The new material system in MBS V6 based on Blender (V2.83 ff.) Part 1

In version V6 of the MBS, a new material system for model building is established and is referred to as PBR (physically based rendering). This is more realistic offers display options.

The effects in use are:

Color	(basic color, nothing changes)	
Metallic	(metallic luster, variably adjustable)	replaces the _ENV_XXX
Roughness	(roughness, variably adjustable)	replaces the _ENV_XXX
emissives	(emitted light, variably adjustable)	replaces the _LS

_LC is retained for the time being.

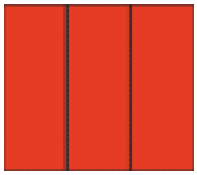
A multi-texture should continue to be used for the colouring in Blender, and textures should also be created for the "metallic, roughness and emissive" parameters.

The power of two as the resolution in pixels is retained as before: eg 64 x 64, 256 x 128,... as *.png, or *.dds.

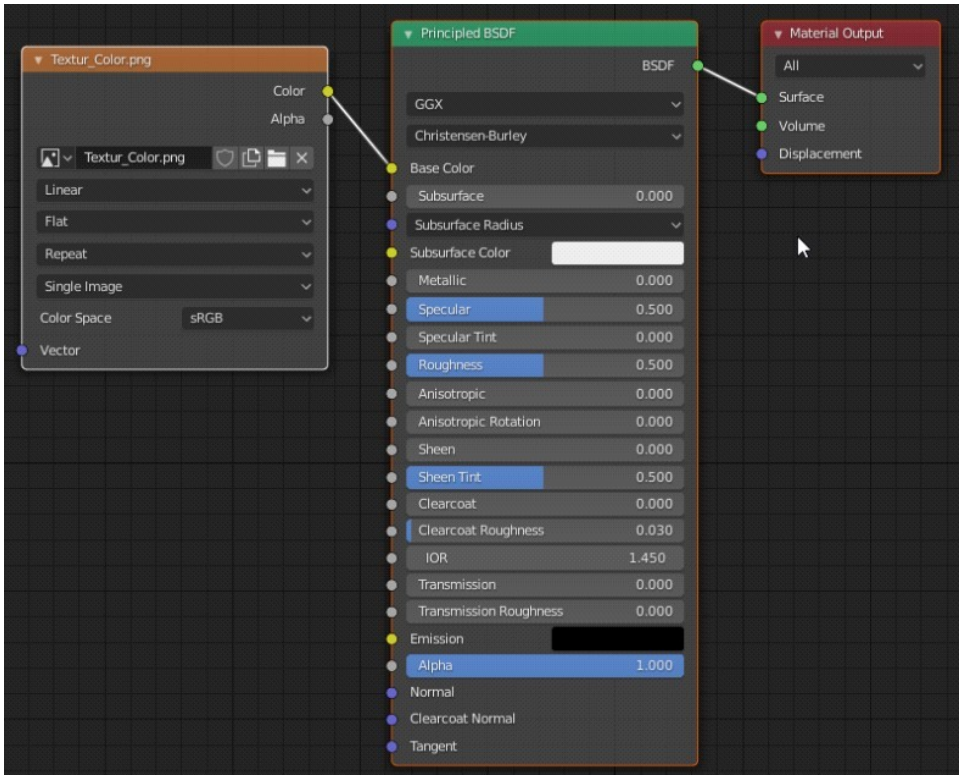
The new possibilities are shown here using an example in Blender

Let's start with Color, Metallic and Roughness, which can be used together or individually. Emissive is dealt with in a separate chapter.

Let's take a simple basic texture for example: Texture_Color.png, in the size of 64 x 64 pixels.



We divide this virtually into 3 sections. In Blender, the color is assigned as before. This results in the following setting in the shader editor:



Now the colour is assigned to the faces as usual via the U/V editor. This can be done beforehand The order doesn't matter.

In this case, with a uni-colored sphere, the area can be very small, namely the segment on the far left in the red area.

For the other 2 balls, the faces are used according to the arrangement in segments 2 - 3.

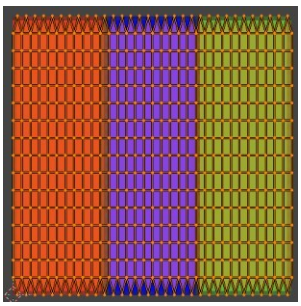
I'll briefly explain how the mapping works in Blender: First, a material must be assigned, as is known.

If it is a texture, the whole thing looks a bit confused because the sides (faces) are not assigned in the corresponding multitexture.

In edit mode you now select the pages to be assigned, or all with "a".

There are still the possibilities to generate different U/V maps with "u" for selected pages, which possibilities that are would lead too far here, but there are countless pages on the Internet where you can read about it.

The assignment of the UV mapping of a sphere then looks like this in Blender:



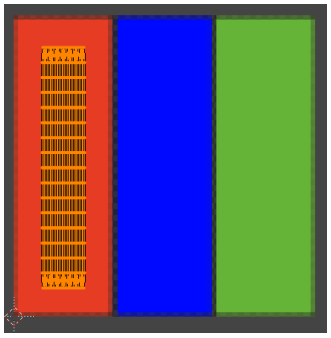
Now the texture assignment can be edited just like in the 3D viewport.

S Scale

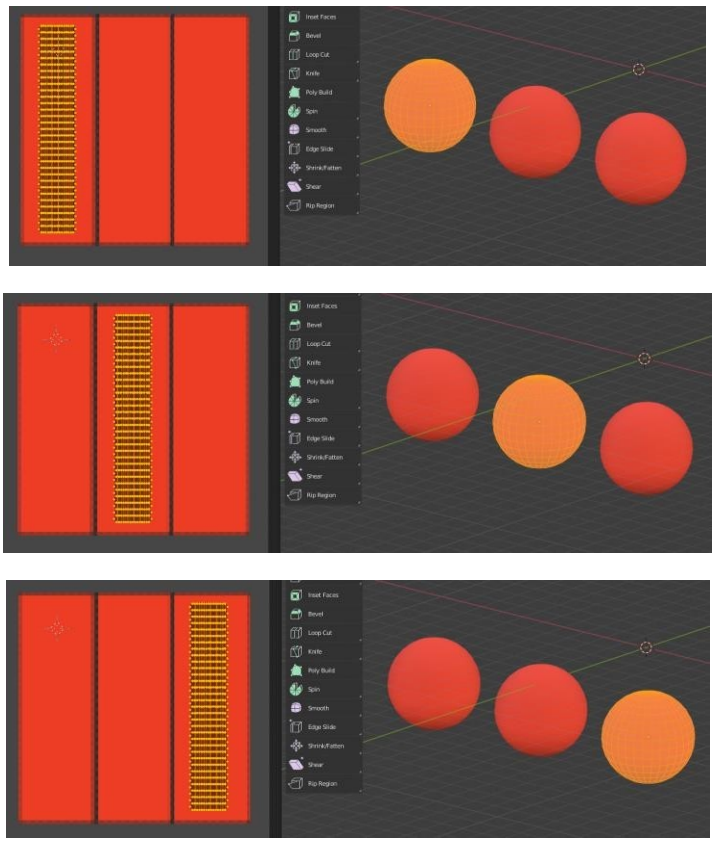
G grave (but only in x,y direction)

or only select and move certain vertices, there are countless possibilities for this.

With this we move and scale the U/V map of the whole sphere into the "Red" area.



Do this as with the other balls, with the difference that the texture has now been changed to "all red" for better understanding.



So far there is no change in the appearance of the balls, they all present themselves the same.

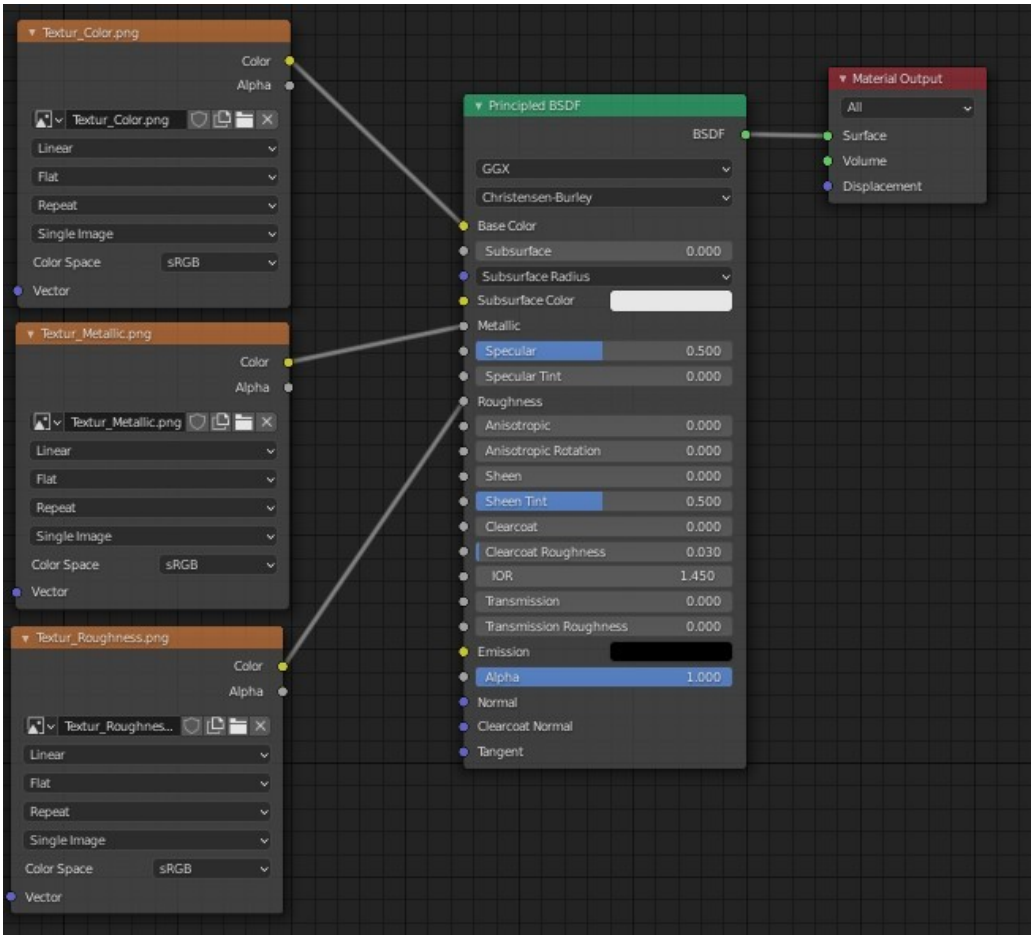
Now we use other previously created textures with the same structure and resolution, although changing the resolution would be possible, it would complicate the procedure, as in the **Texture_Color.png**, in fact:

Texture_Metallic.png Texture_Roughness.png

The naming is irrelevant, in this case it is only for easier understanding.

We now integrate these textures into Blender's shader editor and connect them accordingly with the BSDF shader. (Bidirectional scattering distribution function). Which means something like …. a ray of light hits a surface at a certain angle and is reflected at a corresponding angle. But that's just a side note.

It then looks like this, for example:

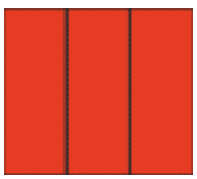


For the additional parameters, these additional textures are required, as already described above, based on the basic texture named **Texture_Color.png**.

Only gray values are used for the metallic or roughness texture, it works with any color, but these have no influence on the percentage values, which results in 256 different values in the RGB system, i.e. 256 gradations in brightness. These gray values are now mapped to the texture area below depending on what is desired, as an example:

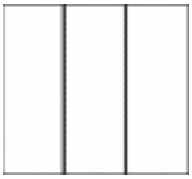
We don't want any effects on the 1st sphere other than the normal color, so all areas (in this case the 1st segment of the square) in the Texture_Metallic.png kept in black except for the roughness effect, which requires a white to get the display as shown in V5.

Texture_Color.png

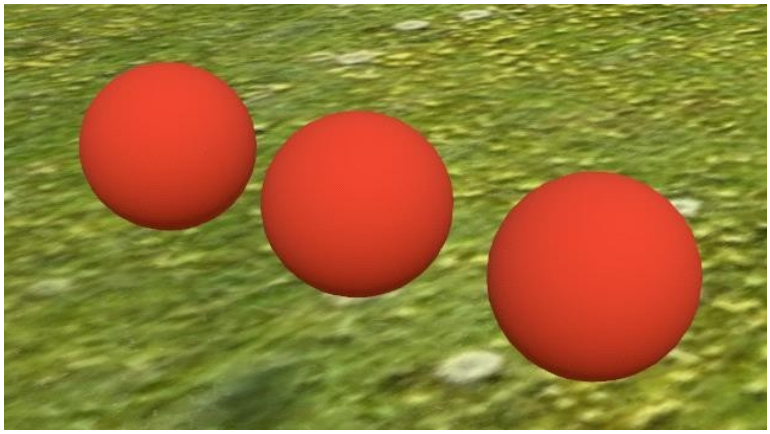


Texture_Metallic.png

Texture_Roughness.png



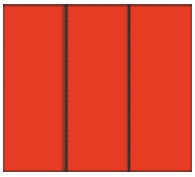
The result in MBS:



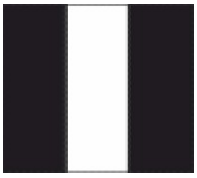
At the moment all spheres appear the same because the same parameters are used for each sphere.

The metallic effect should be used for the 2nd sphere, so the area, in this case the 2nd segment of the square, in the metallic texture is kept completely in white. This results in a 100% metallic effect (with a gray value, the effect is reduced accordingly).

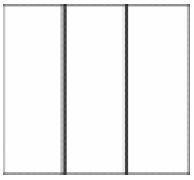
Texture_Color.png



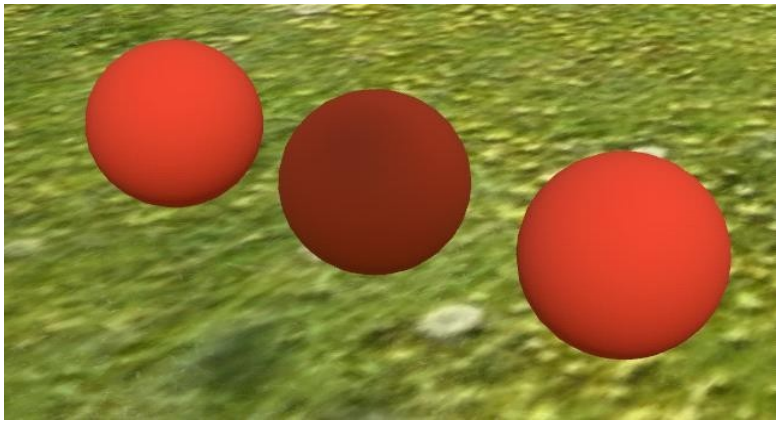
Texture_Metallic.png



Texture_Roughness.png



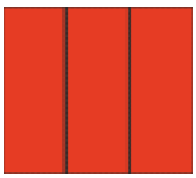
The result in MBS:



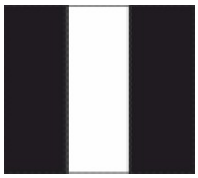
However, this is not quite the result we want because the 100% roughness partially negates the metallic effect, so we need to set the roughness to 0%.

0% roughness means "black". (RGB0,0,0)

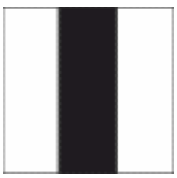
Texture_Color.png



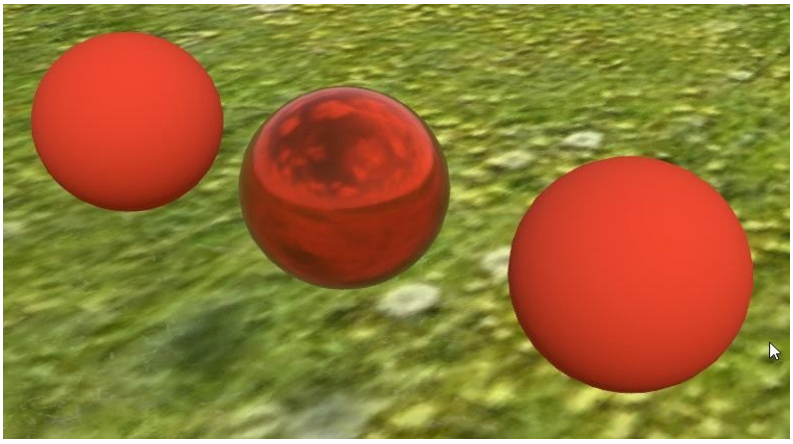
Texture_Metallic.png



Texture_Roughness.png

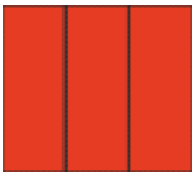


The result in MBS:



This gives a nice metallic effect.
After the 3rd ball has 100% roughness, we change it to black (0%).

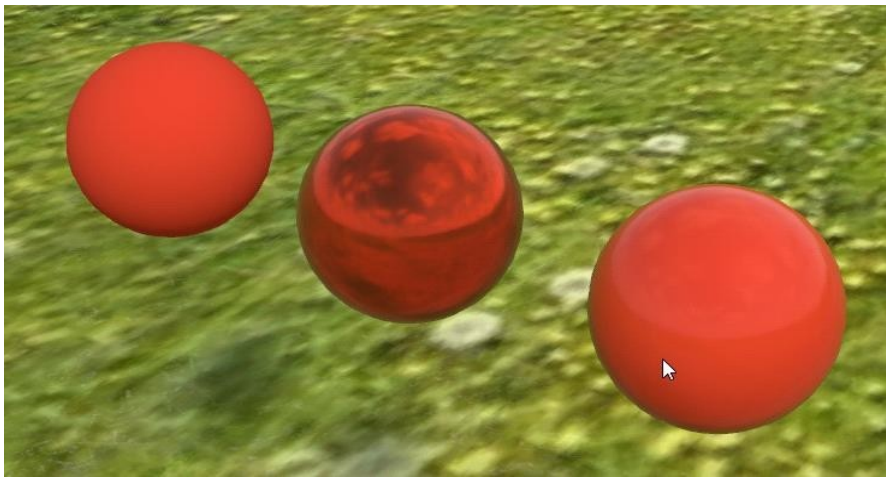
Texture_Color.png



Texture_Metallic.png

Texture_Roughness.png

The result in MBS:



This results in a shiny sphere without too strong a metallic effect, since this is at 0% (black). These examples were only designed as extreme effects, if you change the black/white values to gray values, you can create an almost infinite number of different effects with these color combinations and a little gimmick using only one Material.