Triplanar Shader

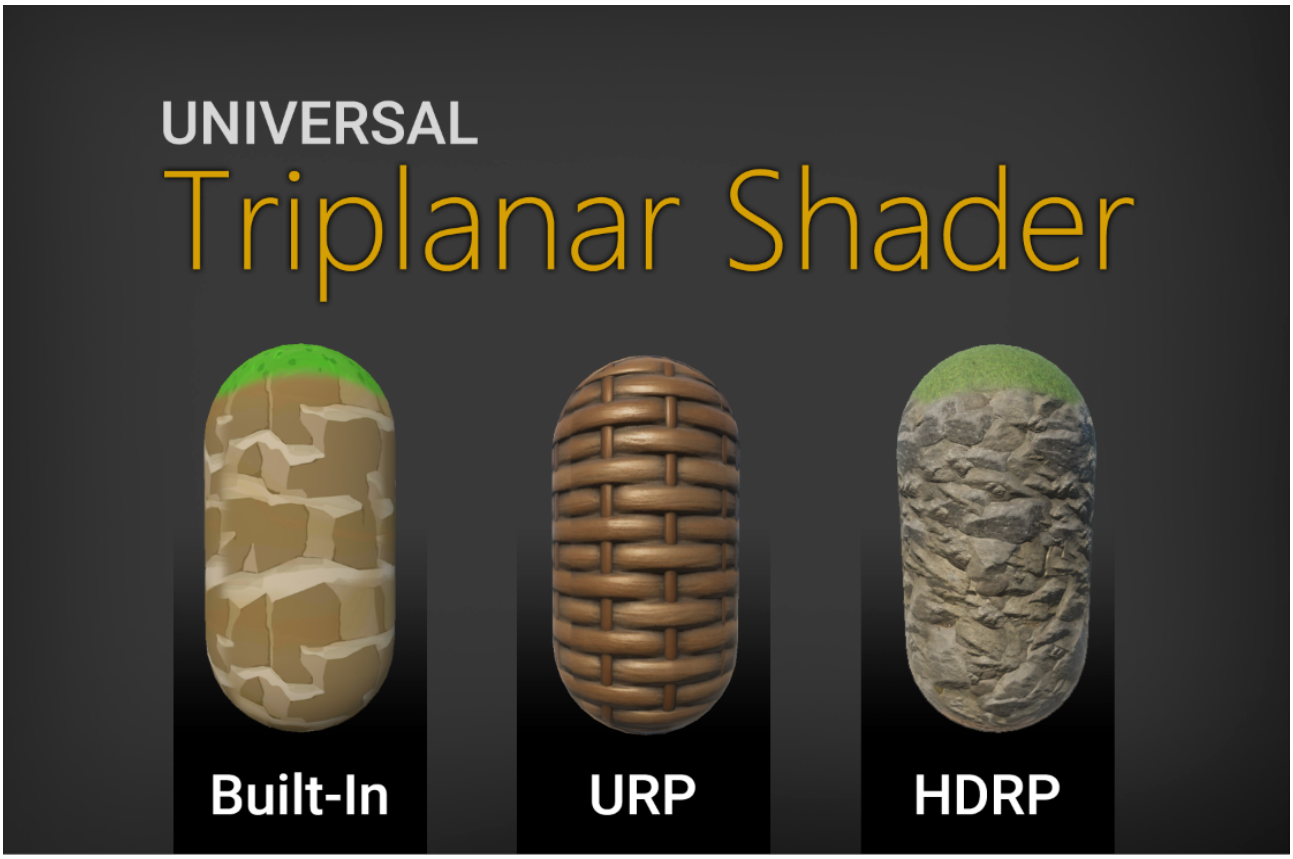


Table of contents

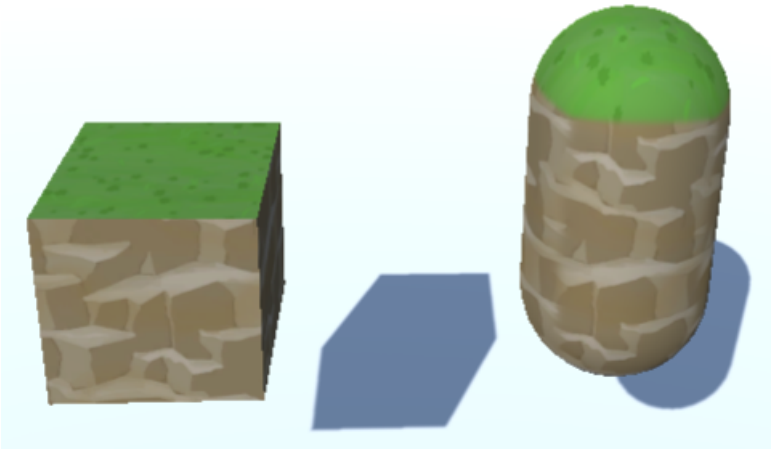
Shader Variations **2**
Simple 2
Complex 2
Shader Parameters **3**
Blend 4
Top Skew 4
Color 4
Tiling & Offset 4
Smoothness 5
Metallicity 5
Normals 5
Occlusion 5
Scripting API **6**
TriplanarMaterial... Components 7
Material Property Setter 8
Material Property Block Setter 9
FAQ **10**
The materials are all bright teal? 10
The materials are all pink? 11
The complex shader does not perform very well on my phone 11

Shader Variations

Not every game needs normal maps or occlusion maps. Especially on mobile it may be better to use a simpler shader.

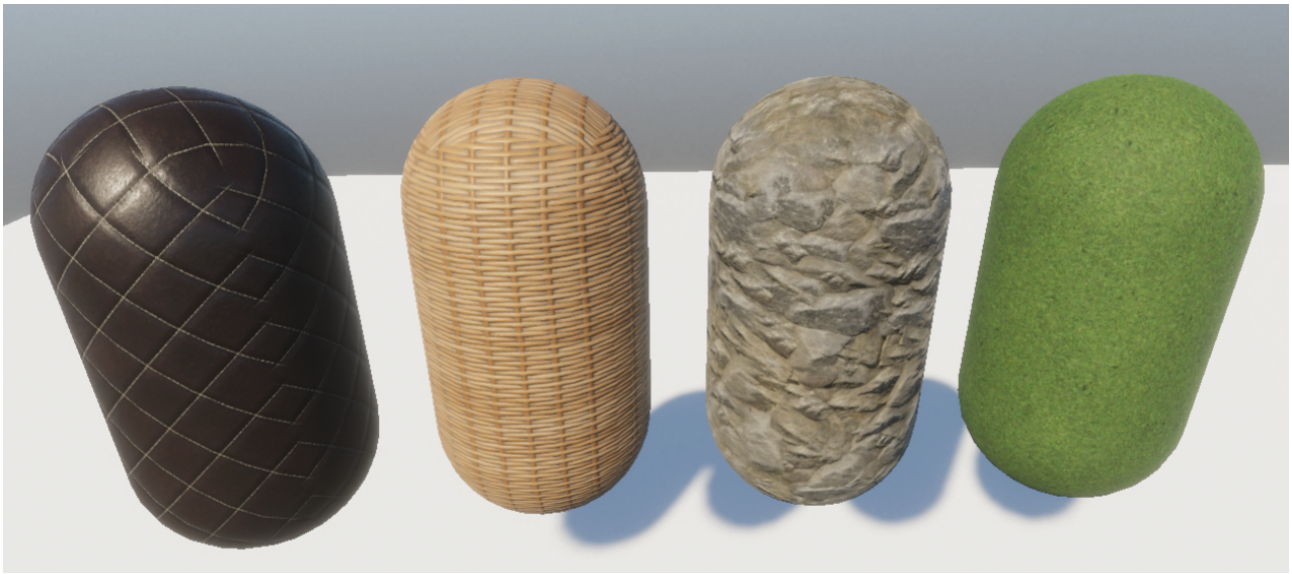
Simple

The simple shader is the best fit for low end devices (low/mid phones). It does not support normal maps.



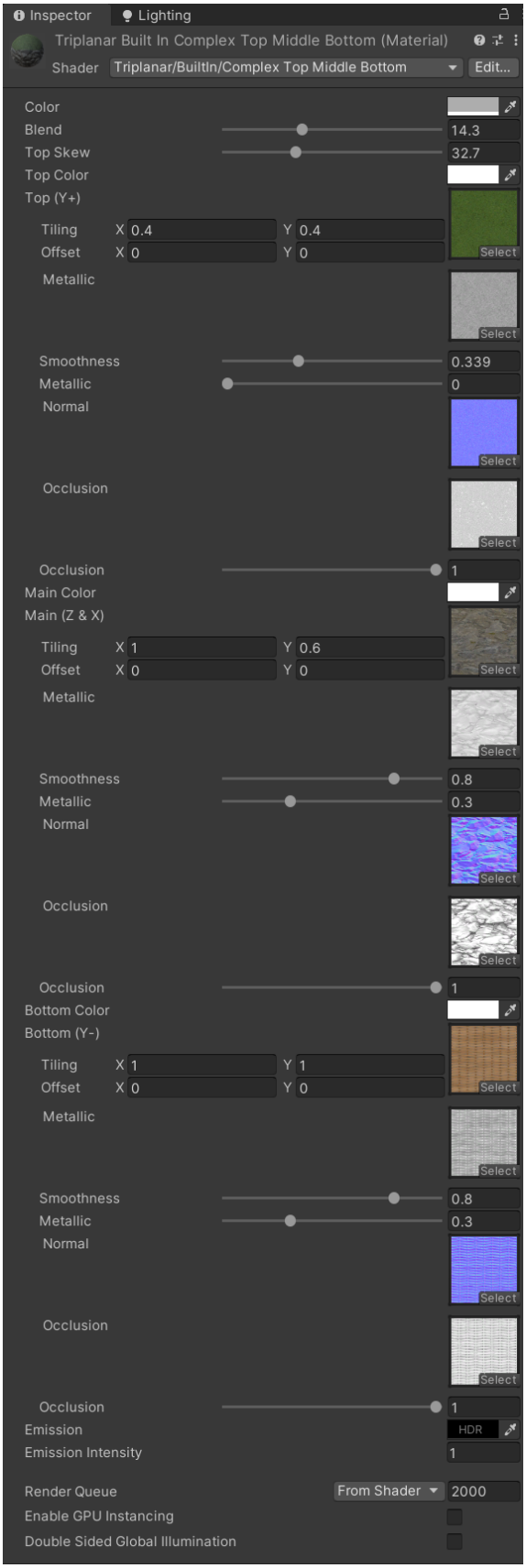
Complex

The complex shader features normal, metallic and occlusion mapping. These are little more taxing on the hardware and thus should only be used on Pc, Consoles or high-end mobile devices.



Shader Parameters

The materials have different parameters depending on which variantion you are using. „Complex“ and „Simple“ materials differ mostly by the extra map textures (normal, metallic, occlusion).



This is how the „complex“ inspector looks like with all textures set.

Blend

Defines how soft the blend between the textures should be (0 = sharpest).


Top Skew

This allows you to „skew" the blending at the top. In the example below it is used to make the grass go further down than it normally would in a triplanar mapped shader. To achieve the maximum effect use it in combination with blend.

NOTICE: Skewing will inevitably cause the texture to distort. After all it still is only projected from the top. Keep that in mind.



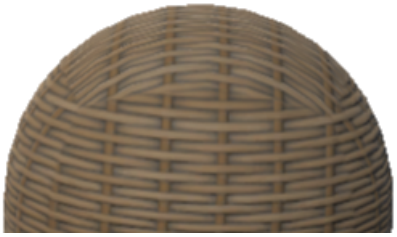
Color

The colors allow you to tint everything or parts of the material.



Tiling & Offset

Tiling and Offset will scale and offset the texture. HINT: You can animate the offset to simulate liquids for example.




Smoothness

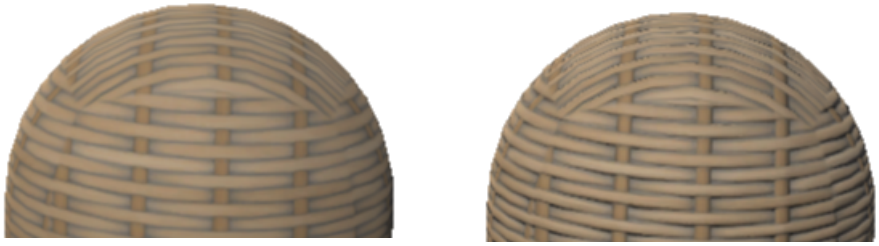
Smoothness is controlled by two properties. A) by the ALPHA channel of the metallic map and B) by the smoothness slider.

Metallicity

Metallicity is controlled by two properties. A) by the RED channel of the metallic map and B) by the metallic slider.

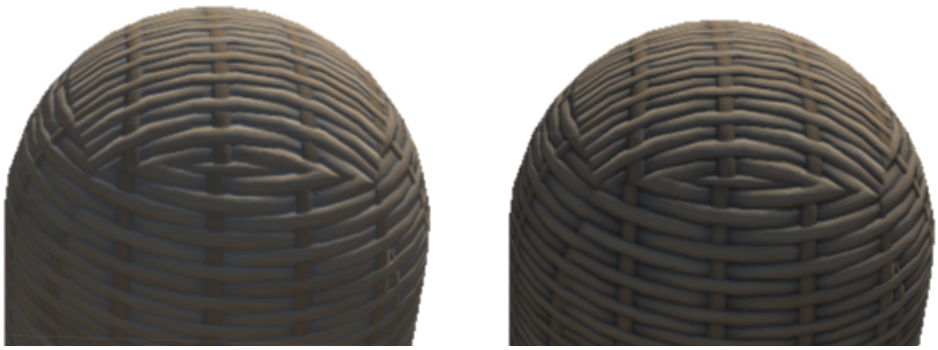
Normals

Unity uses the OpenGL normal map format (Y+). Keep that in mind if you use normal maps from a third party source. If it looks like green light is coming from the top then you're good.



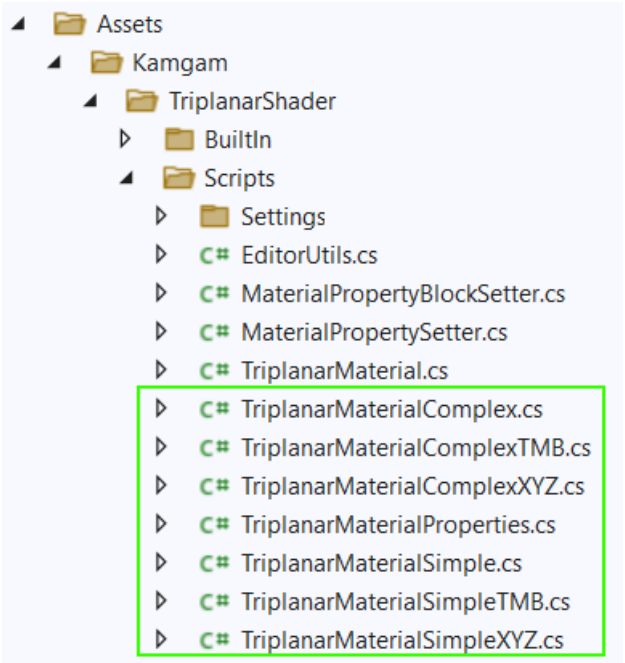
Occlusion

Occlusion is controlled by two properties. A) by the GREEN channel of the occlusion map and B) by the occlusion slider. Occlusion is especially visible in dark areas.



Scripting API

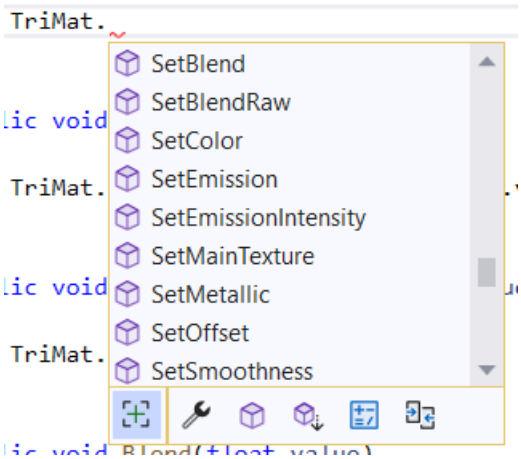
Each shader variation has its very own component. These are called „TriplanarMaterial[Variant]“ where [Variant] is replaced with the name of the variation (like „TriplanarMaterialComplex“ for example).



Changing any property of a material is just one line of code.
Like this:

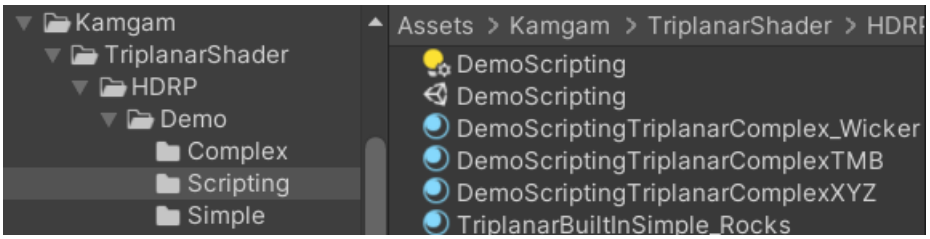
```
public class DemoAPISimple : MonoBehaviour
{
    public TriplanarMaterialSimple TriMat;

    public void RandomColor()
    {
        // Set a random color as the main color
        TriMat.SetColor(new Color(Random.value, Random.value, Random.value, 1f));
    }
```

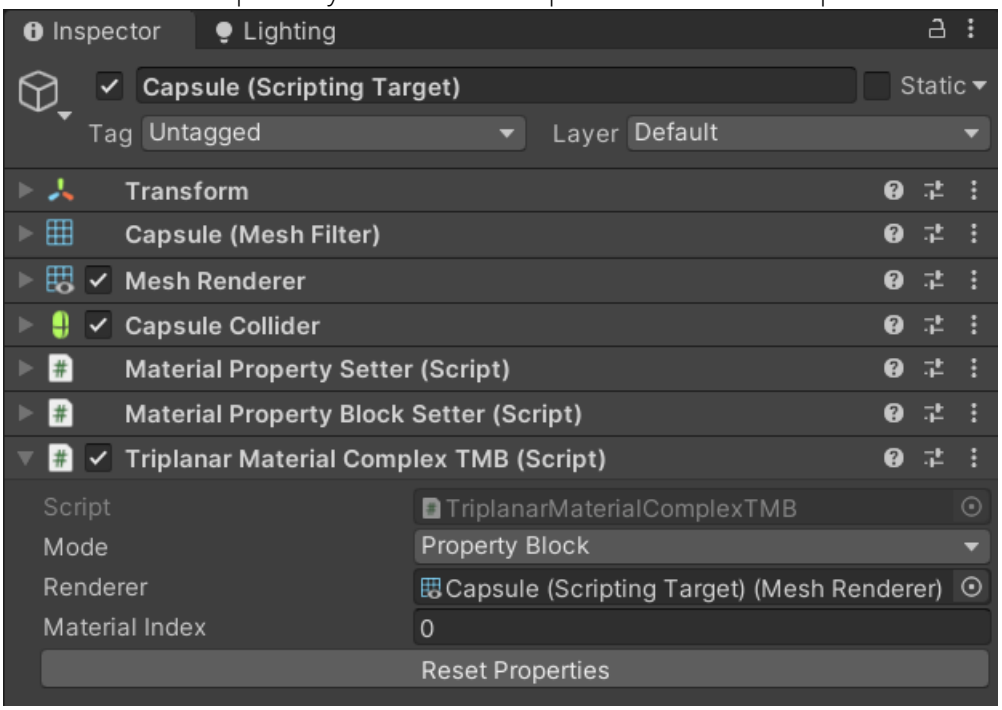


TriplanarMaterial… Components

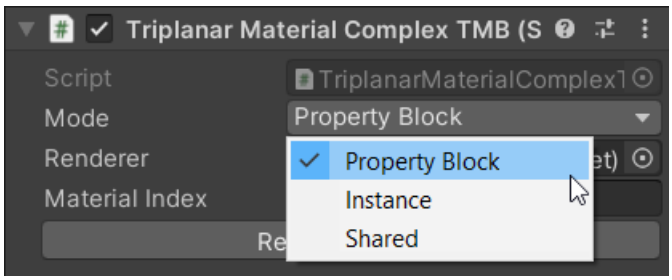
The asset contains a demo scene called **DemoScripting**. It contains an implementation and a UI for each of the TriplanarMaterial* components.



On each of the capsules you will find a TriplanarMaterial* component.



To work it requires a **renderer** and a **material index** (to know which material it should interact with). It also depends on some other components: The MaterialPropertySetter and the MaterialPropertyBlockSetter. These two components are added automatically if you add a TriplanarMaterial* component (details below).

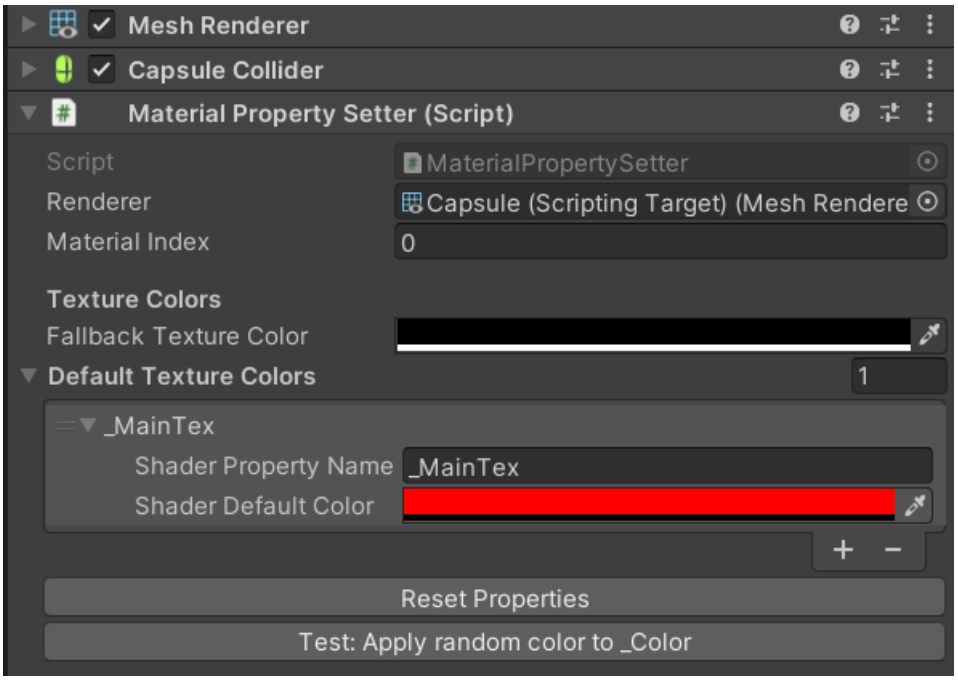


By default Material Property Blocks are used to change the material properties.
You can also choose other modes like creating and modifying an instance of the material (Instance) or modifying the shared material itself (Shared).

Notice that in the Editor any changes to the shared material are NOT reset after the play mode ends. Instead shared mode will edit the material asset itself.

Material Property Setter

The „Material Property Setter" will be used if the „Mode" in the TriplanarMaterial is set to either „Instance" or „Shared". If it is set to „Property Block" (the default) then the „Material Property BLOCK Setter" is used instead (see below).



To work it requires a **renderer** and a **material index** (to know which material it should interact with). You can have multiple of these per object.

You can define a default color for each texture in the material (these are used if a texture gets set to null). In the example above a red color is defined as default in case the „_MainTex" texture is set to null.

For more details on how it works please refer to the in-code documentation.

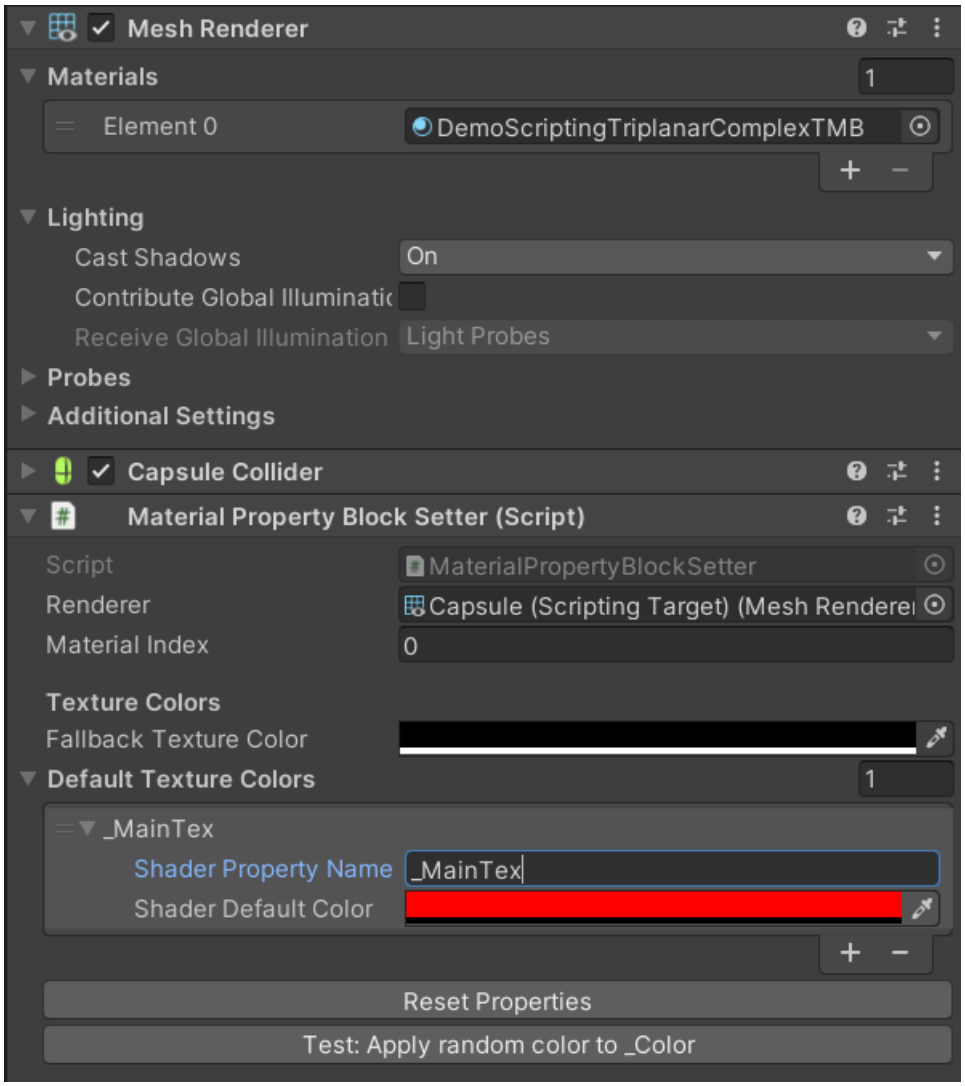
Material Property Block Setter

Material Property Blocks are Unitys way of reducing draw calls for objects with the same material yet different material properties (color for example).

From the Docs: *„Use it in situations where you want to draw multiple objects with the same material, but slightly different properties. For example, if you want to slightly change the color of each mesh drawn. "*

All the **TriplanarMaterial* components** use MaterialPropertyBlocks by default. To do that they use this component. It's an abstraction of the Unity API.

If you're only using the TriplanarMaterial* components then most likely you will never have to interact with it directly. It looks like this:

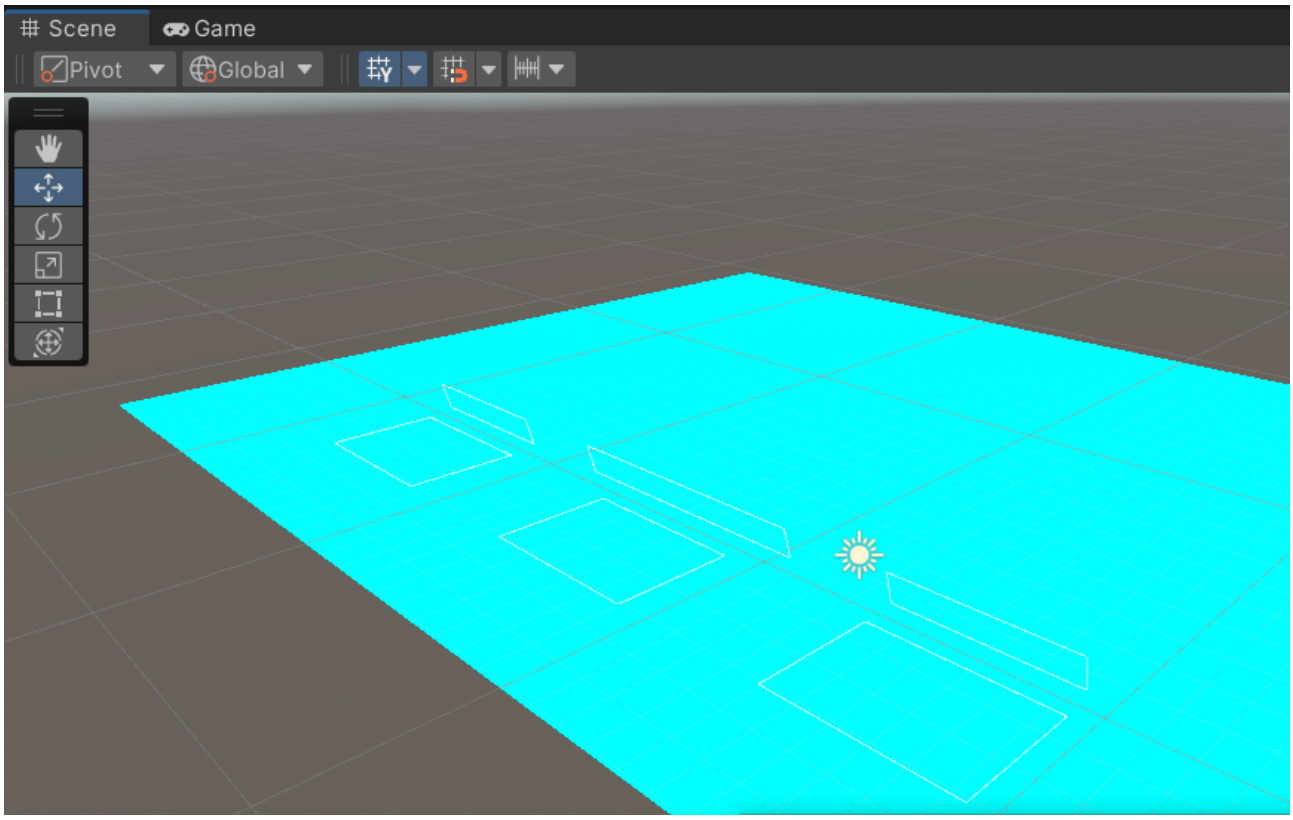


To work it requires a **renderer** and a **material index** (to know which material it should interact with). You can have multiple of these per object. - You can define a default color for each texture in the material (these are used if a texture gets set to null). - For more details on how it works please refer to the in-code documentation.

FAQ

The materials are all bright teal?

If you load the demo scenes for the first time they may look like this:



This is normal.

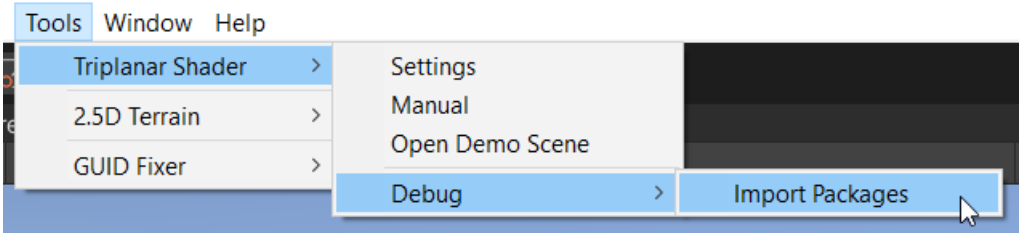
It means that the shaders are still compiling. This can take a while. Especially in the HDRP.

Sometimes the editor window just does not update by itself. Try moving the camera around in the scene or click the light source to trigger a scene update.

The materials are all pink?

This is probably due to the materials not machting your current render pipeline. This should never happen as the tool imports the right materials after installation automatically. Though it may happen if you actively switch the render pipeline after installation.

If it does happen then you can attempt an automatic fix by calling Debug > Import Packages.



The complex shader does not perform very well on my phone.

Having normal, metallic, albedo and occlusion textures can be taxing. Depending on your target device this can be too much for low to mid range mobile devices.

Try the „Simple" shaders instead. You can of course also edit the Complex shaders. They are documented in code so removing things like occlusion maps should be doable. - If you need any help then please don't hesitate to ask in the Unity forum.

Created using assets from ambientCG.com (CC0).

Thanks to ambientCG for all the awesome sample textures <3

They are license under the Creative Commons CC0 1.0 Universal License, see: https://docs.ambientcg.com/books/website-licensing/page/license-information