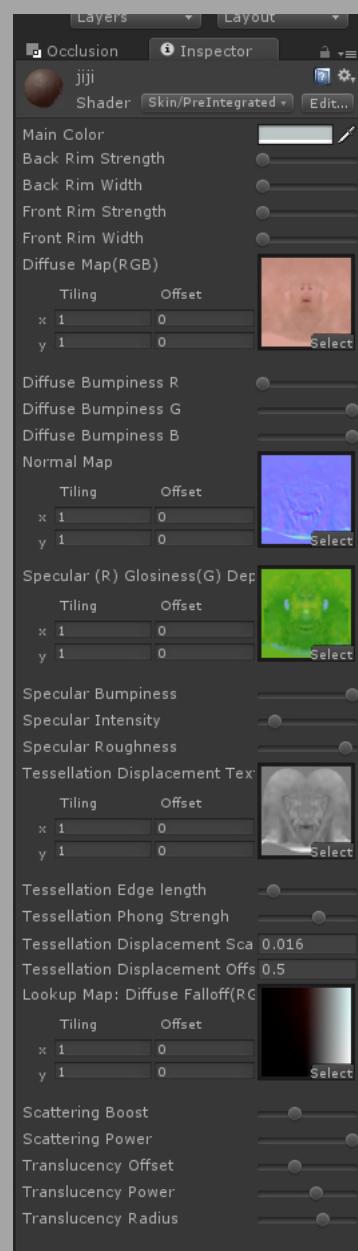
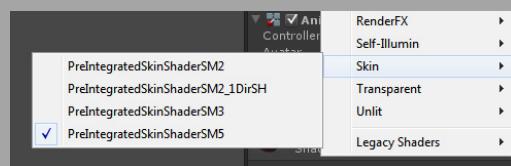


Skin Shader for Unity3d Game engine

by Maciej Kacper Jagiełło

Usage Instructions

1. General
2. Diffuse color
3. Front and Back Rims
4. Bump map
5. Diffuse Bumpiness
6. Specular Glossiness and SSS Depth maps
7. Specular sliders
8. Tessellation
9. Diffuse Falloff map
10. Translucency



1. General

Requirements

This shader works in Unity Pro as well as in the Unity Free, but results are radically different because realistic skin rendering depend heavily on shadows which the free version lacks.

It's been tested on Unity 4.0 and to a limited degree on 3.5.

It should work on large variety of graphics configurations, but features are limited by graphics card capabilities.

Shader Model 5 supports all features: tessellation with displacement map and/or phong smoothing, diffuse scattering with separate weighted normals, fresnel specular reflectance, translucency, rim lighting and real-time shadows from all types of lights.

Shader Model 3 supports all above features except tessellation/displacement.

Shader Model 2 supports only diffuse scattering and specular reflectance without separate normals. Shadows are received only from directional lights.

All versions fallback seamlessly to lower-end ones, except on Unity < 4.0 you need to delete the

PreIntegratedSkinShaderSM5.shader file from the project and use shader model 3 variant directly.

Separately there's a low-end variant which supports fully only one directional light and the rest as spherical harmonics in vertex lighting (PreIntegratedSkinShadersSM2_1DirSH).

Performance

The shader is always rendered in forward mode and the performance depends heavily on the number of lights in the scene. The different shader model variants do influence performance by cutting some features and it could be desirable to choose lower feature-set in some situations. Remember tessellation is not free.

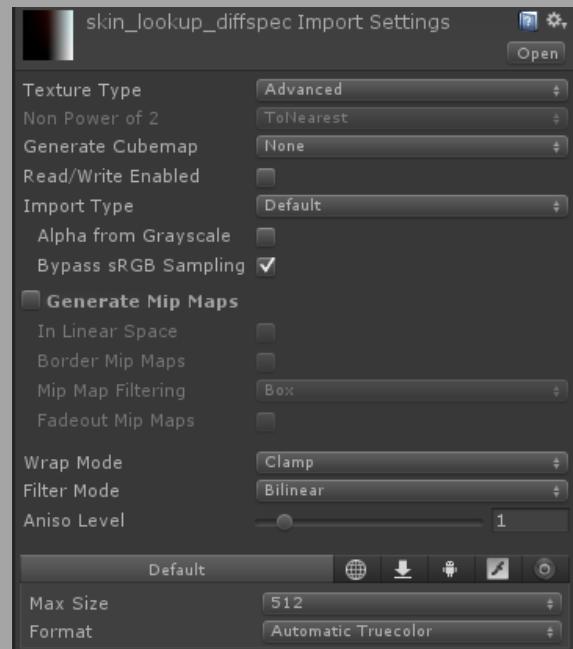
Known issues

With Unity 4.0 when deferred rendering is enabled in player settings the shader model 5 version shows visual artifacts. It's a Unity bug.

The lookup map generation script doesn't apply the settings immediately.

Linear Rendering (Unity Pro only)

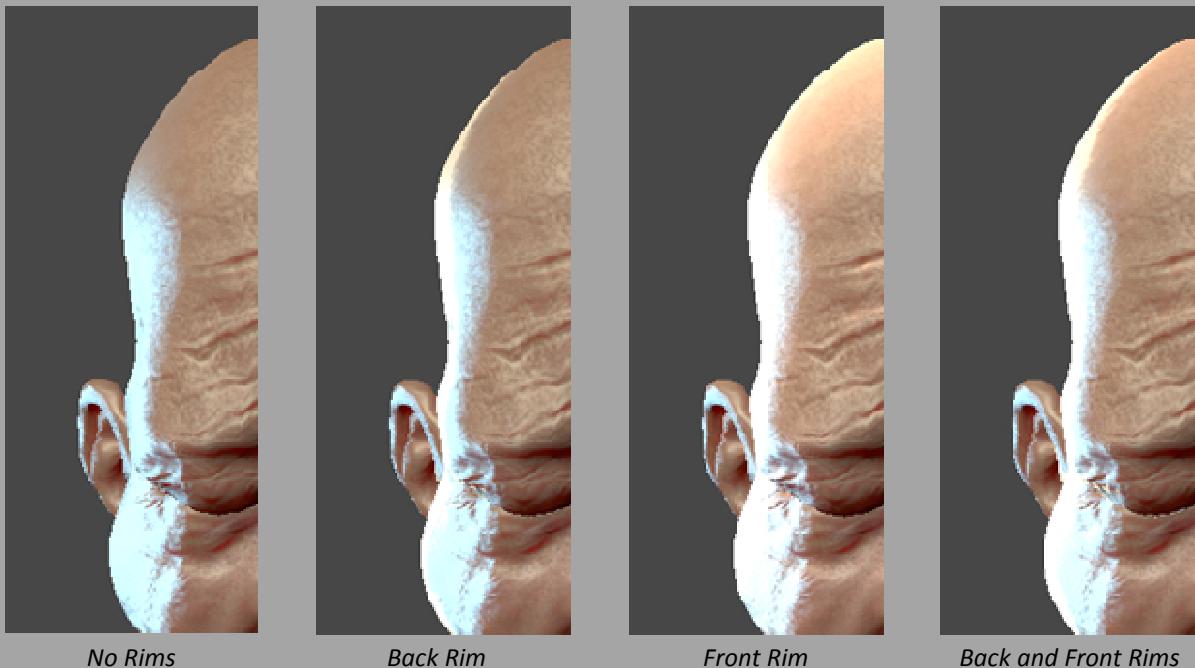
Turning on linear rendering welds much more realistic results, see Unity documentation on the subject. Beware that turning on linear rendering most likely requires adjustments throughout entire project. To take advantage of linear rendering you need to check "Bypass sRGB Sampling" in specular-glossiness-depth map, displacement map and in the lookup map. The last one is already set if imported from asset store or generated with the script.



2. Diffuse color

Main color serves to colorize your diffuse map.

3. Front and Back Rims



No Rims

Back Rim

Front Rim

Back and Front Rims

4. Bump map (Normal map)



Without Normal map

With Normal map

5. Diffuse Bumpiness



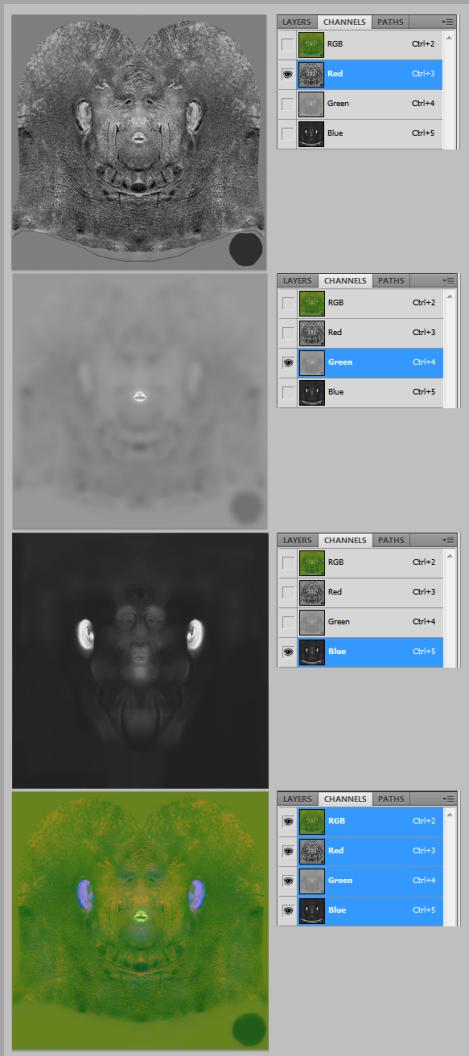
0%



Blue and Green 100%

Changes the look of your Bump details by shifting RGB values of diffuse map, this method has been described by Eric Penner.

6. Specular Glossiness and SSS Depth maps



All three maps are packed in one texture for performance reasons. Specular, glossiness and sss depth maps are stored in Red Green and Blue channels respectively.

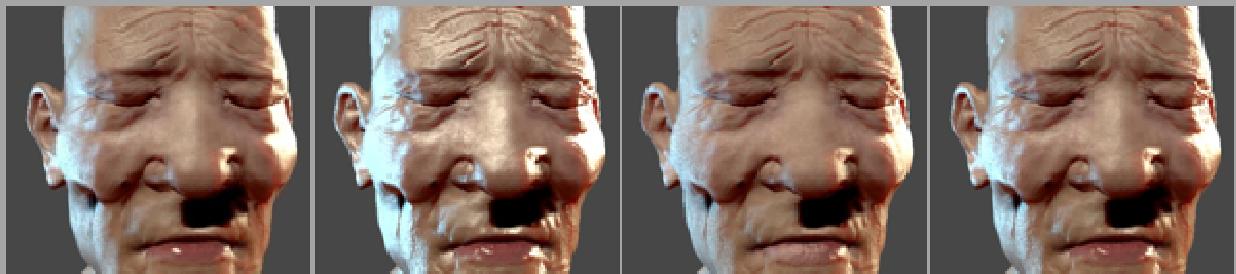
To set your map correctly, you must copy each texture, select correct channel in Photoshop and paste texture with **Ctrl+Shift+V** (paste in place) shortcut.

Tip: most likely the final result should be green, if your result is reddish or bluish, play with brightness and contrast ratios of specular and glossiness maps. Specular map should have strong contrast and overall should be dark in skin folds and middle gray for bumps. Correct glossiness map for skin should have value around 170 of gray, on lips and oily parts should be close to white.

The depth/thickness map can be generated in most 3D packages by baking ambient occlusion with inverted normals.

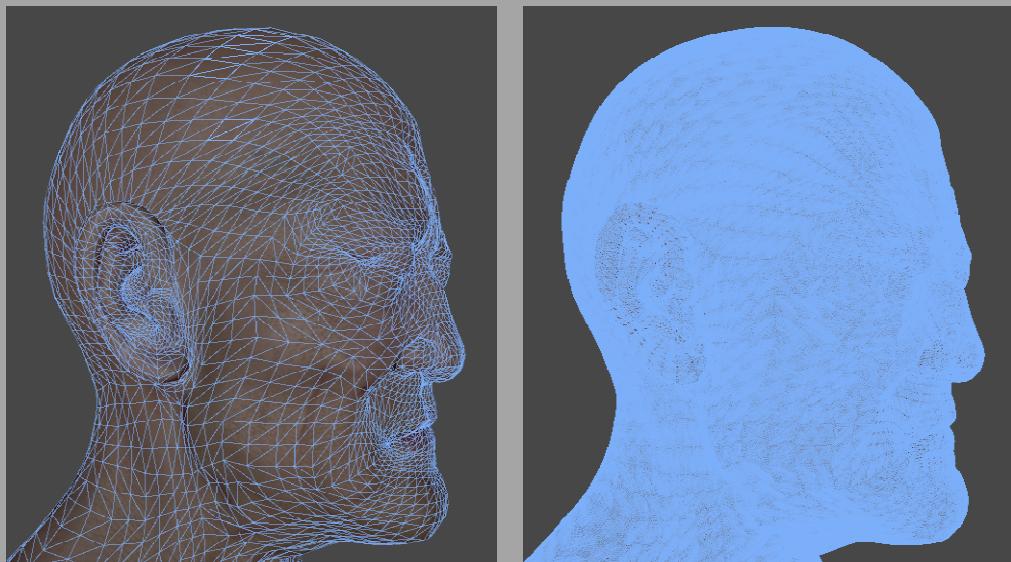
In Vray use vraylight material and for color map, use vray dirt with white for occluded and very dark gray for unoccluded color. Radius around 2cm and with invert normals checkbox checked. Render to texture, save it and tweak it in Photoshop on eyes and lips.

7. Specular Intensity and Bumpiness sliders



<i>Specular Bumpiness</i>	0%	100%	100%	100%
<i>Specular Intensity</i>	70%	70%	25%	25%
<i>Specular Roughness (Gloss)</i>	100%	80%	25%	90%

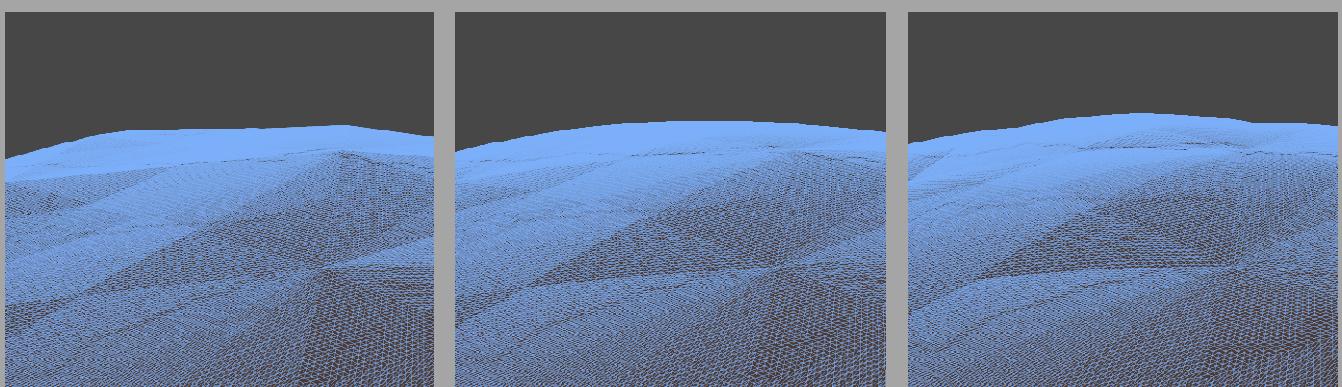
8. Tessellation



Edge length 100%

Edge length 0%

Amount of smoothing is dependent of edge length, shorter means more details



Phong Strength 0%

Phong Strength 50%

Phong Strength 100%

Phong smoothing is basically an edge smoothing, similar to *poly subdivision* from commonly used 3d software, but with one exception, when it's set to 100%, object becomes bulky. So leave it around 50-70%

Now, the most important part of tessellation settings. First select your displacement map, set your scale to something really low such as 0.01, when you see that your object exploded, lower scale until you get result similar from middle image. Now, it still looks bad, so to correct that, there is offset that comes in handy, increase it slowly to get desired result.



Scale 0.01

Scale 0.003

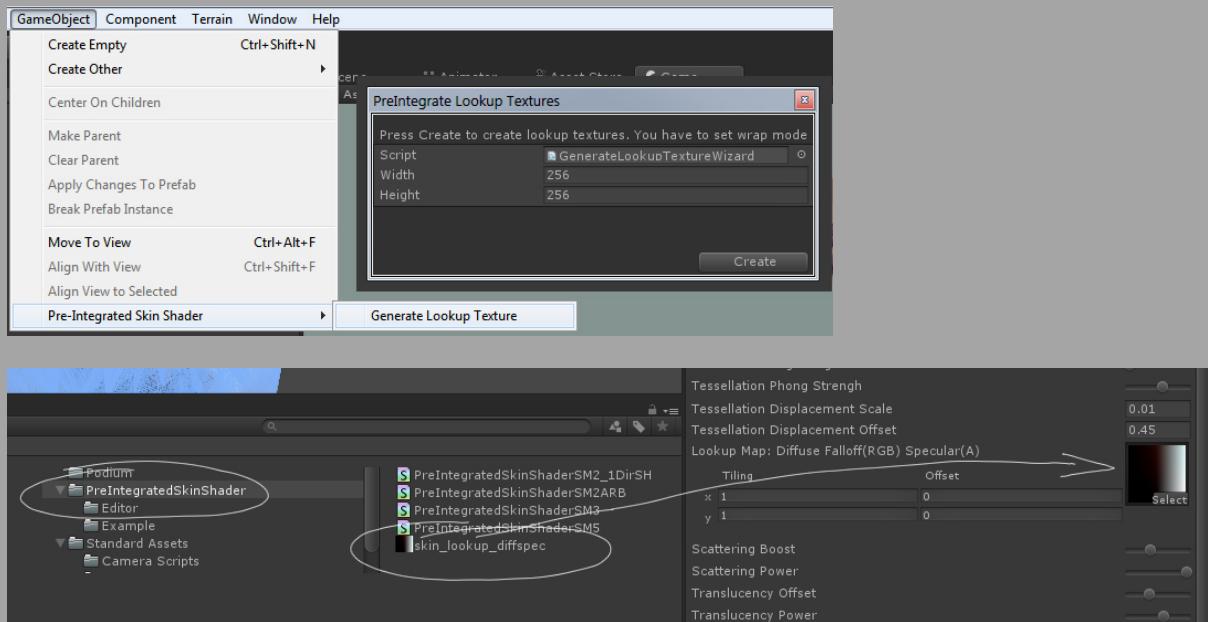
Scale 0.003, Offset 0.42

9. Diffuse Falloff Lookup map

In the “Lookup Map: Diffuse Falloff(RGB) Specular(A)” material property you should simply set the bundled skin_lookup_diffspec texture. It’s necessary for correct functioning of the shader.

The texture needs to be clamped in import settings and the texture compression turned off.

If needed the texture can also be generated with the script from Game Object → Pre-Integrated Skin Shader → Generate Lookup Texture menu. It writes to the PreIntegratedSkinShader folder in your project. It sets up all necessary settings.



10. Translucency

Scattering Boost	Scattering Power	Translucency Offset	Translucency Power	Translucency Radius
	0%	0%	0%	0%
	0%	100%	0%	100%
	0%	100%	25%	100%
	80%	100%	30%	100%
	20%	100%	25%	80%

Brief explanation:

Scattering Power - adjusts scattering strength

Scattering Boost - boosts scattering uniformly

Translucency Power - back scatter strength

Translucency Radius - radius of back scatter

Translucency Offset - shifts back scatter color from white to dark red

Credits

This shader in major part is an implementation of Eric Penner's Pre-Integrated skin shader as described in his Siggraph presentation. The specularity algorithm and diffuse profiles are taken from Nvidia's GPU Gems 3 (D'Eon et al.). The shader incorporates code to calculate transmittance by Jorge Jimenez et al. The lookup map generation script is a slightly modified version of the one made by Jon Moore and published in Gamasutra.

Thanks to those mentioned above, though they are probably rolling their eyes by now because of how I use their work. Big thanks to my brothers Karol and Jerzy for supporting me and providing me support and demo content. Thanks also to folks at Infinite Realities for the freely available head scan that served me as testing material.

Copyright 2013 Maciej Kacper Jagiełło, Karol Wiktor Jagiełło and Jerzy Roman Jagiełło
All Rights reserved.