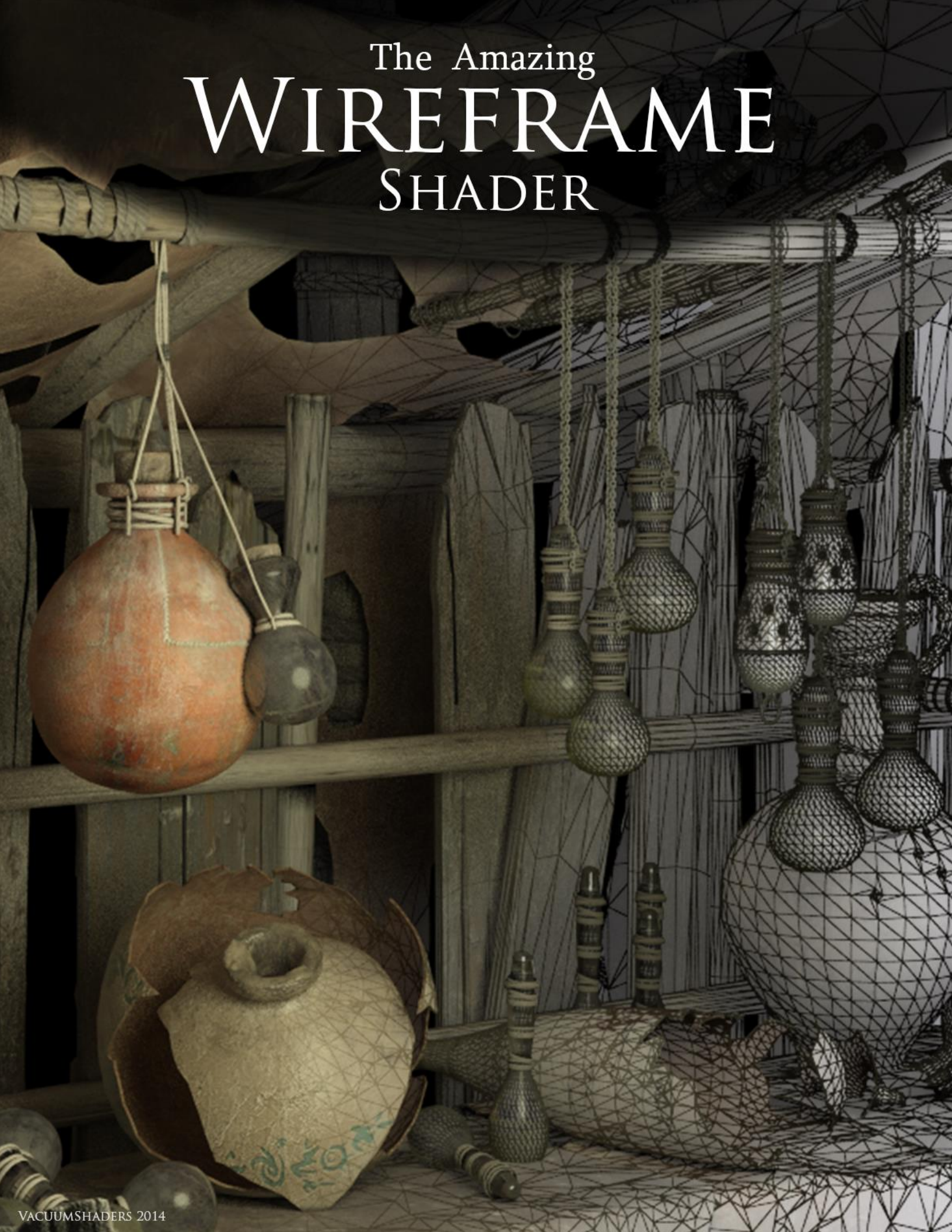
The Amazing WIREFRAME SHADER



VACUUMSHADERS 2014

Thanks for purchasing **The Amazing Wirframe** shader.

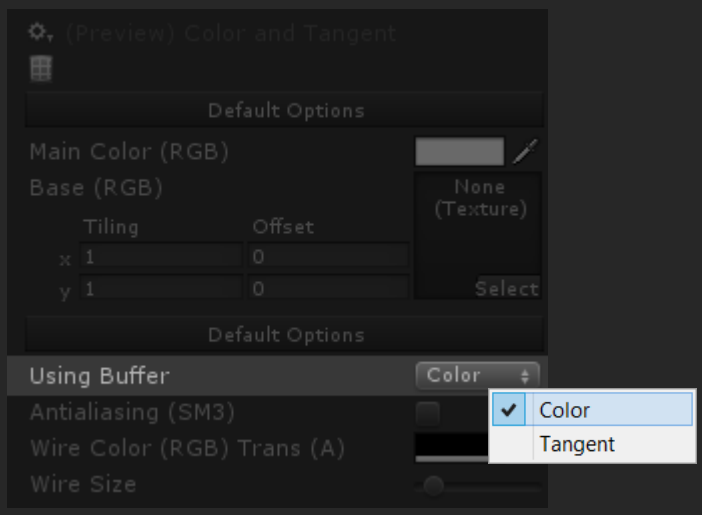
Please consider leaving a review or just rating the asset if you find it useful.

Thanks!

Introduction

Create new material and set wireframe shader **VacuumShaders/The Amazing Wireframe/(Preview) Color and Tangent**. Apply material to the mesh.

For properly rendering of wireframe, shader requires mesh with barycentric coordinates built into color or tangent buffers. Within material editor you choose from where shader reads these data.



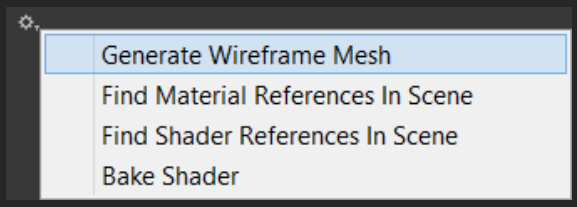
If mesh has no barycentric coordinates it is possible to generate them inside editor or in game mode.

First one will create new asset file same as original but with required data stored inside. Wireframe will be visible inside editor and game mode.

Second one will generated required data only in game mode.

Generating barycentric coordinates inside editor

1. Click on the gear icon on the top-left side of material editor and choose **Generate Wireframe Mesh**

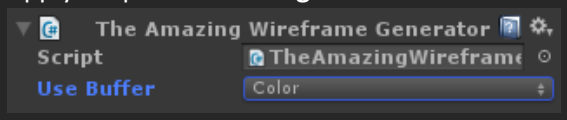


2. Set mesh name and buffer where barycentric coordinates will be saved

3. Hit **Generate** button. New asset file will be created and replace currently used mesh.
4. Now wireframe will be rendered correctly.

Generating barycentric coordinates in Game mode

1. Apply script **The Amazing Wireframe Generator** to the mesh and choose which buffer use.



2. Enter game mode to see wireframe.

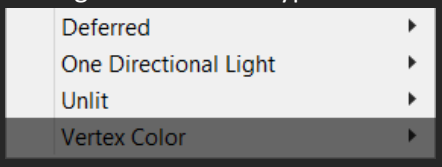
Script generates wireframe by using `WireframeManager.GetWire` function.
`WireframeManager` stores instance ID of the meshes whose wireframes has been generated, to avoid wireframe calculation for meshes with the same ID.
`WireframeManager.RemoveMesh` - removes mesh ID from the `WireframeManager`

For direct wireframe calculation use - `WireframeGenerator.Generate`

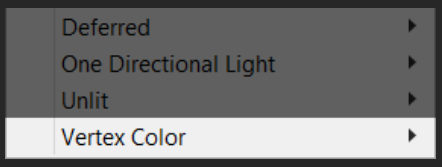
Which buffer use?

If data is saved inside color buffer, only mesh's vertex color will be lost, other data will remain untouched including tangents, which are necessary for skinned mesh animations.
Saving data inside tangent buffer will overwrite original tangents. Choose this option for vertex color shaders only.

Package contains two type of shaders:



Deferred, One Directional Light and Unlit shaders can be used only if barycentric coordinates are saved inside color buffer.



Vertex Color shaders can be used only if barycentric coordinates are saved inside tangent buffer.

A little bit more about material editor

Wireframe shader editor has three sections

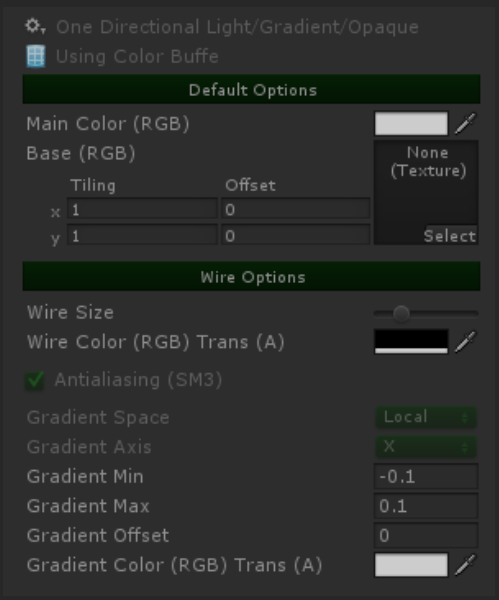
- Toolbar - shows shaders type and which buffer currently selected mesh is using.
- Default options - give controls to the default visual elements like: Base Texture, Color, Specular, Reflection, Bump, Alpha Cutout.
- Wire options - contains all features required for wireframe rendering. Most parameters here are optional. Effects are excluded from shader calculations, if they are turned off.

Shader baking

Most of **Wireframe** shader files contain lots of variants inside. Using these variants gives possibility to have many shaders and effects inside one file (instead of writing hundreds of shader files). Turning on/off optional parameters automatically switches shaders inside material editor and users even doesn't notice that.
Backside of these variants is that they increase shader file size (that goes into build) and compilation time. Some of mobile devices may even crash because of huge memory required for shader compilation.

For example: One Directional Light / Gradient.shader uses 4614 variants and is 33MB in size.

Before building project or testing on mobile device it is required to bake **Wireframe** shaders.
Baking removes all variants from shader and generates one shader file with all used optional parameters as one effect.
Baked shaders go to Baked submenu inside VacuumShaders/The Amazing Wireframe/ shaders choosing menu and is automatically assigned to the material.
After baking it is not possible to turn on/off optional parameters.



For example: This is baked One Directional Light / Gradient.shader.
All unused optional parameters are removed.
Optional parameters that were turned on before baking cannot be turned off.

Now it contains only 30 variants (Unity system variants) and is 190KB after compilation.

Baked shader has green title bars.

Note:
Shader baking has no impact on their performance, just removes variants and reduces file size.
How much and what optional parameters are used does not matter.

Shader baking is available within gear icon menu