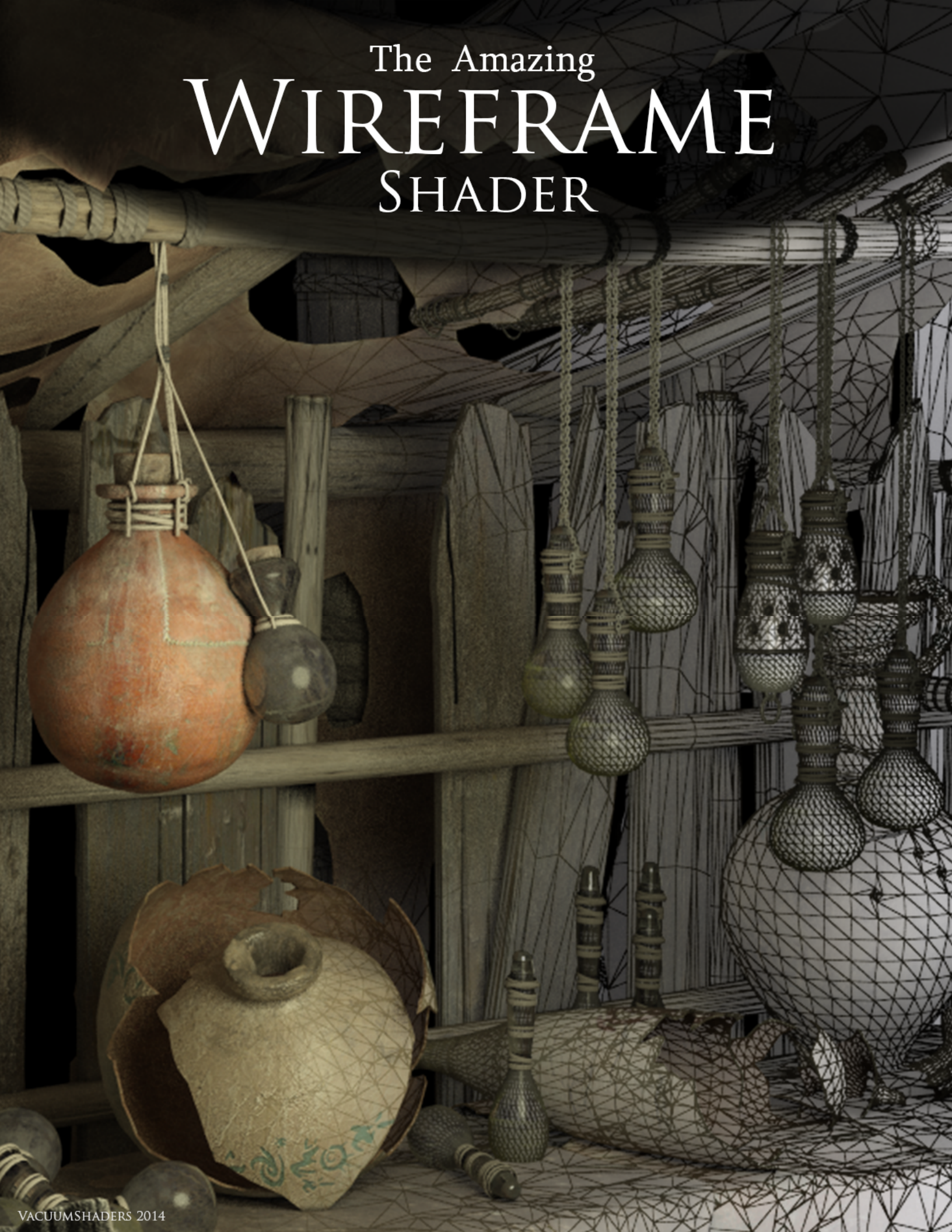


The Amazing
WIREFRAME
SHADER



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

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

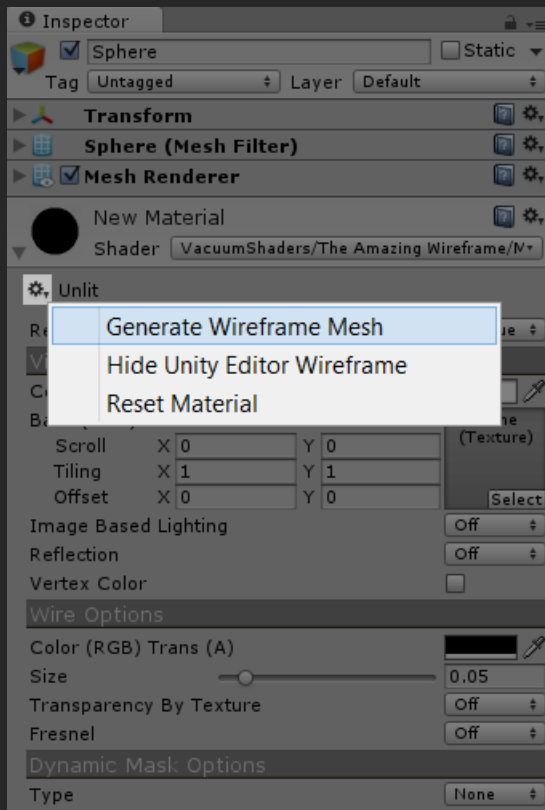
Thanks!

Introduction

Create material and set wireframe shader [VacuumShaders/The Amazing Wireframe/Mobile/Unlit](#). Apply material to mesh.

Mesh will be rendered completely black (color of wireframe by default), it means that mesh has no barycentric coordinates for wireframe rendering.

For generating mesh with barycentric coordinates select mesh inside scene (Hierarchy window) and inside material editor click on 'Gear' icon, choose [Generate Wireframe Mesh](#) option.



Generated mesh asset file will be saved inside [Assets/\(Temporary\)/Wireframe](#) folder and automatically replace original mesh from MeshFilter or SkinnedMeshRenderer.

Now shader will render wireframe.

For multiple mesh conversion can be used [Menu/Window/VacuumShader/Wireframe Generator](#) tool.

Note:

- Barycentric coordinates are generated per-vertex and saved inside mesh UV0 buffer (under uv[3] and uv[4] indexes).
- Barycentric coordinates saved inside UV0 buffer does not disturb texture coordinates saved inside the same buffer (under uv[0] and uv[1] indexes).
- During generating barycentric coordinates mesh vertex count may increase and if it is more then 65.000 – conversion will fail.

Wireframe can be rendered directly from texture (avoiding heavy calculations inside shader), for that wireframe must be exported using [Menu/VacuumShaders/Wireframe Texture Exporter](#).



Source **GL** renders wireframe for all meshes, while **Baked** source only for meshes with barycentric coordinates inside.

Texture is exported in PNG format without black background and only with transparent wireframe data.

Mesh with barycentric coordinates can be generated at run-time.

Two run-time functions are available within `VacuumShaders.TheAmazingWireframeShader` namespace:

`static public Mesh Generate(Mesh _origMesh)` – Generates optimized mesh (used by editor tools).

`static public Mesh GenerateFast(Mesh _origMesh)` – Generates non-optimized mesh. Is the fastest but mesh vertex count has limit – 21.000

Check [Example_5 \(Runtime\)](#) example scene.

Dynamic Mask

Dynamic mask is world-space modifier defining areas where wireframe is visible:

- Plane – Wireframe visibility is defined by plane equation and takes into account two parameters:

1. Plane world-space position `half3 _V_WIRE_DynamicMaskWorldPos`
2. Plane world-space normal `half3 _V_WIRE_DynamicMaskWorldNormal`

(Wireframe is visible only on positive side of the plane)

- Sphere - Wireframe visibility is defined by sphere equation and takes into account two parameters:

1. Sphere world-space position `half3 _V_WIRE_DynamicMaskWorldPos`
2. Sphere radius `half _V_WIRE_DynamicMaskRadius`

(Wireframe is visible only inside sphere)

Dynamic mask parameters are not displayed inside material editor and can only be updated from custom scripts.

Check [Example_2 \(Mask - Plane\)](#) and [Example_3 \(Mask - Sphere\)](#) example scenes.

Wireframe projector

Effect completely relies on Unity Projector effect and uses [VacuumShaders/The Amazing Wireframe/Projector](#) shader.

Wireframe is projected on meshes with any type of shader, but mesh itself must contain barycentric coordinates.

Check [Example_4 \(Projector\)](#) example scene.

Geometry Shaders

Geometry shaders do not need mesh with barycentric coordinates (including Projector).

Just assign shader to a mesh and it will be rendered with correct wireframe effect.

Works with all mesh types, has no limit on vertex count, supports all mesh features.

Requires device with GeometryShaders support.