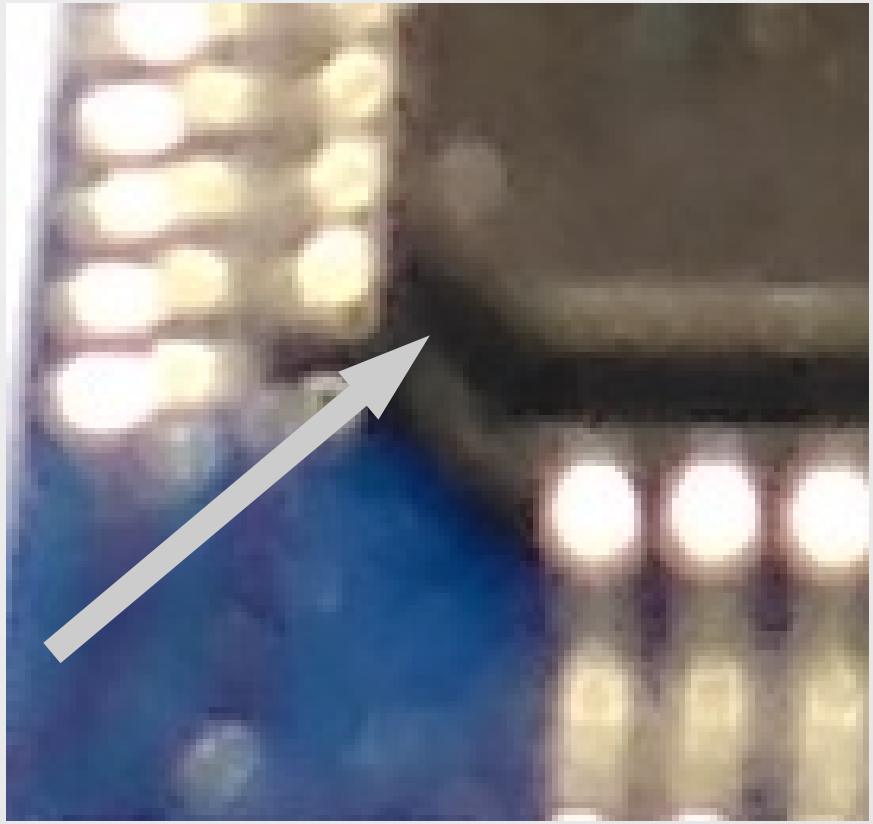
Notice: There are no hi lights in the package body



Notice: The PCB shine around the component

Create materials from photos - ambient

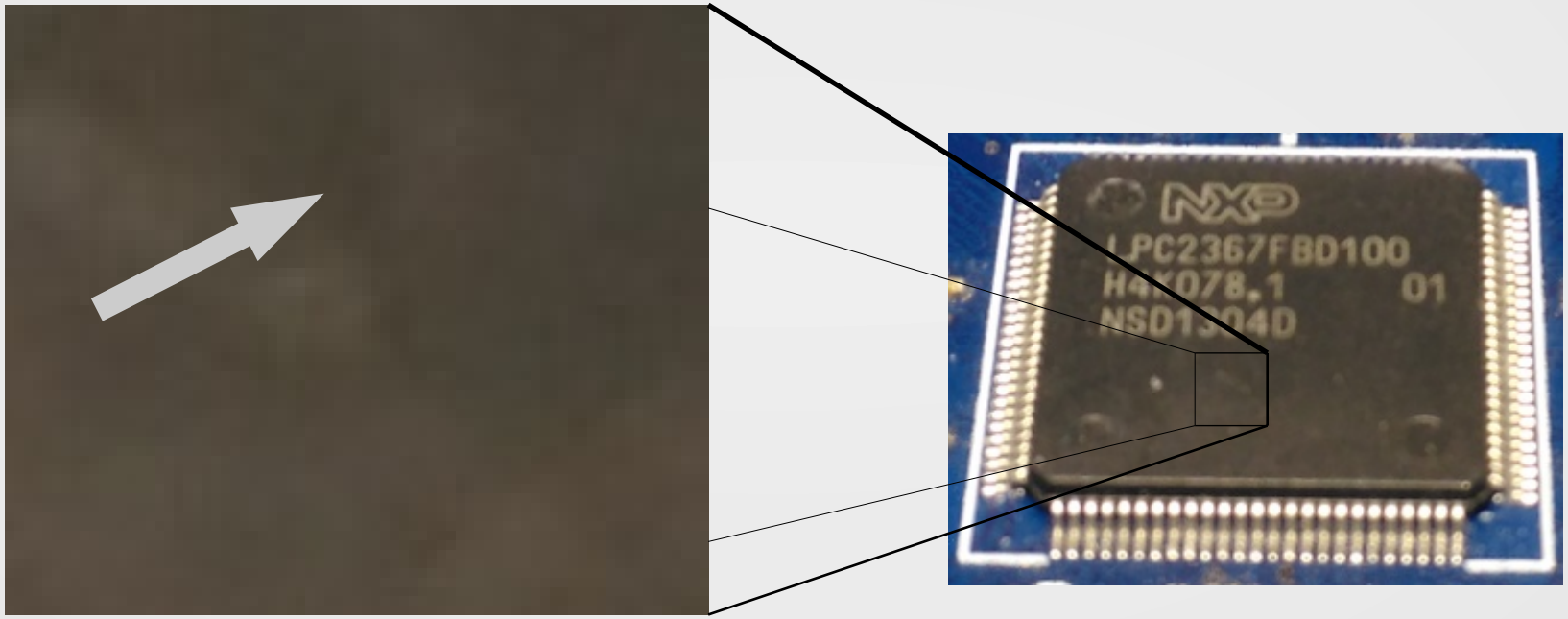
- Lets starting with the body package.
  Look for the darkest pixel point (use for example The Gimp to pick up the colors)
- You should select a pixel that is not completely dark but that still looks to be the color of the material.



- **Ambient**:
  RGB(27 27 29)
  RGB(0.105 0.105 0.113)

Create materials from photos - diffuse

- Choose a (not bright) color that is the color of the object material.



- **Diffuse**:
  RGB(63 59 50)
  RGB(0.246 0.230 0.195)

Create materials from photos - ambient

- If you are calculating the ambientIntensity property in VRML 2, you need to perform the following conversion:

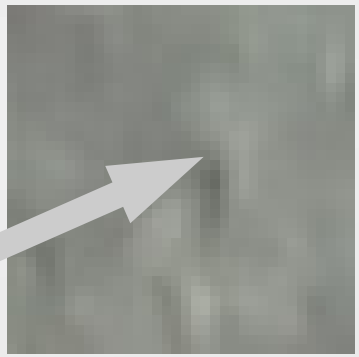
- **Diffuse**:
  RGB(63 59 50)
  RGB(0.105 0.105 0.113)

- **Ambient**:
  RGB(27 27 29)
  RGB(0.105 0.105 0.113)

- `ambientIntensity =`
  `(0.212671 * amb.r + 0.71516 * amb.g + 0.072169 * amb.b) /`
  `(0.212671 * dif.r + 0.71516 * dif.g + 0.072169 * dif.b)`

Create materials from photos - specular

- Taking now the other photo with the light reflection, choose a medium bright color.
- The specular final color will be:
  - Specular = (Picked Color – Diffuse)



- **Specular**:

RGB(113 112 107) - RGB(63 59 50)==**RGB(50 53 57)** == **RGB(0.195 0.207 0.226)**

Back to the theory...

- Why do we subtract Specular with Diffuse?

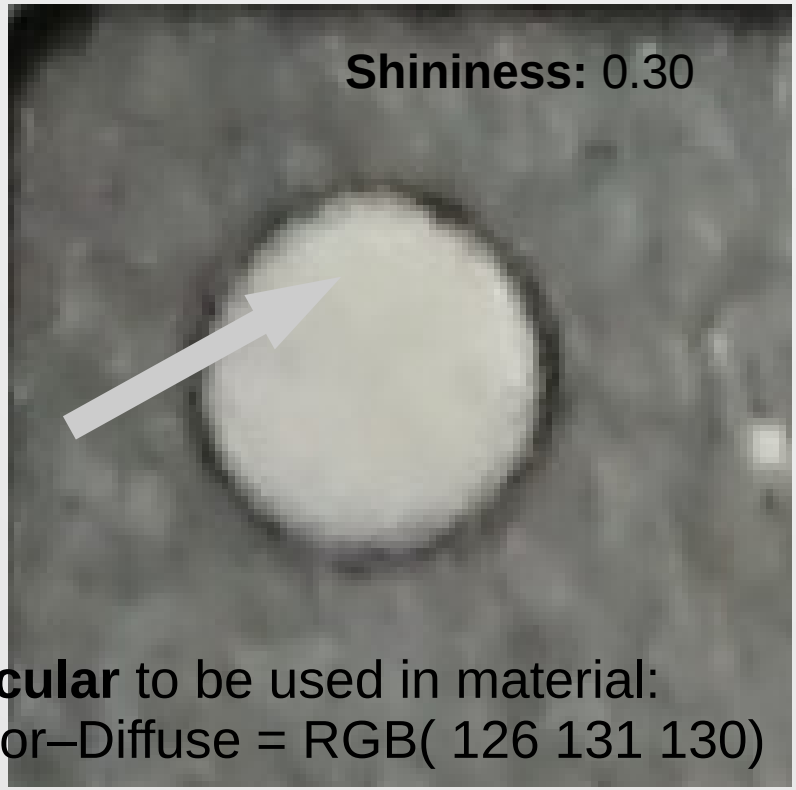
$$I_\mathrm{p} = k_\mathrm{a} i_\mathrm{a} + \sum_{m \in \text{ lights}} (k_\mathrm{d}(\hat{L}_m \cdot \hat{N}) i_{m,\mathrm{d}} + k_\mathrm{s}(\hat{R}_m \cdot \hat{V})^\alpha i_{m,\mathrm{s}}).$$

  (simplifing) Final color = Ambient + Diffuse + Specular
  So if we pick our final color, the A+D must be subtracted to get the Specular component.

- Will be that way always?
  - Due some "mathematical issues" with the Blinn-Phong model, when you use high values for shininess, it may be possible that you want to increase a bit more the specular value, so you will get bright saturated spots (if that is what you desire).
    http://www.rorydriscoll.com/2009/01/25/energy-conservation-in-games/

Create materials from photos

- If your model (as an example) have a different material for the mark, you can do the same process for all other materials.
- **Specular** (190 190 180)$_{(picked)}$ **Diffuse**: 80 72 58

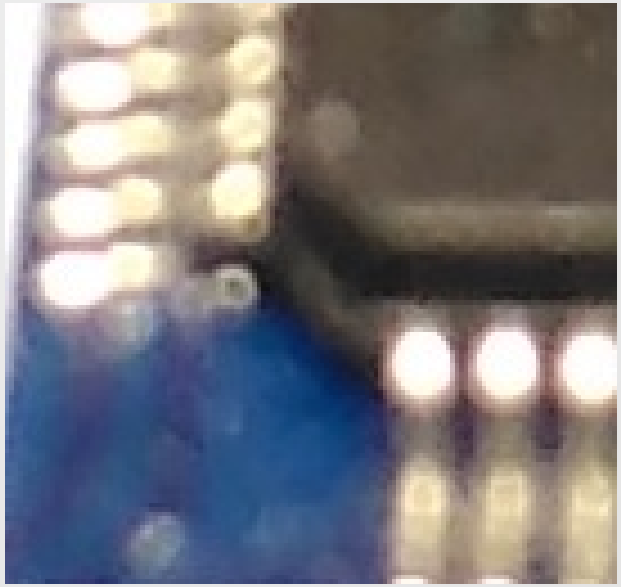


Final Specular to be used in material:
Picked color–Diffuse = RGB(126 131 130)

Create materials from photos

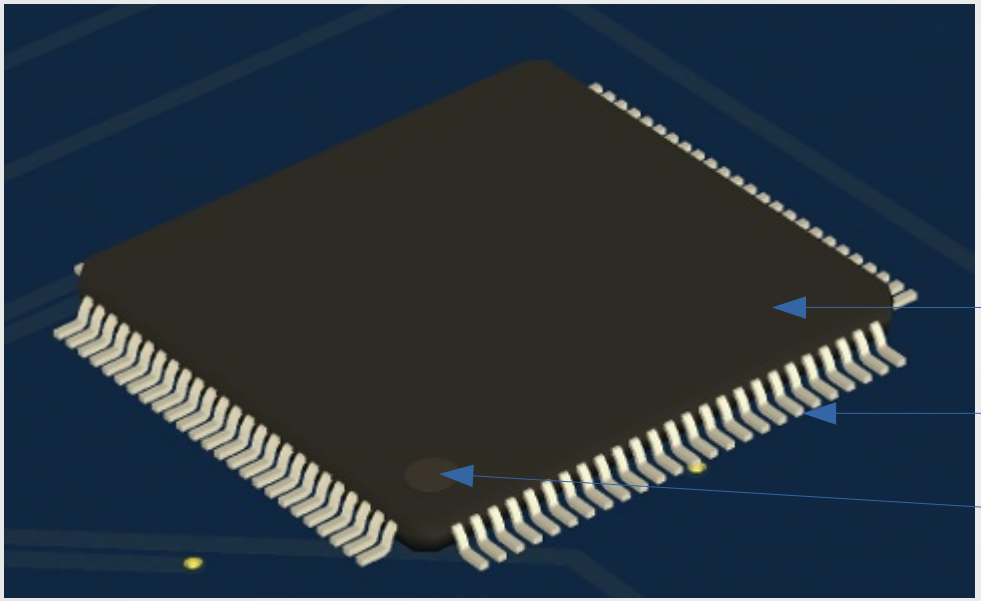
- Do the same for the pins
  Ambient: RGB(97 82 63)
  Diffuse: RGB(251 241 206)
  Specular: RGB(255 253 248) → RGB(4 12 42)
  Shininess: 0.6 ← high value because they shine a lot!

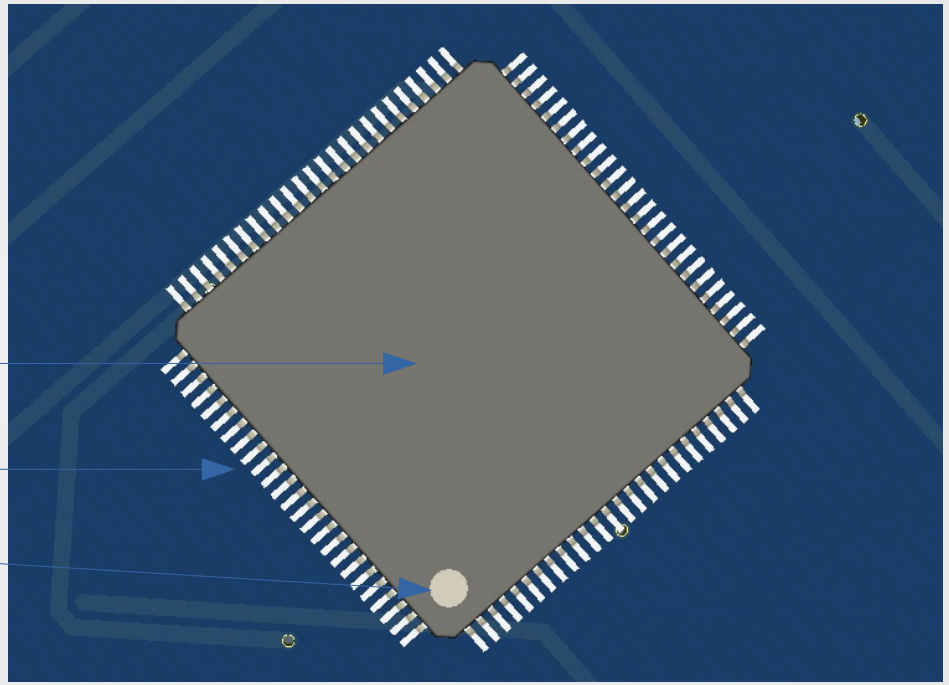




Create materials from photos

- Final result render in KiCad 3D-viewer





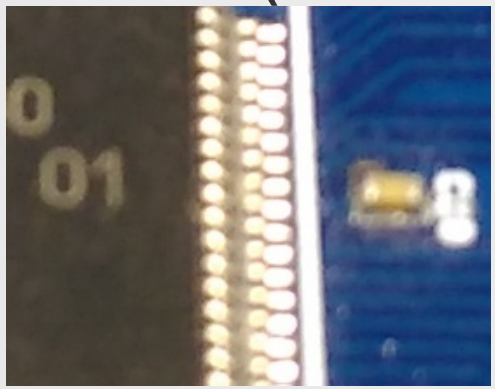
Note: Consider that the KiCad 3D-viewer place a directional light on the top and other in the bottom (0.7 intensity) Also, there is an head light in front of the camera(0.3 intensity)

Criticize the result

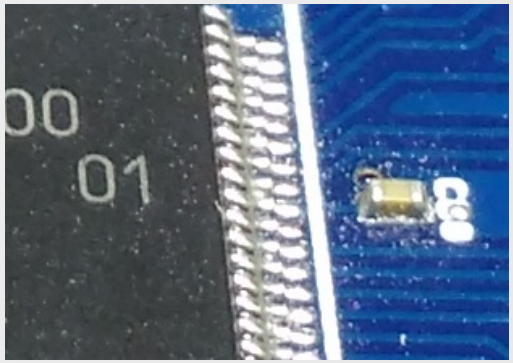
- The results evaluation are both human and technical relative.
- I personal like the result, it have a "natural looking", but I admit that it may look a bit brown, but trust me, it was much how it look in the reality. In fact there was other more darker / black ICs in the board than this one.
- There maybe are some reasons for that:
  - The light I was using was not totally white (it was a more 'warm light', so the pin reflection was a bit more yellow.)
  - The phone camera is doing (or was not doing correctly) some internal auto-white-balance (since I took the picture in to a blue PCB)
  - The IC pins are soldered. The solder resin could make the pins more 'yellowish'.

Criticize the result

- Considering other camera settings (right) it will look more bluish. However it may be too much. So maybe something in the between will be a better choise.
- RGB(63 59 50)

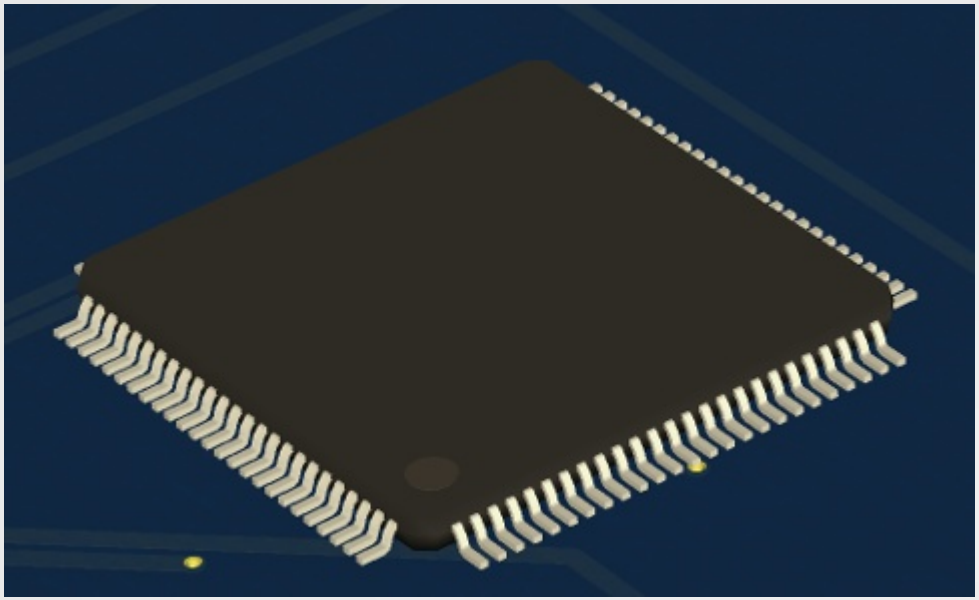


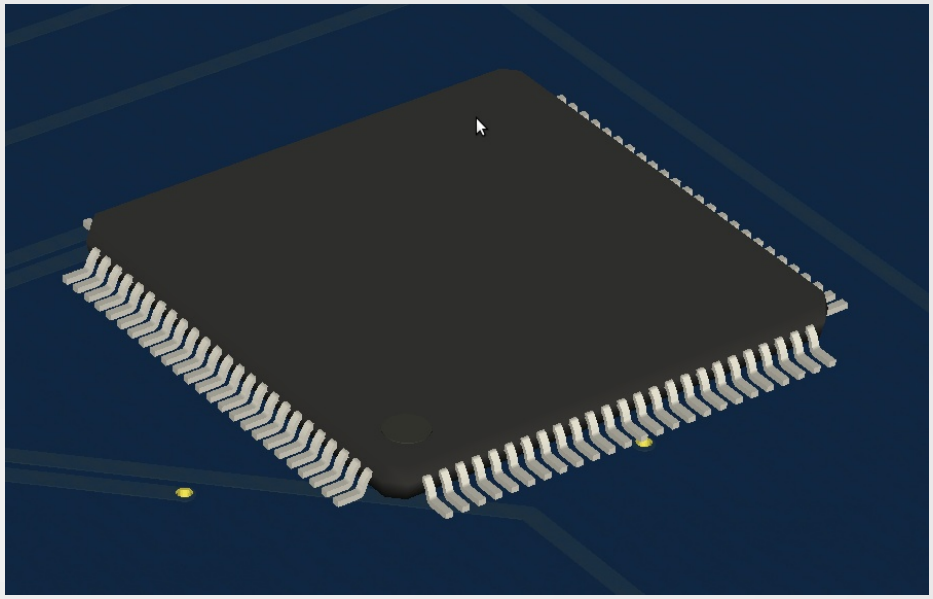
RGB(66 70 71)



RGB(65 64 61)

Criticize the result





RGB(65 64 61)

This is the type of judgement exercise that you must do when choose the right values for the materials.

Tips

- You can try to get the values from a single good photo

- You may would like to blur a bit some parts of the image when you are not sure which pixel to pick

Tips

- If you are developing VRML (VRML 1, 2 or X3D) files for KiCad, view3dscene is a good software for testing.
  http://castle-engine.sourceforge.net/view3dscene.php
- You can add to your test models the following lines:

```
DirectionalLight {
  on TRUE
  intensity 0.7
  ambientIntensity 0.08
  color 1.0 1.0 1.0
  direction 0.0 0.0 -1.0
}

DirectionalLight {
  on TRUE
  intensity 0.7
  ambientIntensity 0.08
  color 1.0 1.0 1.0
  direction 0.0 0.0 1.0
}
```

- Enable also the head light and configure it to 0.3 intensity.
- Mind that if your model have bad normals then it will not look right. (in the view3dscene choose to remove the model normals)
- It is better that you let the software to calculate the model normals (KiCad 3d-Viewer can calculate good normals)

Tips

- While checking your models on KiCad, make sure the model is correct reloaded: It is better to close and open again all the software.
- Make sure you have the option selected to use model materials.