Triplanar Shader

UNIVERSAL 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?	
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).

Inspector	🌻 Lighting				6
	r Built In Compl	ех Тор М	Aiddle E	Bottom (Materia	al) 😧 구보
Shader	Triplanar/Builtln/	/Comple>	: Тор Мі	ddle Bottom	▼ Edit
Color Blend			_		- 14.3
Top Skew			- -		- 32.7
Top Color					ð
Top (Y+)					
Tiling	X 0.4		Y 0.4		
Offset	хо		Y 0		Selec
Metallic					
					Selec
Smoothnes			_		0.339
Normal					
					Selec
Occlusion					
					Selec
Occlusion Main Color					1
Main (Z & X)					12/15/55
Tiling	× 1		VQQ		
Offset	× I X O		Y 0.6		Selec
Metallic	× O		10		
					Selec
Smoothnes	s			_	0.8
Metallic			•		- 0.3
Normal					Salar
Occlusion					Selec
Occlusion					1
Bottom Color				ľ	
Bottom (Y-)					
Tilina	X 1		Y 1		
Offset	X O		YO		Selec
Metallic					
					Selec
Smoothnes					0.8
Metallic			•		0.3
Normal					Selec
Occlusion					Selec
Occlusion					1
Emission					HDR 👌
Emission Inte					1
DanderO				From Cleader	2000
Enable CDL	e			From Shader	2000
Double Sider	I Global Illumineti				
Double Sideo	r olobal hiuminati	011			

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 <u>OpenGL normal map format</u> (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).

4	📄 Asse	ets						
	🔺 🛅 Kamgam							
	🔺 📄 TriplanarShader							
	⊳	P	BuiltIn					
		P	Scripts					
		⊳	Settings					
		⊳	C# EditorUtils.cs					
		⊳	C# MaterialPropertyBlockSetter.cs					
		⊳	C# MaterialPropertySetter.cs					
		Þ	C# TriplanarMaterial.cs					
		⊳	C# TriplanarMaterialComplex.cs					
		⊳	C# TriplanarMaterialComplexTMB.cs					
		⊳	C# TriplanarMaterialComplexXYZ.cs					
		⊳	C# TriplanarMaterialProperties.cs					
		⊳	C# TriplanarMaterialSimple.cs					
		⊳	C# TriplanarMaterialSimpleTMB.cs					
		Þ	C# TriplanarMaterialSimpleXYZ.cs					

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





TriplanarMaterial... Components

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

Kamgam	٠	Assets > Kamgam > TriplanarShader > HDR
 HDRP Demo Complex Scripting Simple 		 DemoScripting DemoScripting DemoScriptingTriplanarComplex_Wicker DemoScriptingTriplanarComplexTMB DemoScriptingTriplanarComplexXYZ TriplanarBuiltInSimple_Rocks

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

Capsule (Scripting Target) Static Tag Untagged Layer Default Transform Image: Imag	Inspector			а	÷		
Tag Untagged Layer Default Image: Transform Image: Transform Capsule (Mesh Filter) Image: Transform Image: Capsule Collider Image: Transform Image: Capsule Collider <td< td=""><td>Capsule (Scripting Ta</td><td colspan="6">🕞_ 🔽 Capsule (Scripting Target)</td></td<>	Capsule (Scripting Ta	🕞_ 🔽 Capsule (Scripting Target)					
Image: Script Mode Transform Image: Script Mode	Tag Untagged	✓ Layer Default			•		
▶ III Capsule (Mesh Filter) ♀ II ▶ III Mesh Renderer ♀ III ▶ III ♀ Capsule Collider ♀ III ▶ III Material Property Setter (Script) ♀ IIII ▶ III Material Property Block Setter (Script) ♀ IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	🕨 🙏 🛛 Transform		0	4			
Mesh Renderer Image: state of the sta	E Capsule (Mesh Filter)		0				
Image: Capsule Collider Image: Capsule Collider Image: Material Property Setter (Script) Image: Capsule Collider Image: Material Property Setter (Script) Image: Capsule Collider Image: Material Property Block Setter (Script) Image: Capsule Collider Image: Material Property Block Setter (Script) Image: Capsule Collider Image: Capsule Collider Image: Capsule Collider	🕨 🔣 🖌 Mesh Renderer		0				
▶ # Material Property Setter (Script) ♀ ↓ : ▶ # Material Property Block Setter (Script) ♀ ↓ : ♥ # ▼ Triplanar Material Complex TMB (Script) ♀ ↓ : Script ■ TriplanarMaterialComplexTMB ⊙ Mode Property Block ▼	🕨 🎚 🗹 Capsule Collider	0					
▶ # Material Property Block Setter (Script) ♀ ↓ : ▼ # ▼ Triplanar Material Complex TMB (Script) ♀ ↓ : Script ■ TriplanarMaterialComplexTMB ⊙ Mode Property Block ▼	🕨 🗯 🛛 Material Property Sette	r (Script)	0	- 1 -			
Triplanar Material Complex TMB (Script) Image: Complex TMB Script Image: Complex TMB Mode Property Block	🕨 🗯 🛛 Material Property Block	Setter (Script)	0	цт.			
ScriptTriplanarMaterialComplexTMBModeProperty Block	🔻 # 🖌 Triplanar Material Comp	olex TMB (Script)	0	÷.			
Mode Property Block	Script	TriplanarMaterialComplexTMB					
	Mode	Property Block			▼		
Renderer 🛛 🖾 Capsule (Scripting Target) (Mesh Renderer) 💿	Renderer	思Capsule (Scripting Target) (Mesh Renderer)					
Material Index 0	Material Index	0					
Reset Properties		Reset Properties					

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 MaterialProperty<u>Block</u>Setter. These two components are added automatically if you add a TriplanarMaterial* component (details below).



By default <u>Material Property Blocks</u> 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).

<u>Notice</u> 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).

🕨 🔣 🖌 Mesh Renderer		0		:
🕨 🎚 🗹 Capsule Collider		8		
🔻 # 🔹 Material Property Sett	er (Script)	0		
Script	MaterialPropertySetter			
Renderer	₿Capsule (Scripting Target) (Mesh R	end	ere	\odot
Material Index	0			
Texture Colors				
Fallback Texture Color				ø
Default Texture Colors		1		
=▼_MainTex				
Shader Property Name	_MainTex			
Shader Default Color				*
		+	-	
	Reset Properties			
Test: Ap	ply random color to _Color			

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

<u>Material Property Blocks</u> are Unitys way of reducing CPU memory usage for objects with the same material yet different material properties (color for example).

<u>From the Docs:</u> "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:

🔣 🖌 Mesh Renderer		8	τ ι	:
Materials		1		
Element 0	DemoScriptingTriplanarComplexTM	1B	0	\mathbf{D}
		+		Г
Lighting				
Cast Shadows	On			▼
Contribute Global Illumination				_
Receive Global Illumination	Light Probes			•
Additional Settings				
		~		
		8	÷	:
# Material Property Block	k Setter (Script)	8		:
Script	MaterialPropertyBlockSetter			\odot
Renderer	₿Capsule (Scripting Target) (Mesh Re	nd	erei	0
Material Index	0			
Texture Colors				
Fallback Texture Color				Ø.
Default Texture Colors		1		
=▼_MainTex				
Shader Property Name	_MainTex			
Shader Default Color				*
		+	-	
	Reset Properties			
Test: Ap	ply random color to _Color			

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.

	Tools	Window Help					
5	Ti	iplanar Shader	>	Settings			
·e	2.	5D Terrain	>	Manual	Manual		
L	G	UID Fixer	>	Open Demo Scene			
				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 <u>Unity forum</u>.

Created using assets from <u>ambientCG.com</u> (CC0).

Thanks to <u>ambientCG</u> for all the awesome sample textures <3

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