KoddeShader 2.0
A CGFX real-time shader for Maya
Head model and textures by Lee Perry-Smith

Table of Contents

Description

Description

The "KoddeShader v2.0" is a real-time CgFX shader for Autodesk Maya. It was authored using Mental Mill 1.0 and Notepad++.

I created this shader to learn more about real time shaders and to provide the game art oriented Maya community with a "game like" viewport shader. This shader has very little focus towards being optimized.

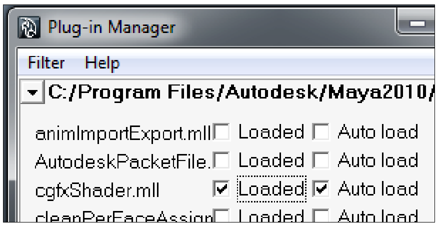
This guide assumes that the user has some prior basic knowledge of using Autodesk Maya and 3D graphics in general. General terms such as a "Normal Texture" are not explained in detail while more exotic ones such as "Blended Normals" are.

The author on a summer day.

Getting Started

Loading the CgFX plug-in

To use a CgFX shader in Maya you need to load the "cgfxShader.mll" plug-in which comes with Maya. It is not loaded by default. To load plug-ins go to the Window → Settings/Preferences → Plug-in Manager.



Plug-in Manager

Filter Help

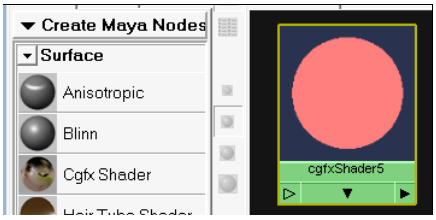
C:/Program Files/Autodesk/Maya2010/

animImportExport.mll □ Loaded □ Auto load
AutodeskPacketFile. □ Loaded □ Auto load
cgfxShader.mll ☑ Loaded ☑ Auto load
cleanPerFaceAssign □ Loaded □ Auto load

The "Plug-in Manager".

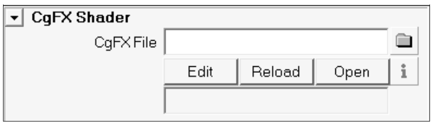
Creating a CgFX Shader surface node

Given that the "cgfxShader.mll" shader plug-in is loaded you should have a "Cgfx Shader" surface node in your Hypershader Editor under your "Create Maya Nodes".



A "Cgfx Shader" node.

Create a new "Cgfx Shader" surface node and open it in your attribute editor (double click the shader). To the right of the empty "CgFX File" field click the folder icon to browse for the external .cgfx file. As you open a CgFX file in Maya it might take a few seconds to fully load depending on the complexity of the shader. Apply the shader to your object as you would with any other shader in Maya. Remember to enable "Hardware Texturing" ("6" key) to see the results in your viewport.



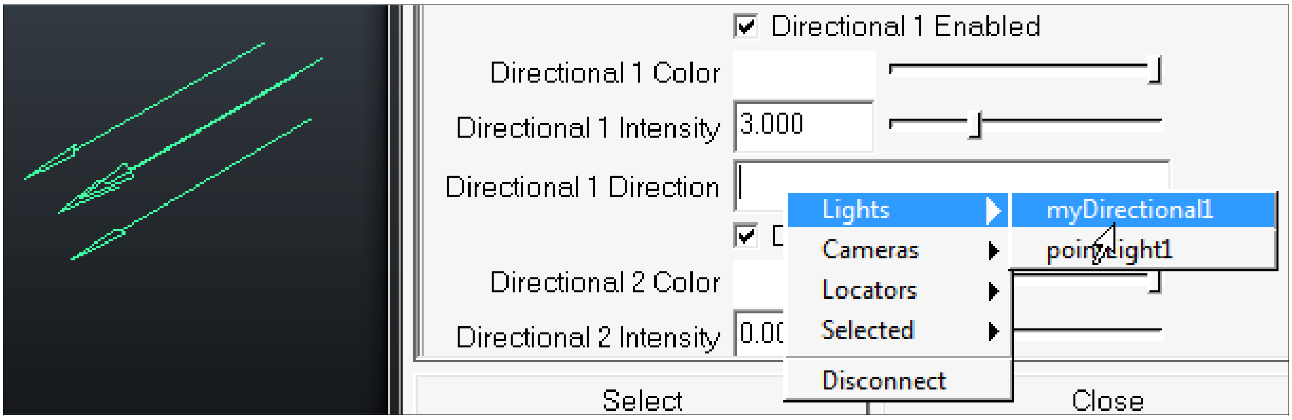
The "CgFX File" field.

Linking the lights

With a CgFX shader you have lights contained within the shader itself. This means you cannot just create a light in Maya as usual and expect it to work. The type and amount of lights possible to use with your CgFX shader is predefined by the author of the shader. The "KoddeShader v2.0" lets you use up to 3 directional lights and 3 point lights. Adding many light slots to a CgFX shader can increase the loading time and performance of the shader as well as cluttering the long list of parameters in the attribute editor.

Hopefully the lighting options in the shader will be enough for most users needs. Remember if you are lighting a big object or several objects with the same shader you can always create more shader nodes to link even more lights. The limit being a maximum or 3 directional lights and 3 point lights lighting the sampled surface point a the same time. This is not counting the Ambient Light feature.

All parameters for the light contained within the shader. So any time you want to change intensity, vary the color, tweak the falloff, etc. you have to go through the shader. The exception being the lights direction/position. What you do when it comes to a lights direction or position is link this parameter to an object in your scene. Then just rotate or position the linked object to define the lights direction or position.



☑ Directional 1 Enabled

Directional 1 Color

Directional 1 Intensity 3.000

Directional 1 Direction

☑ D

Directional 2 Color

Directional 2 Intensity 0.00

Linking a Maya directional light to the "Direcational 1 Direction" parameter.

A tip is to actually use the regular Maya directional light or point light in Maya as the objects to define the shader lights direction/position. The reason for this being that when right-clicking the empty light slots in the shader you get a context menu which lists the lights in your scene making it an quick and easy way. Also the representations for the directional/point light in the viewport makes good sense.

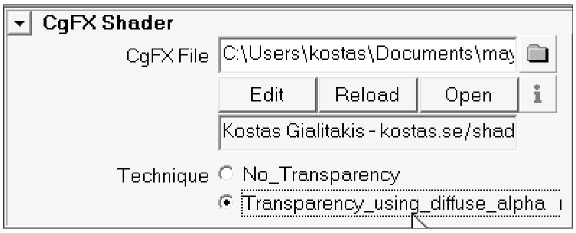
Here's a quick step by step on how to link a directional light in the shader.

1. Create a regular directional light in Maya. Name it to something appropriate as you will have to select it in a list.
2. Open the shader you want to link it to in the attribute editor. Scroll to the lower part where the light parameters are and find the "Directional 1 Direction" slot.
3. Right-click the empty slot and select Lights → <YourLightsName>.
4. Voila! That should be it. Now as you rotate the newly created directional light in the scene you should see the light on your shaded object update.

Warning! Your linked object's scale affects the shaders light. Therefor it's a good thing not to scale your linked object. For any polygonal object the Freeze Transformations command will solve this issue, however not for Maya lights. If you are using the regular Maya directional or spot light to link your lights and you want them to appear bigger try this: Open your light in the Attribute editor. Make sure you are on the light's shape tab. Go to the "Object Display" section and increase the "Locator Scale" attribute. This will make the lights representation bigger without changing the scale values. When it comes to point lights I don't know if there is any way at all to make them have a bigger representation, not even the regular scale seems to work.

Shader Techinques

The "KoddeShader v2.0" has two techniques to choose from. They are located at the very top of the shader in the attribute editor in the "CgFX Shader" section.



CgFX Shader

CgFX File C:\Users\kostas\Documents\may

Kostas Gialitakis - kostas.se/shad

Technique ○ No_Transparency
● Transparency_using_diffuse_alpha

The "Technique" choices.

"No_Transparency"

This is the regular shading technique. As the name suggests this technique has no form of transparency.

Specularity and reflections on a semi transparent orange surface.

"Transparency_using_diffuse_alpha"

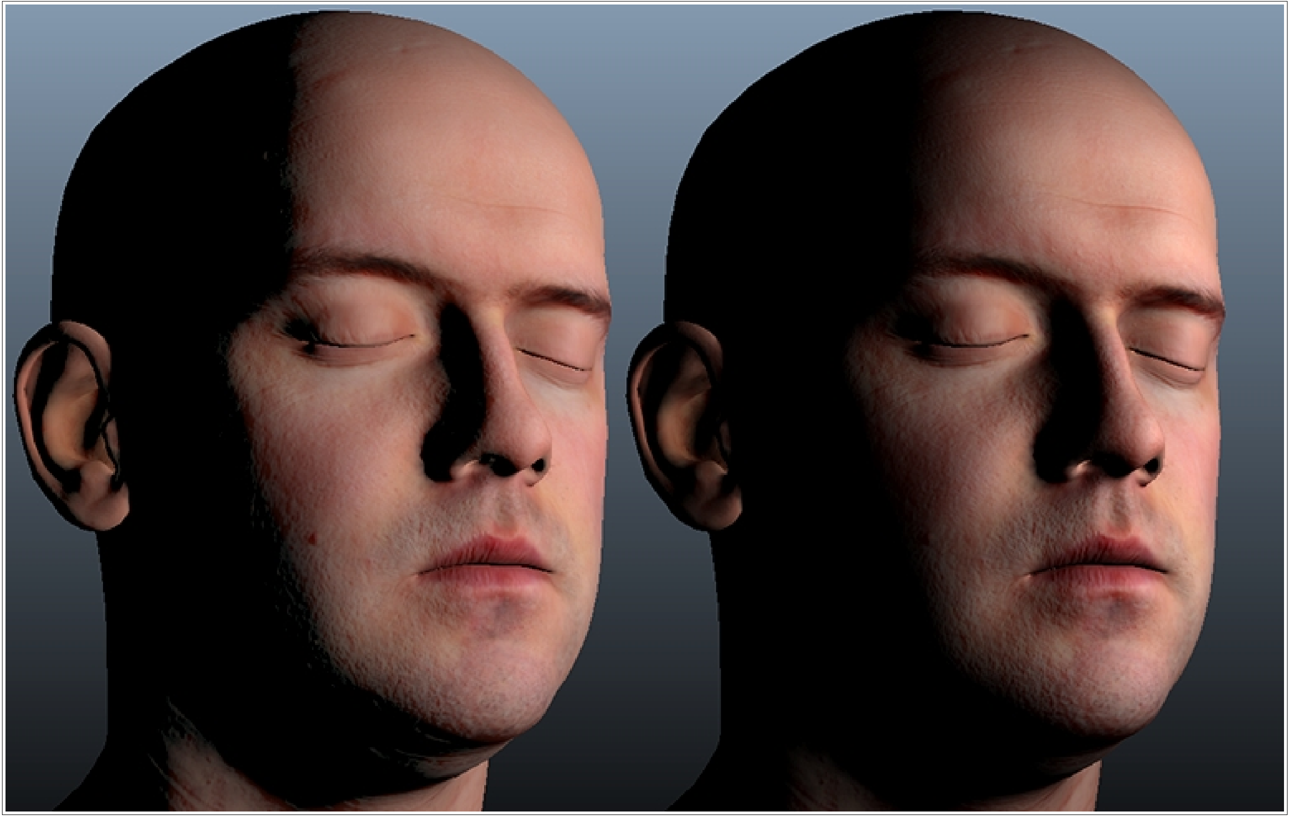
This technique uses two passes per frame to achieve it's results. The first pass will be the regular shading solution with transparency according to the diffuse texture's alpha channel. White is visible, black is transparent. The second pass will add specularity and reflections on transparent surfaces if the option "Specular and Reflections on Transp Surf" is checked. This is good for imitating transparent surfaces such as glass, plastic, etc. Keep in mind that Maya's viewport can be quite picky with transparency in general.

Parameter Descriptions

Gamma Correction

This parameter will switch between using gamma correction or not. Gamma correction gives more believable lighting results. More towards how light works in reality.

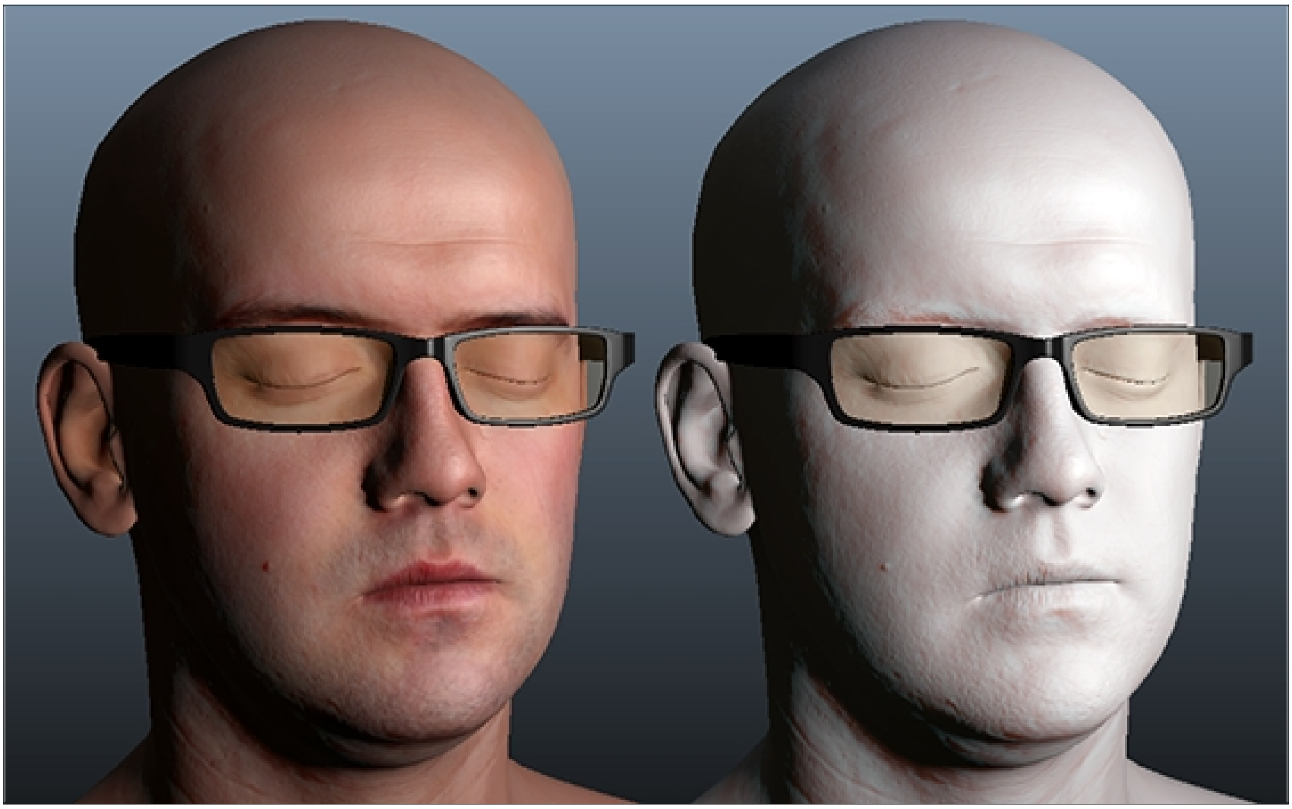
Technically speaking what this means is the diffuse, specular (including gloss), environment cube map, reflectivity and emissive map gets converted to linear space (pow 2.2) as they are loaded. Then all the shader magic takes place, and before displaying the shaders results on screen the shader converts back to gamma space (pow 1/2.2).



Left: *Gamma corrected. Notice the sharp light terminator line.* ***Right:*** *Non gamma corrected.*

Diffuse Texture

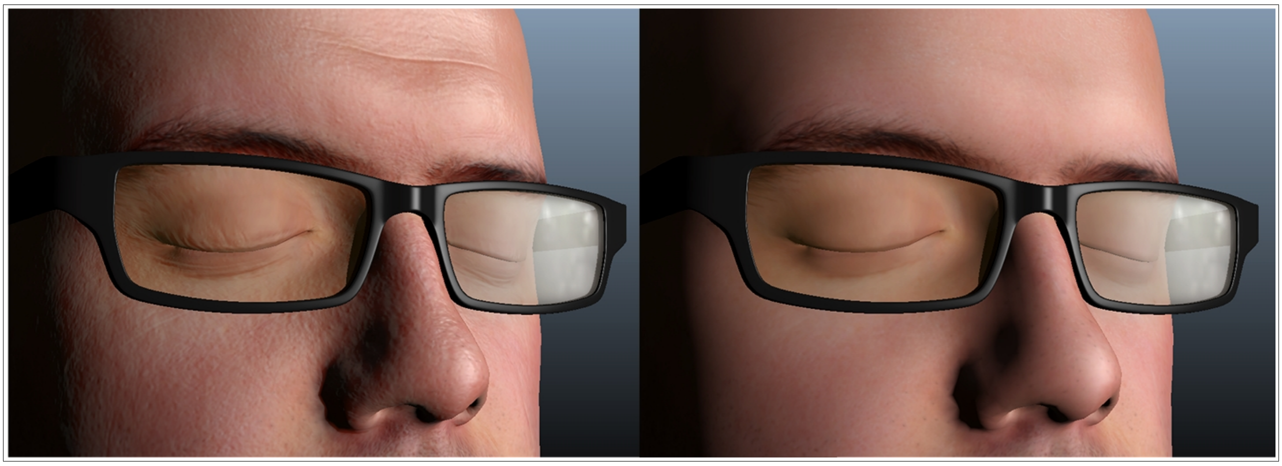
This is where you load your diffuse texture. If you want to use transparency using the "Transparency_using_diffuse_alpha" technique then store your transparency data in the diffuse textures alpha channel. More on transparency in the Getting Started → Shader Techniques.



Left: *Using a diffuse texture.* ***Right:*** *Using no diffuse texture.*

Normal Texture

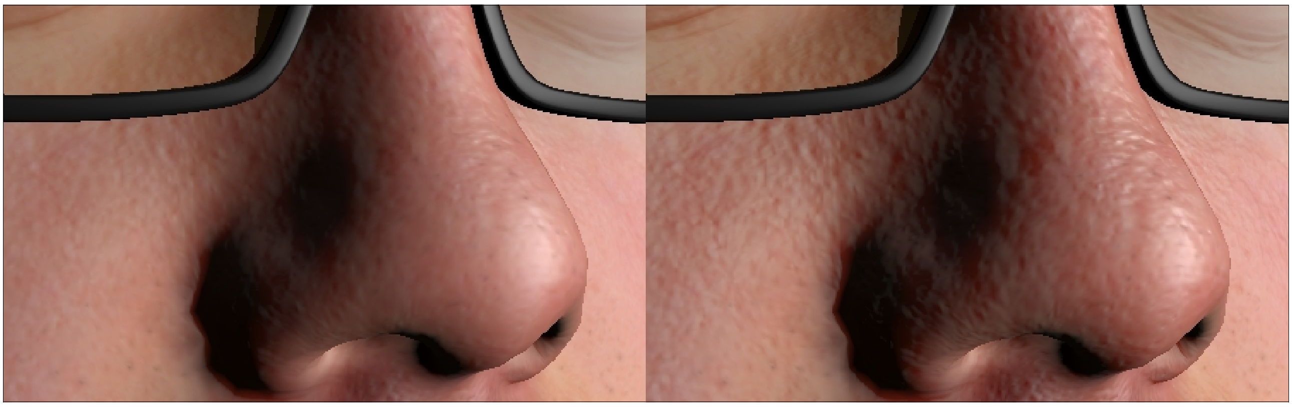
This is where you load your normal texture.



Left: *Using a normal map.* ***Right:*** *Using no normal map.*

Normal Multiplier

This parameter lets you vary the normal texture's effect. Using a high value will give more of the effect. A value of 1.0 represents the common strength at which a normal texture would be used with. If you are using the Blended Normals feature then try pushing this multiplier a bit further. I've found that you can generally push the effect of the normal texture a bit further when using Blended Normals than you would usually do.

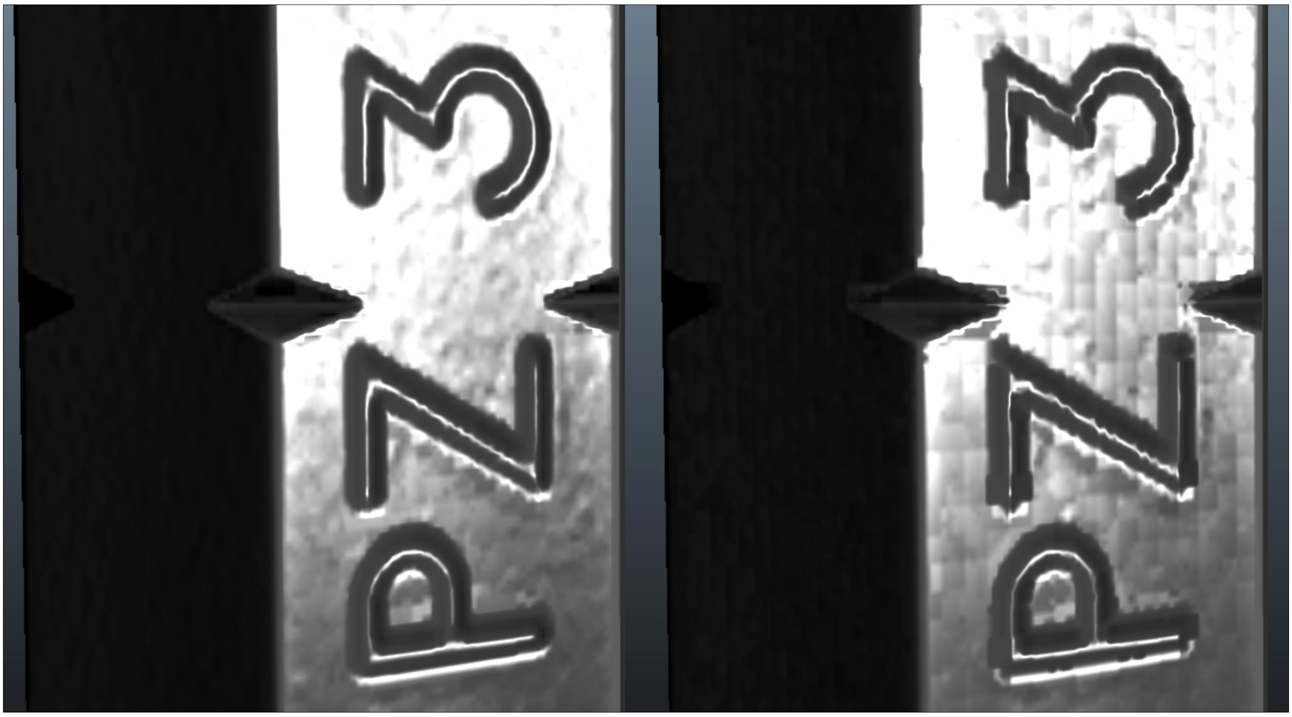


Left: *Using Normal Multiplier at 1.0.* ***Right:*** *Using Normal Multiplier at 2.0.*

Use DXT5nm style Normal Texture

This option was one I considered not including at all, but in the end I kept it. If you are not using compressed texture or have no interest in using compressed textures then there's no point in reading on.

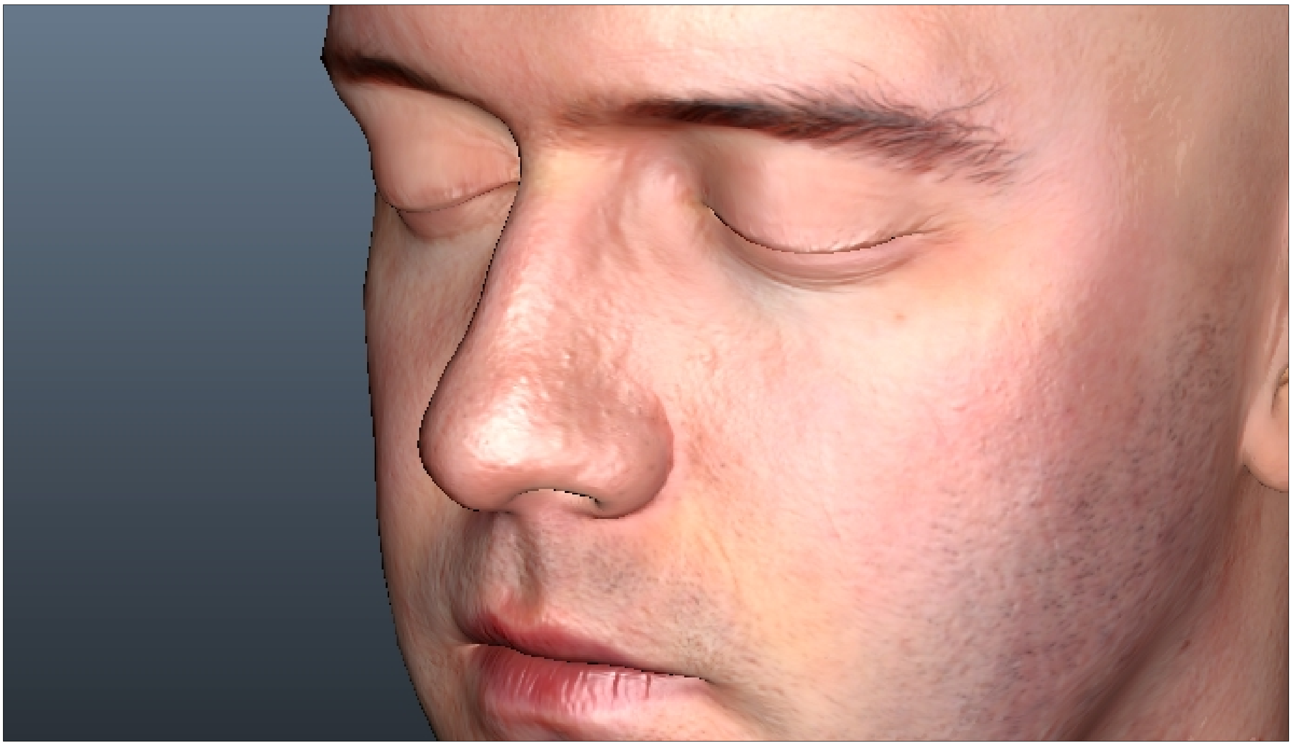
What this feature does is switch to using a normal texture which stores it's XY data in the R and A channels of a texture instead of the usual XYZ data in the RGB channels. This is purely an optimization technique used in game development to push the quality level of a compressed normal texture while still keeping memory size down. This does cost a bit more performance as this option will recreate the Z component in real-time. I decide to keep this option as it might be nice for people who want to see the difference of using a DXT1 DDS normal texture compared to a DXT5nm DDS normal texture.



An image from a previous normal texture compression study of mine. Left: Using DXT5nm compression. Right: Using DXT1 compression.

Specular Texture

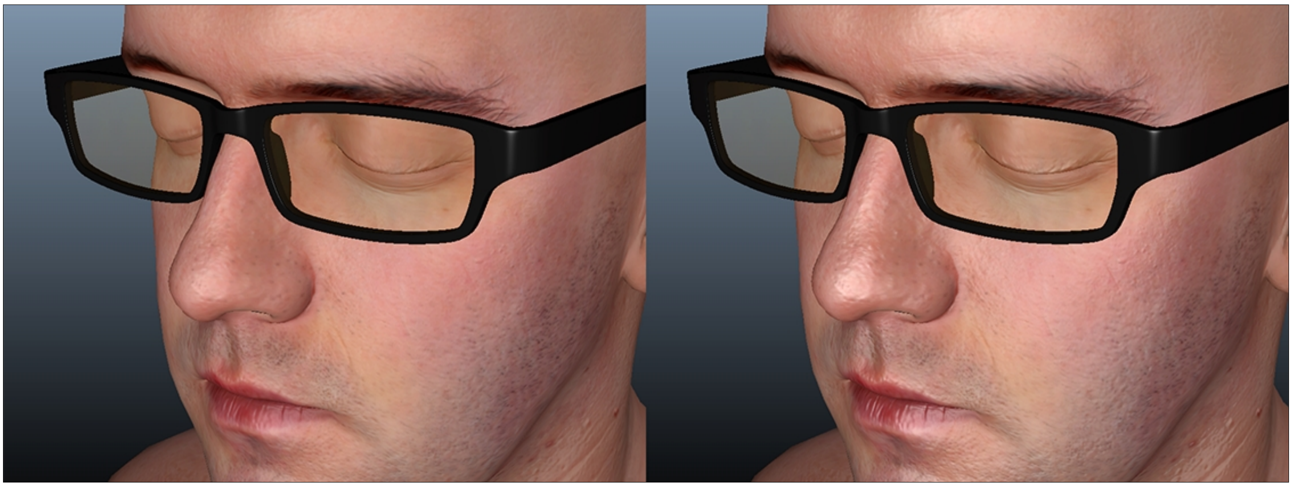
This is where you load your specular texture.



Notice how the specular map in this shot makes the specularity on different areas vary. For example the brows hardly have any specularity while the tip of the nose has lots of specularity.

Specular Multiplier

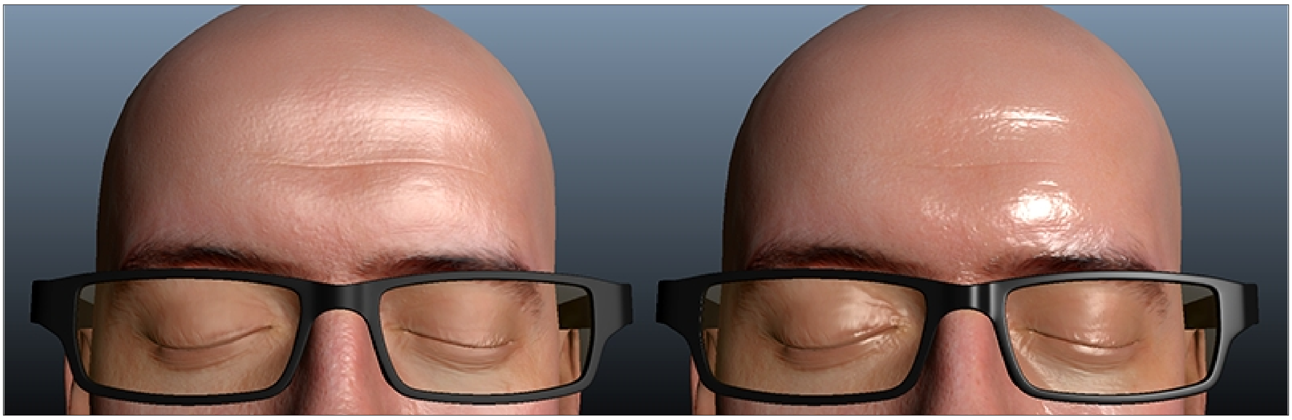
This parameter lets you modulate the specularity's strength. The higher the value the stronger the specularity.



Left: *Specular Multiplier of 0.8.* ***Right:*** *Specular Multiplier of 1.8.*

Gloss Factor

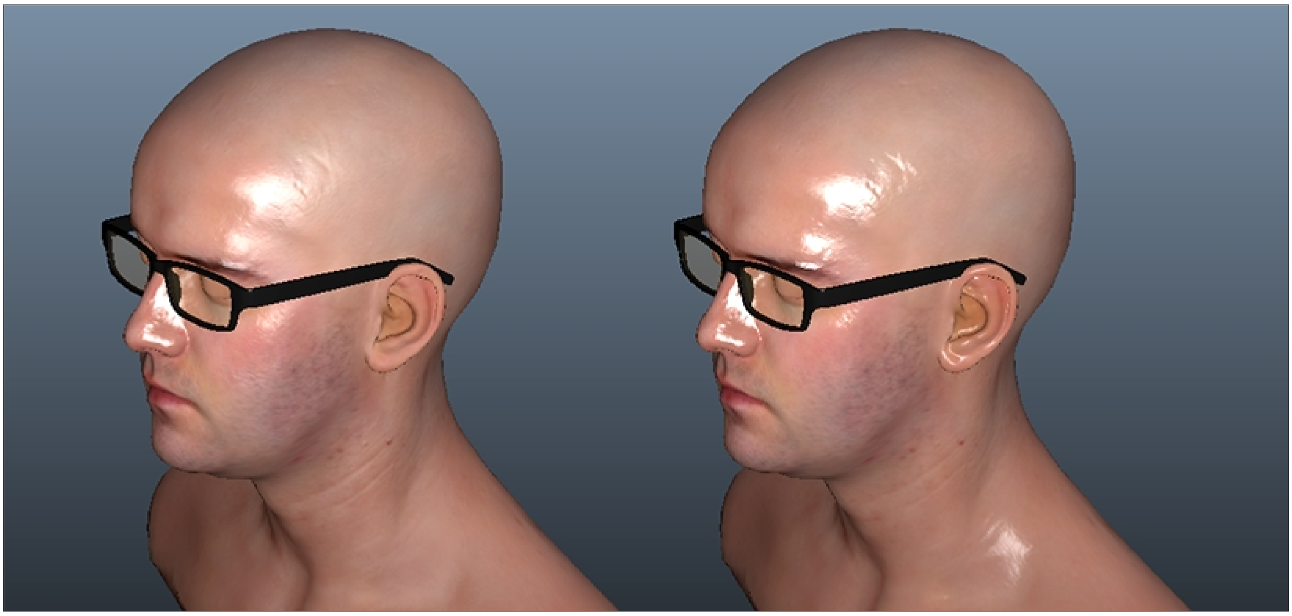
This parameter controls how glossy/rough your specular highlights will be. A low value will result in a broad highlight while a high value will result in a narrow highlight. This value also determines the mip level to be used for reflections if you are using a proper cube map texture. With a good Cube Map and the "Use Specular Texture Alpha as Gloss" feature enabled you can get reflections to vary from being sharp to blurred across a surface. More on cube map mip levels in the Parameter Descriptions → Reflection Mip Level Min/Max.



Left: *A gloss factor value of 6.* ***Right:*** *A gloss factor value of 40.*

Use Specular Texture Alpha as Gloss

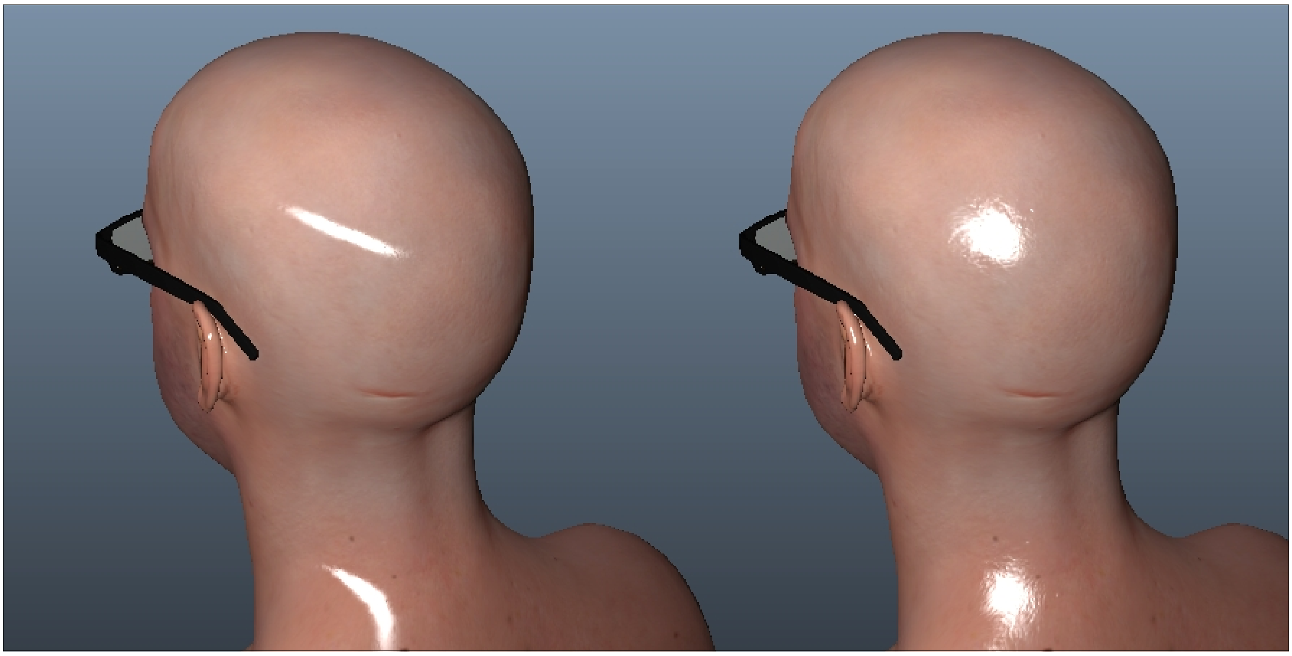
This parameter enables using your specular textures alpha channel as the source for the gloss factor described above. I highly recommend you use this feature as it is a very important aspect of mimicking a materials real life properties.



Left: *Using specular textures alpha channel as source for gloss. Note the varying glossiness across the model.* ***Right:*** *Not using specular textures alpha as source for gloss.*

Use Ward Specularity

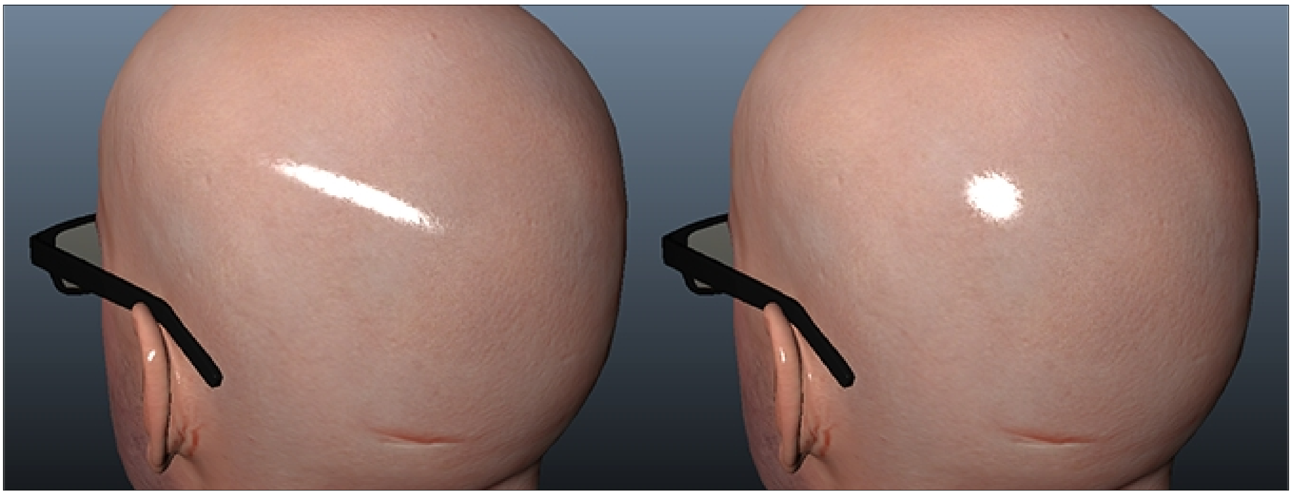
This parameter enables using Ward specularity as opposed to using the regular Phong based specularity. The Ward based specularity lets the user change the shape of the specular highlight. This is commonly referred to as a anisotropic type of specularity. This is useful for mimicking specularity of hair, velvet, etc. Since I found the effect of the normal texture varied a lot from using Ward/Phong specularity the Ward specularity method has an internal multiplier of it's normal texture effect by x10.



Left: *Using Ward specularity.* ***Right:*** *Using the default Phong specularity.*

Ward Spec Gloss U/V

These two parameters control the shape of the Ward specular highlight. Low values produce a broad highlight and high values a narrow highlight. The further apart the U and V values are from each other the more "stretched" the highlight will be. The closer the U and V values are to each other the more uniform the highlight will be.



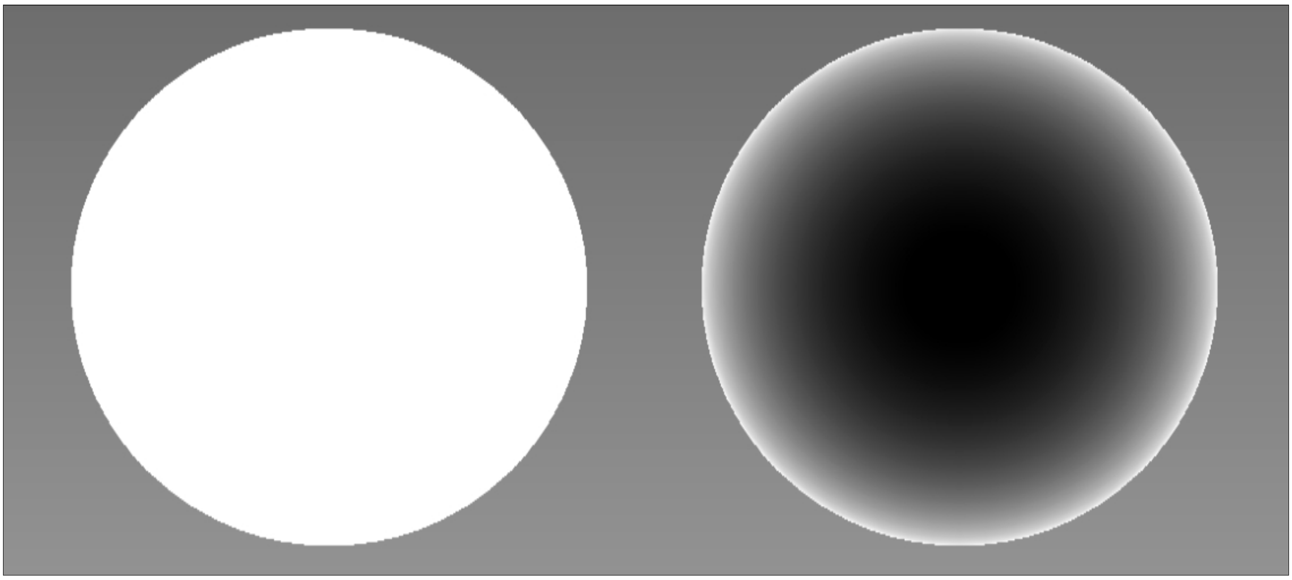
Left: *An U/V value of 5/30.* ***Right:*** *An U/V value of 15/15.*

Fresnel Foreword

The Fresnel features in this shader are based around a surface's normal vector towards the camera. The illustrations for the Fresnel parameters below are perfectly round polygonal spheres. The Fresnel effect modulates the specularity and reflectivity of this shader. A white color represents where specularity/reflections will be applied and a black color represents where they will not be applied.

Fresnel Falloff Amount

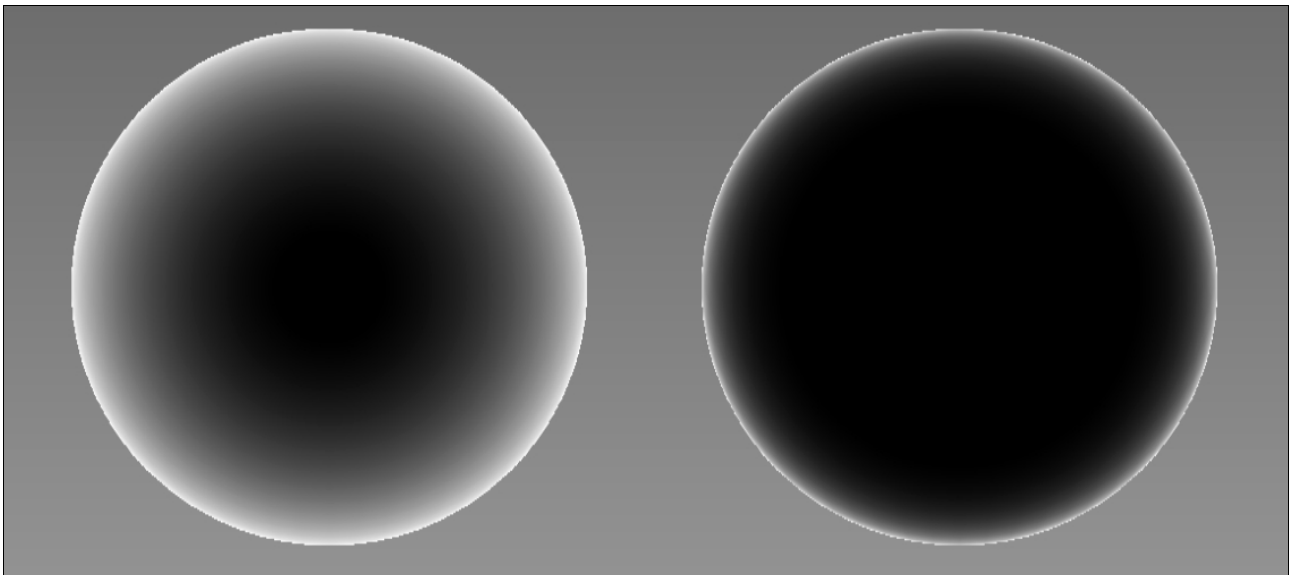
This parameter controls how much of the Fresnel falloff effect will be applied. A high value means that the white area will start creeping back towards surfaces which are at more glancing angles (facing away from camera). This is the base parameter for the other two Fresnel parameters. If this is left at 0.0 the other two Fresnel parameters will have no effect.



Left: *Fresnel falloff amount of 0.0.* ***Right:*** *Fresnel falloff amount of 1.0.*

Fresnel Exponent

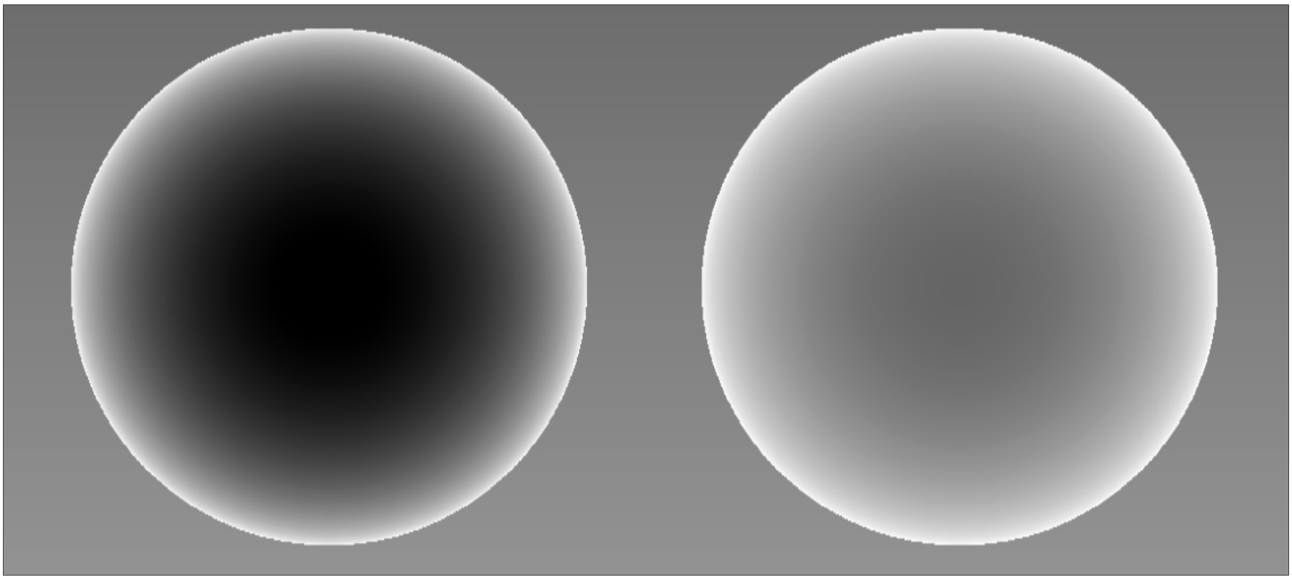
This parameter controls the exponential type of curvature to the Fresnel falloff. A value of 1.0 will leave the falloff effect being linear. A value higher than 1.0 will give an exponential "down" type effect, the white will mostly be apparent at very glancing angles. A value lower than 1.0 will give an exponential "up" type effect, the white will be apparent on most angles except the ones which are facing directly towards the camera (or close to directly). Technically speaking this is a Power Of mathematical function.



Left: *Fresnel Exponent of 1.0.* ***Right:*** *Fresnel Exponent of 3.0.*

Fresnel Bias

This parameter will shift the darkest color from black to white the higher the supplied value is. This parameter is intended to be used within the 0.0-1.0 range. This is useful if you don't want the specularity or reflections to completely be removed from surfaces facing the camera but still want to modulate them by a Fresnel effect.

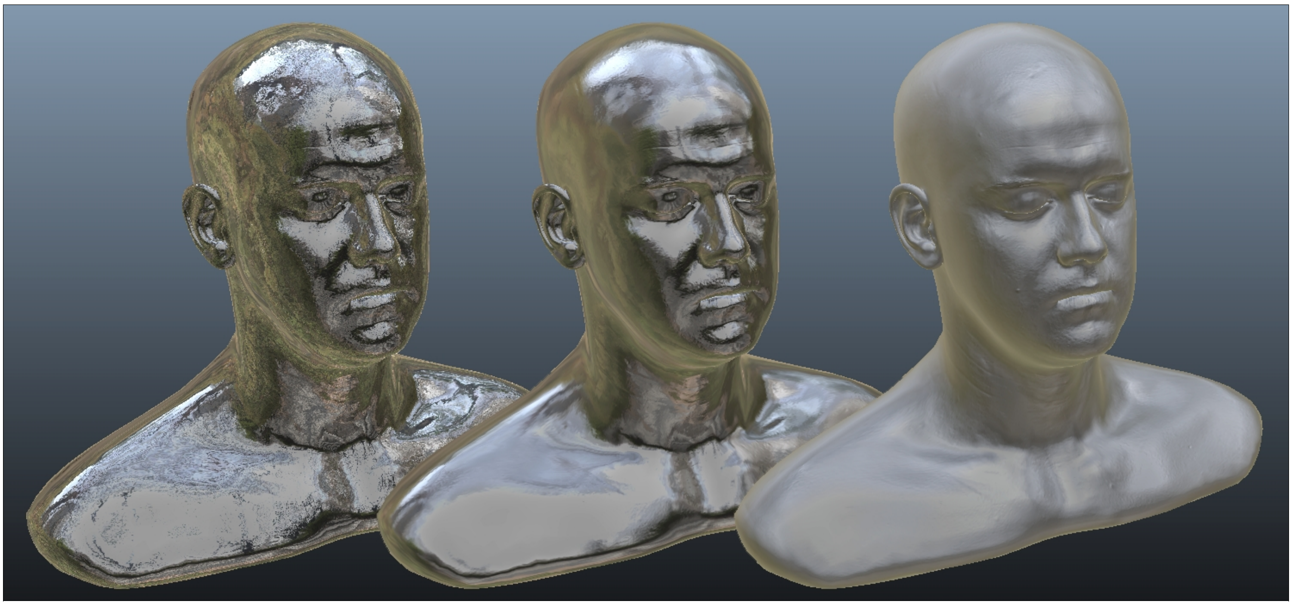


Left: *Fresnel bias of 0.0.* ***Right:*** *Fresnel bias of 0.4.*

Environment Cube Map

This is where you load your cube map which will be used as the reflections and ambient light source. This texture needs to be in the DDS format and saved as a Nvidia Horizontal Strip type cube map. When processing cube maps I can personally recommend the ATI CubeMapGen software. It has some great tools including proper blurring of cube maps.

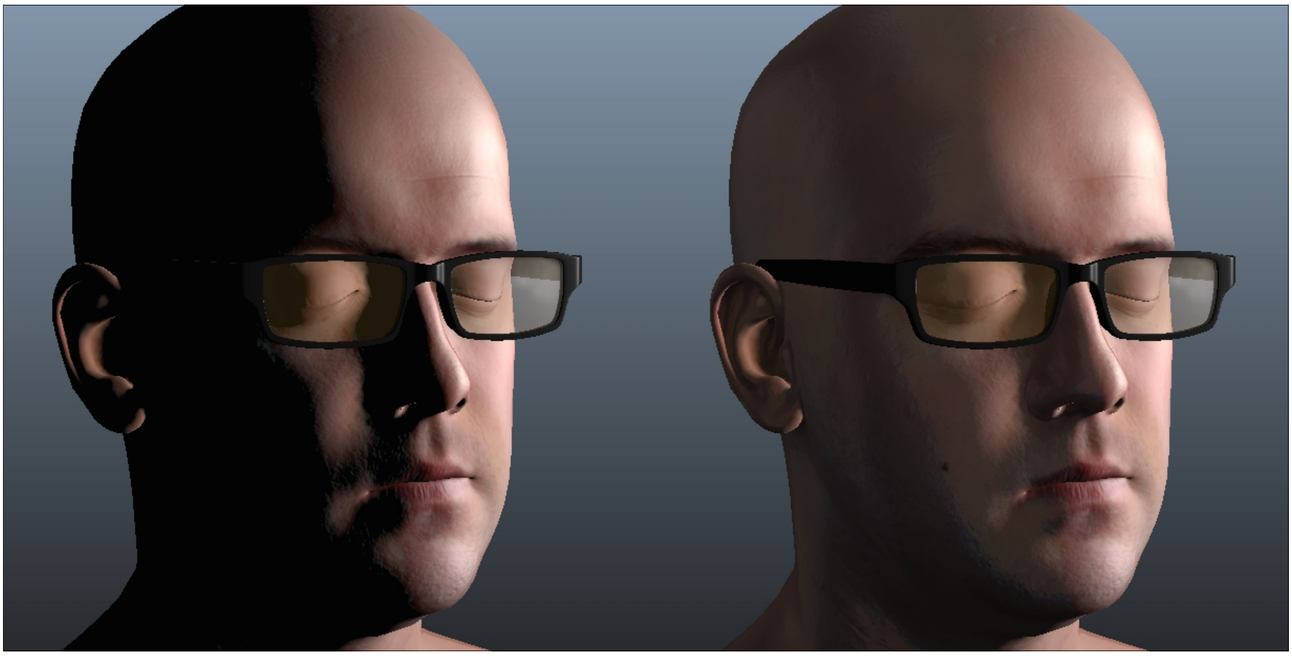
To get the most of the "Ambient Mip Level", "Reflection Mip Level Min" and "Reflection Mip Level Max" features described further on you want to consider blurring/sharpening your cube map mip levels according to the effect you want to achieve. For example, if you pre-blur your mip levels you can use the mip level parameters to vary your ambient light or reflections from being sharp to being blurry.



Cube map reflections using pre-blurred mip levels. The three copies of the model use different mip levels to show the achieved effect.

Ambient Light Multiplier

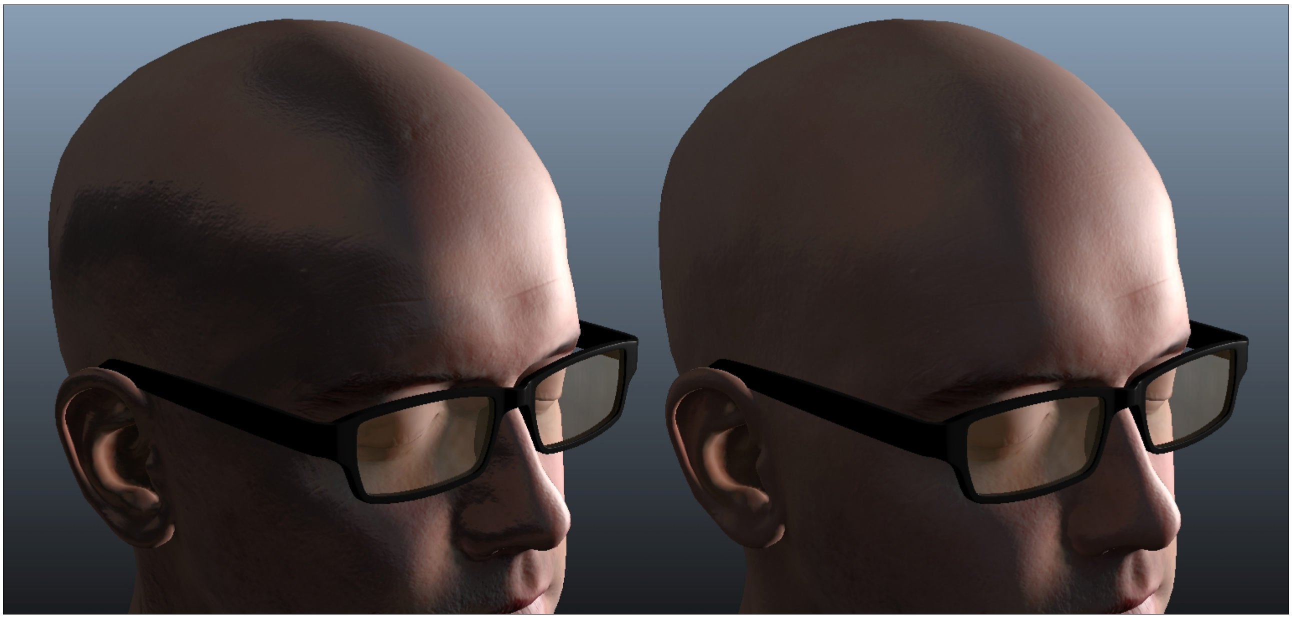
This parameter controls how much ambient light you want to add to your shader. The higher the ambient light multiplier value the more ambient light. This feature requires that you have loaded a cube map as this will be used as your ambient light source.



Left: *Using ambient light multiplier of 0.0.* ***Right:*** *Using ambient light multiplier of 0.06.*

Ambient Mip Level

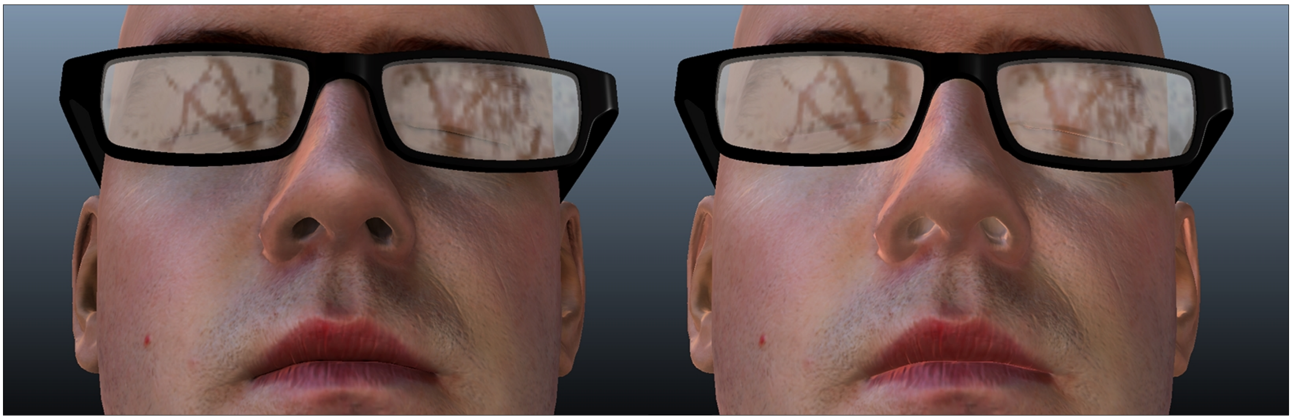
This parameter directly controls which cube map mip level will be used as the ambient light source.



Left: *Using a higher (low value) mip level resulting in more sharp ambient light, you can make out shapes.* ***Right:*** *Using a lower (high value) mip level resulting in a more blurred ambient light.*

Ambient Occlusion Texture

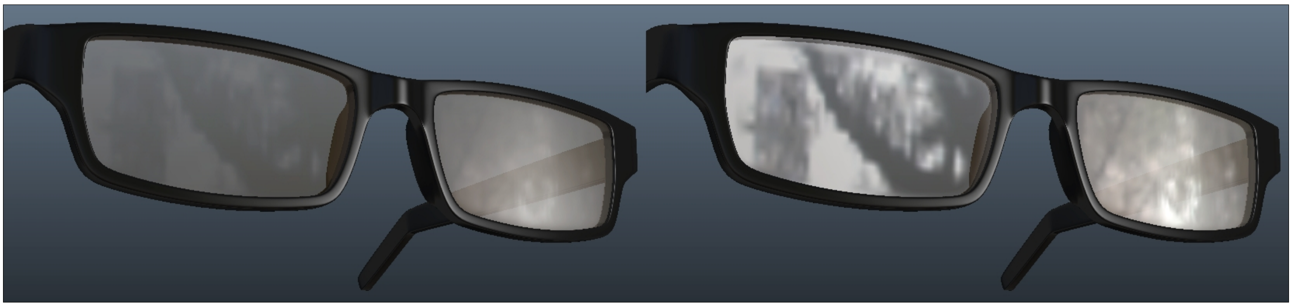
This is where you load your ambient occlusion texture. A texture which basically masks where ambient light will be applied and where it won't be.



Left: *Using an ambient occlusion texture. Note the dark areas where the lips meet, in the nostrils, around the eyes, etc.* ***Right:*** *Using no ambient occlusion texture. For illustrative reasons this shot only uses x1.0 ambient light as its only light source.*

Reflectivity Multiplier

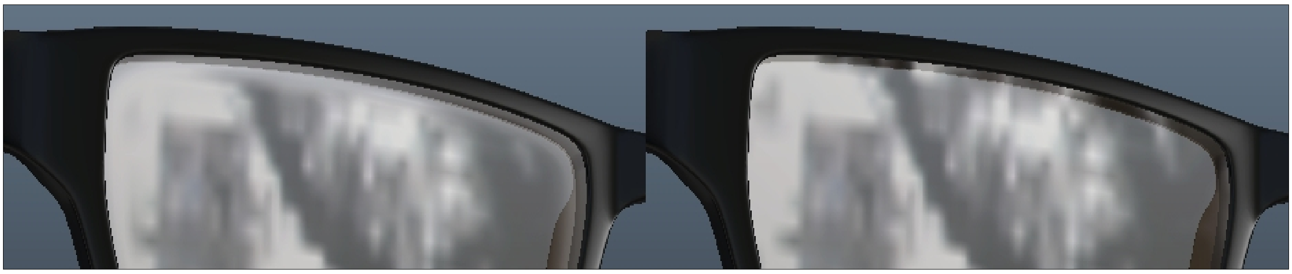
This parameter controls strength of the reflections being applied to your shader. The higher the value the stronger the reflections. This feature requires that you have loaded a cube map as this will be used as your reflections source.



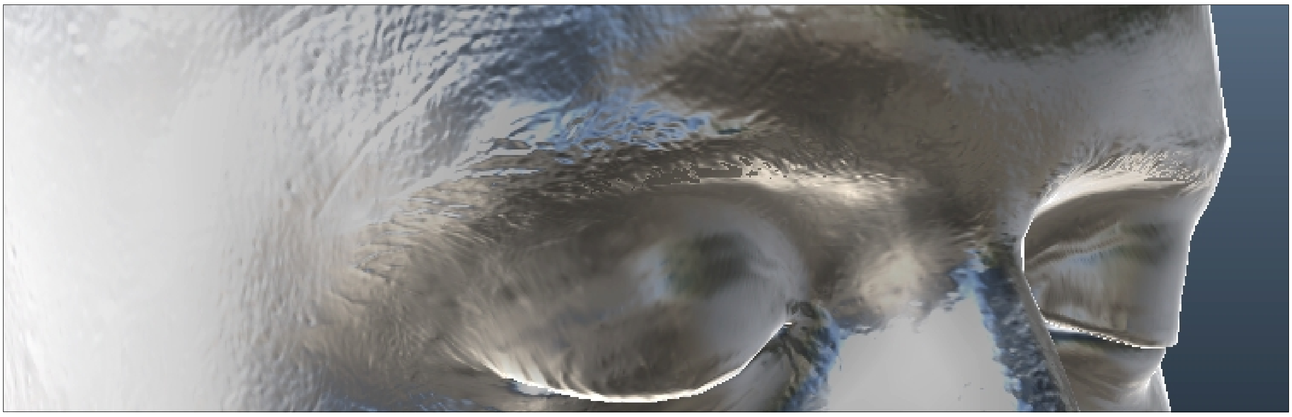
Left: *Using a reflectivity multiplier of 0.4.* ***Right:*** *Using a reflectivitiy multiplier of 1.4.*

Reflection Mip Level Min / Reflection Mip Level Max

These parameters are used to control which mip level(s) the reflections will use. If you are not using the "Use Specular Texture Alpha as Gloss" feature the "Reflection Mip Level Max" parameter will control which mip level will be uniformly used on your surface. If you are using the "Use Specular Texture Alpha as Gloss" feature then your gloss texture data will be used in the equation. The "Reflection Mip Level Min" will represent the mip level that black texture areas will be using and the "Reflection Mip Level Max" will represent the mip level that white texture areas will be using. All values in between will interpolate accordingly.



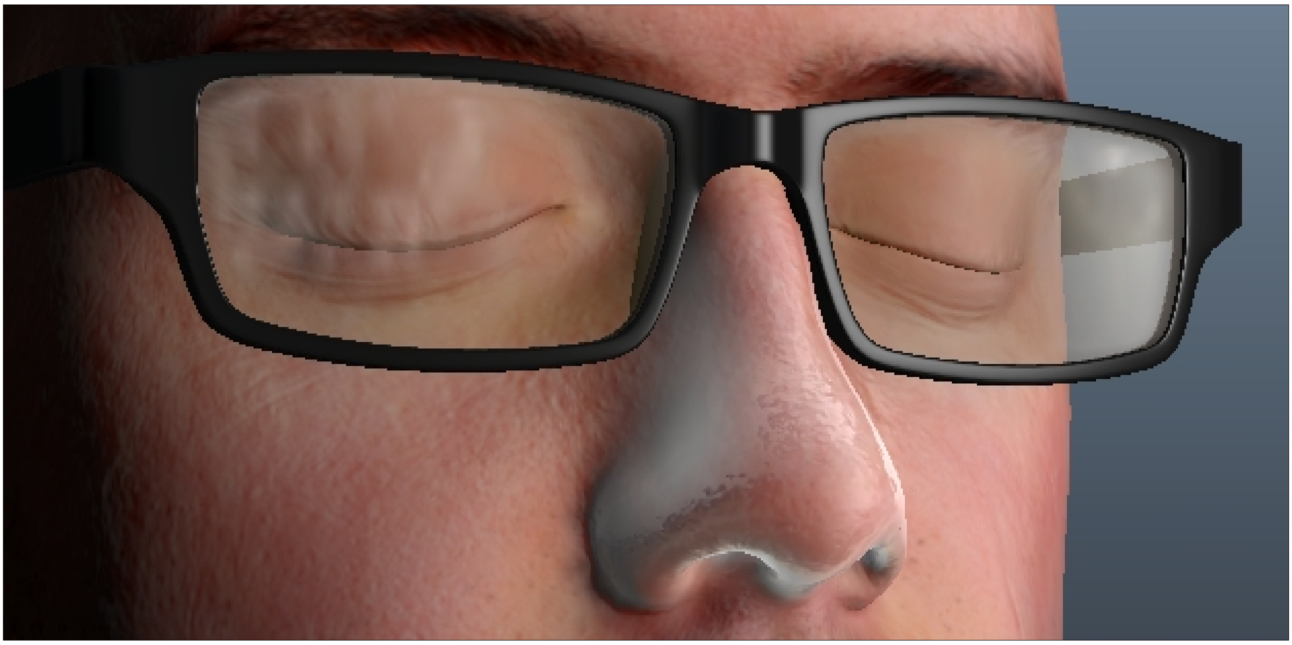
Left: *Using varying reflection mip levels to achieve a more "smudged" appearance on the glass near the frame.* ***Right:*** *Using a uniform mip level across all of the glass surface.*



An exaggerated example of varying mip levels for a blurred effect across a model.

Reflectivity Texture

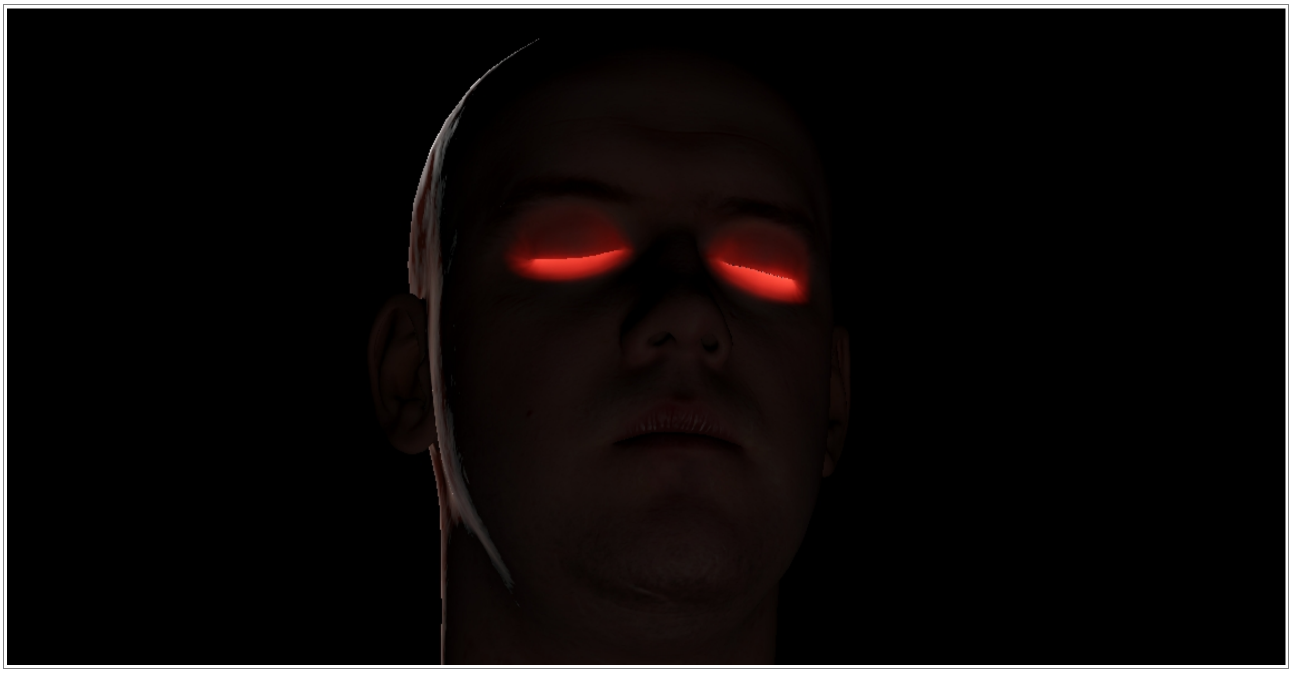
This is where you load your reflectivity texture which will essentially mask which areas will be reflective and which will not. A white color reflects fully while a black color won't reflect at all.



A highly reflective nose achieved by a reflectivity texture with white coloring on the nose texture area.

Emissive Texture

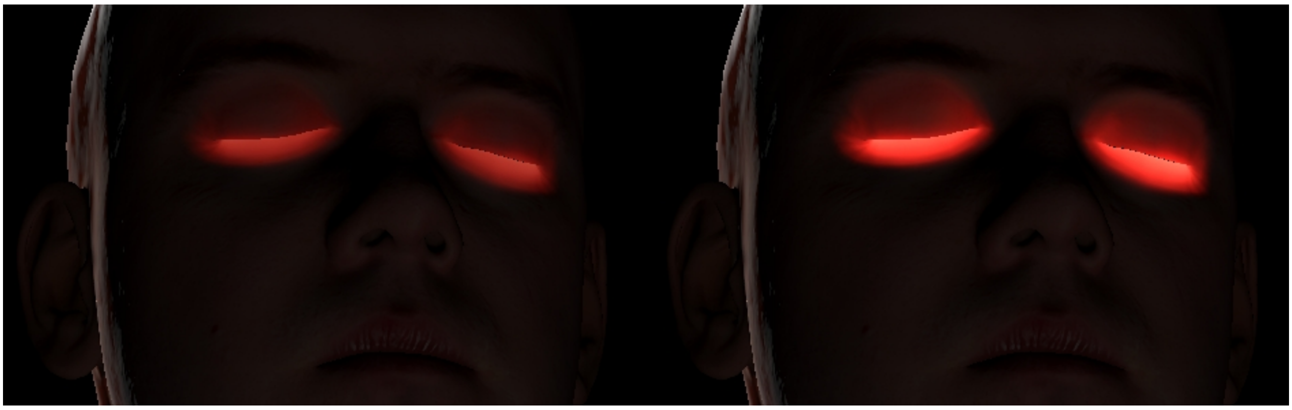
This is where you load your emissive texture. Emissive in the sense that it will seem "full bright". This does not emit any light or add any glow. This feature adds the colors you place in your texture to your current shading solution.



Emissive texture being used to simulate some form of red glow around the models eyes.

Emissive Multiplier

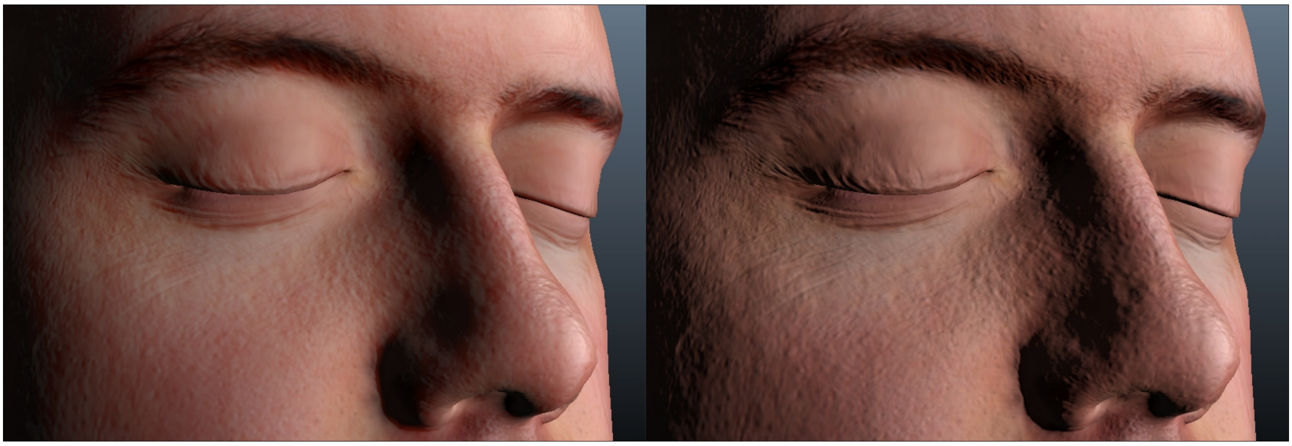
This parameter controls the strength of the emissive addition. A value of 1.0 will add the exact amount of colors in the texture. Raising or lowering the value beyond 1.0 will strengthen or weaken the effect.



Left: *An emissive multiplier value of 0.5.* ***Right:*** *An emissive multiplier value of 2.0.*

Skin: Use Blended Normals

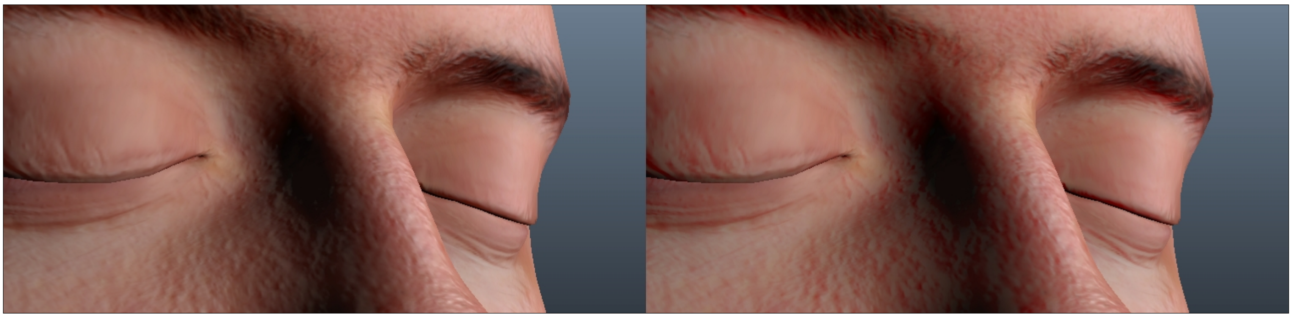
The parameter enables the use of "blended normals". A quite simple trick which produces a quite soothing red skin tone in fine detail areas which would normally be black. This feature works in conjunction with the details in your normal texture to modify the diffuse light base. For instance a small skin bump present in your normal texture would have a dark side on the model when being lit from the side, when using blended normals you can make it seem red rather than black on this dark side. This illustration says it all:



Left: *Using blended normals.* ***Right:*** *Not using blended normals.*

Skin: Blend Weights

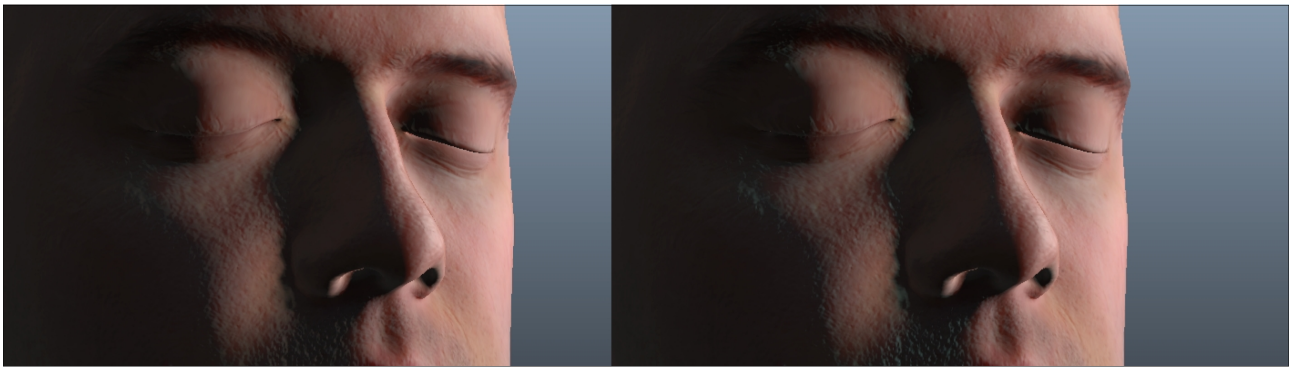
This parameter contains three values which control how the actual blending of normals work. Technically speaking setting a value in the low end will make that color channel lean more towards using the geometry normals. Setting a value in the high end will make that color channel lean more towards using the normal texture modified normals. These values work within the 0.0-1.0 range. To mimic a skin look you want to set the first value (Red) to a rather low value and the other two values (Green & Blue) to rather high values. The default values are set to 0.3, 0.8, 0.9 as these work quite well for skin.



Left: *Using blend weights of 0.3, 0.8, 0.9.* ***Right:*** *Using blend weights of 0.0, 1.0, 1.0.*

Skin: Blend Cyan Suppress

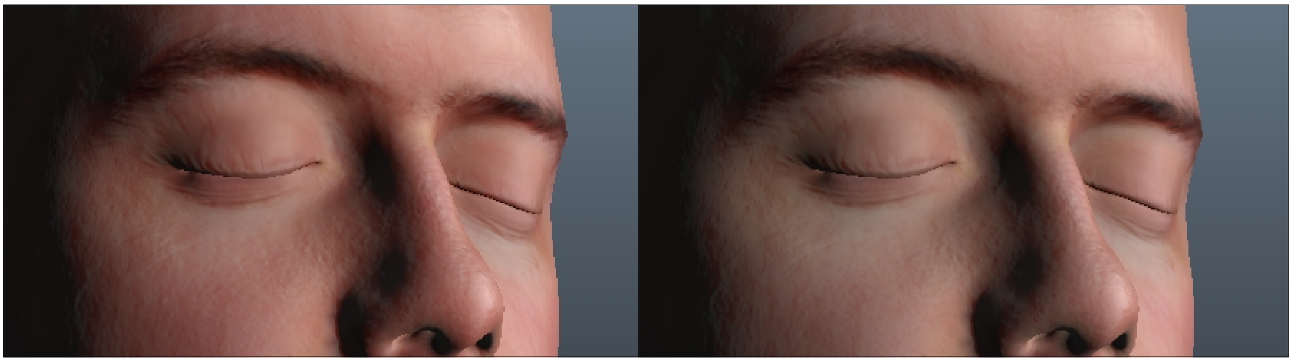
This parameter can be used to suppress the cyan color which is a bi-product of enabling blended normals and setting the blend weights to mimic skin. The higher this parameter value is set the more cyan will be suppressed.



Left: *Using a blend cyan suppress value of 0.95.* ***Right:*** *Using no blend cyan suppress at all.*

Skin: Tint Red

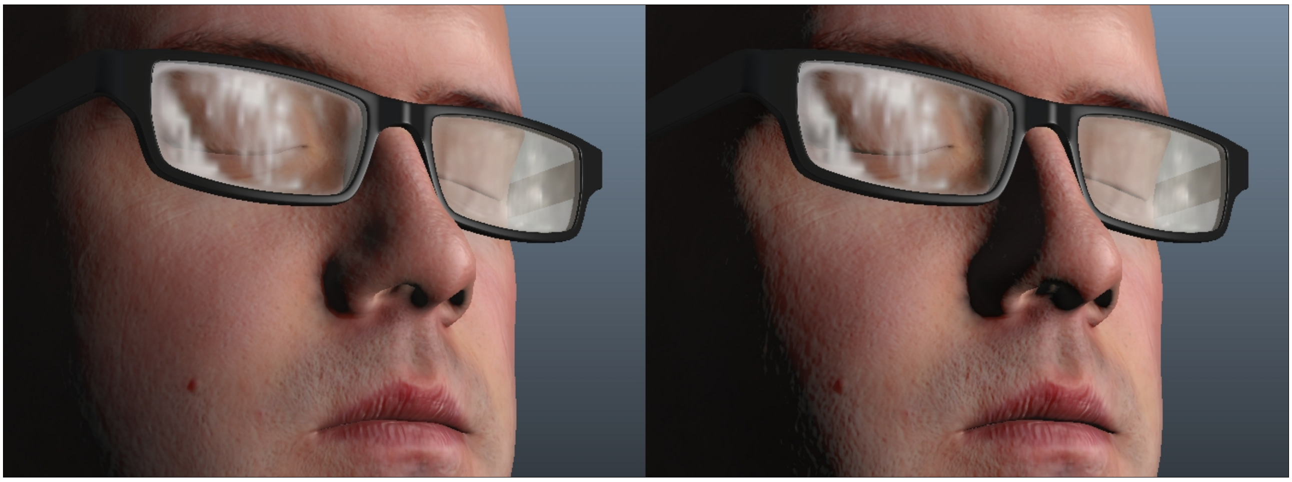
This parameter will add some red coloring to the darker areas of the diffuse lighting. Meaning that fully lit areas will be unchanged while darker areas will receive this red tint. The higher this parameter value the more red tint. This is supposed to mimic a bit of the scattering of light in the skin.



Left: *A tint red value of 1.0.* ***Right:*** *A tint red value of 0.0.*

Skin: Terminator Push

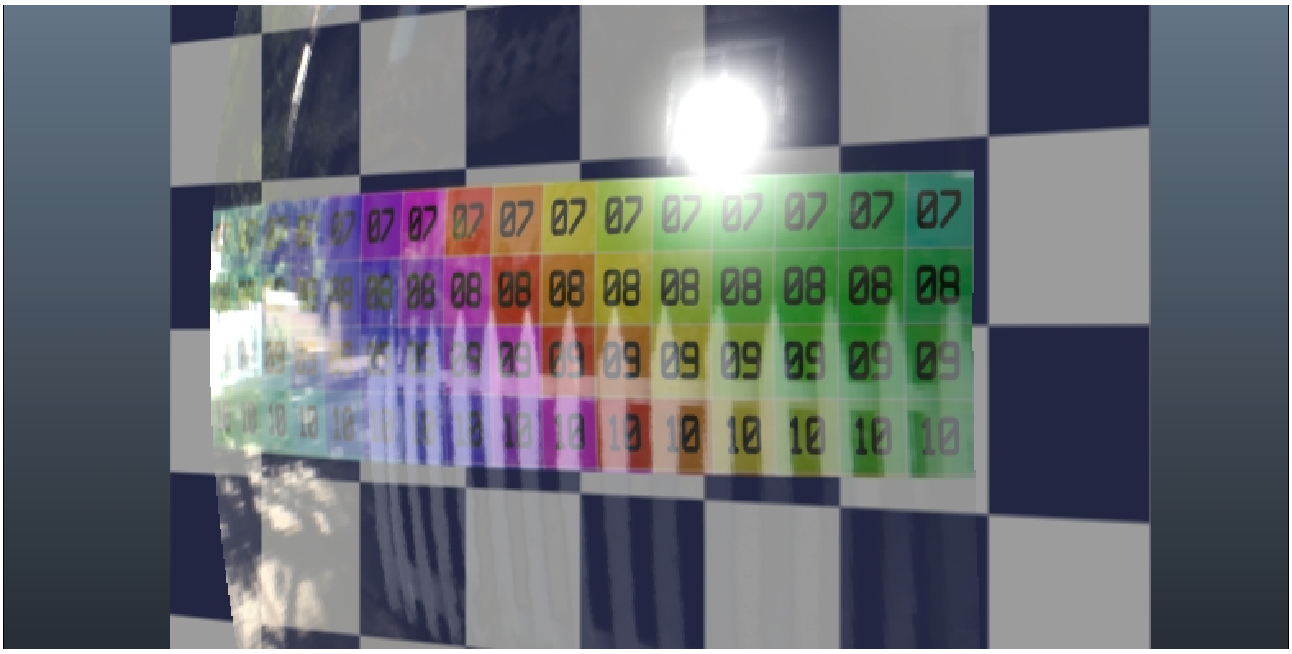
This parameter will essentially push back the light terminator line the higher the parameter value is set. This is intended for mimicking how light would scatter inside the skin and "reach" a bit beyond where the light would end if the object was completely opaque.



Left: *Using a terminator push value of 0.25.* ***Right:*** *Using a terminator push value of 0.0.*

Specular and Reflections on Transp Surf

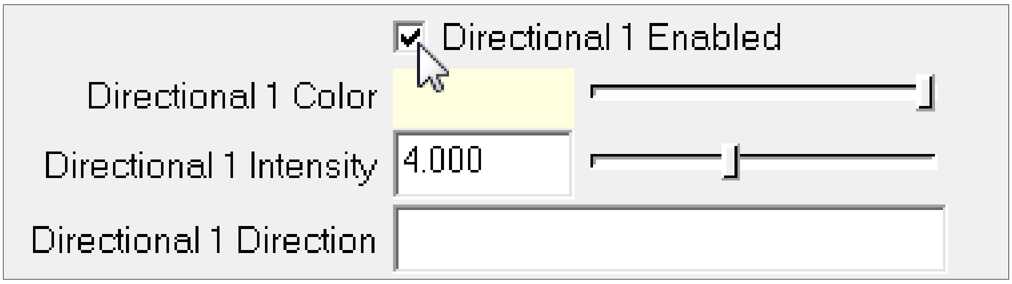
This parameter enables specular highlight and reflections to appear on transparent areas of a model. For more information on transparency in this shader please refer to the Getting Started → Shader Techniques.



A spherized polygon plane with transparent areas which receive specularity and reflections. The plane in the background is there to accentuate the transparency. Yes there is a picket fence in the cube map.

Light Enabled

This parameter exists on all lights and basically enables/disables the current light. This is useful if you just temporarily want to disable a light without having to change it's intensity parameter back and forth.



☑ Directional 1 Enabled

Directional 1 Color

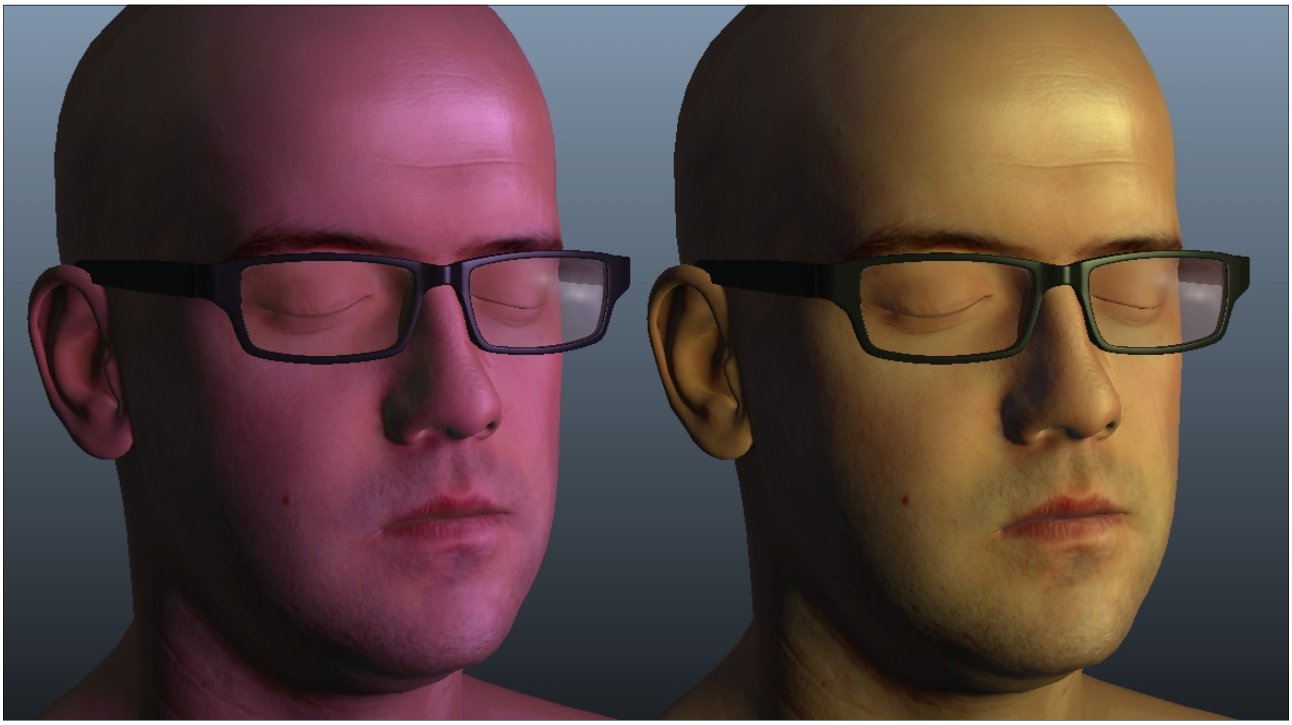
Directional 1 Intensity 4.000

Directional 1 Direction

The "Light Enabled" parameter.

Light Color

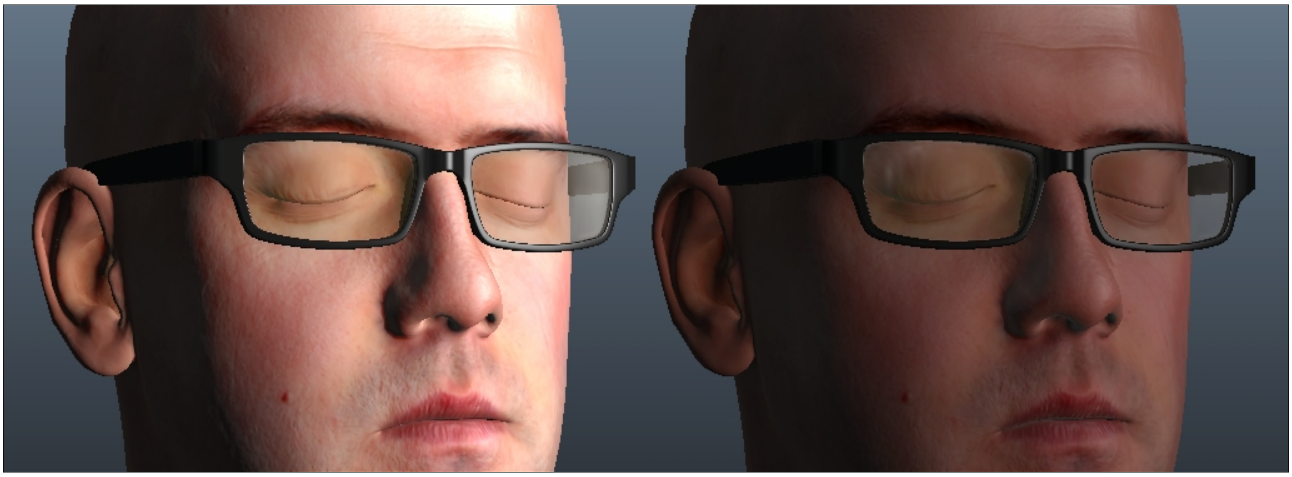
This parameter exists on all lights and is used to set the light color.



Left: *Purple colored light.* ***Right:*** *Lime green colored light.*

Light Intensity

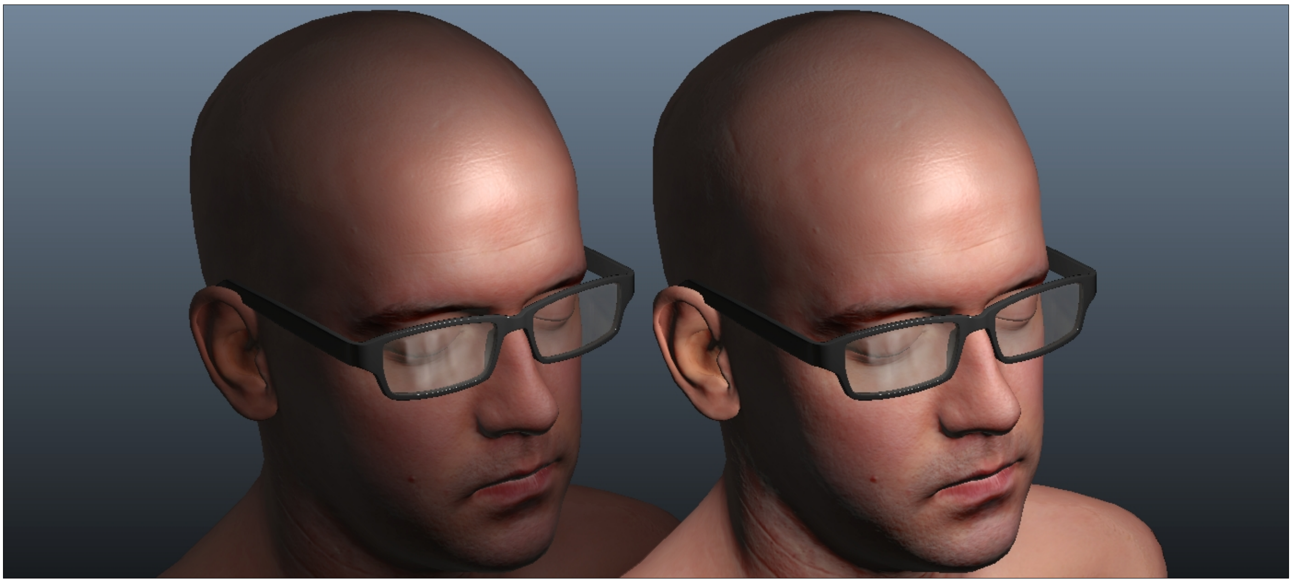
This parameter exists on all lights and controls the lights intensity. A high parameter value will result in a intense light and vice versa.



Left: *Light intensity of 4.0.* ***Right:*** *Light intensity of 1.0.*

Point Light Distance Falloff Exp

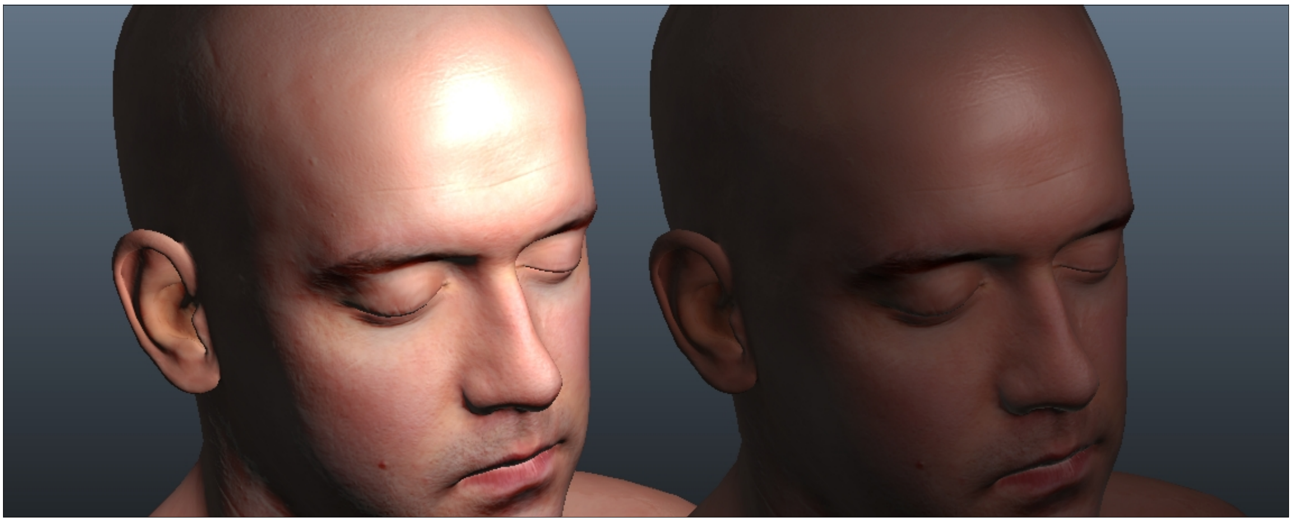
This parameter will apply and exponential type of curvature to your point light's falloff distance. A higher parameter value will give more of this effect.



Left: *A distance falloff exponent of 1.3, light intensity increased to achieve similar intensity at light center point.* ***Right:*** *A distance falloff exponent of 0.0.*

Point Light Distance Scale

This parameter works in conjunction with the Distance Falloff Exponent and will scale the distance of the lights reach.



***Left:** A quite low distance scale value. **Right:** A rather high distance scale value.*

FAQ & Troubleshooting

My shader is all green/white and nothing seems to work.

You probably forgot to enable "Hardware Texturing" in your specific Maya viewport. You can enable this in the viewports "Shading" menu or by pressing "6" on your keyboard.

The lights seem very "blown out".

Check and make sure that you did not scale your object which you have linked to a shader lights direction/position slot. More about this issue in the Getting Started → Linking the lights, at the end in the warning note.

When using transparency objects seem to disappear or have shading errors.

This is due to Maya having trouble handling transparency properly. As far as I have been able to tell this issue seems mostly related to using several layers of transparency and having semi-transparent transparency (not 1-bit). Unfortunately there's not much I have been able to find out or been able to do to remedy this issue.

I get pink error messages when loading the shader.

So do I. To my knowledge this is an error which comes along with exporting Mental Mill CgFX shaders. It does not seem to have any actual effect on the shading results but I also find it annoying. If someone knows how to get rid of these then please do send me an email and I shall forever be grateful.

Is there no support for shadows?

This is an issue where Maya limits authoring real-time shaders as far as I know. Maya does have some form of real time shaders with it's native directional light and correct viewport settings. But beyond this I have yet to seen an external real-time shader produce any shadows in Maya nor have I found any information pointing towards this being possible.

I don't have any CgFX shader material node.

Then you probably forgot to load the "cgfxShader.mll" plug-in or Maya doesn't like you and didn't load the plug-in this time around when starting Maya. More info about this in the Getting Started → Loading the CgFX Shader plug-in.

Credits & Contact Info

Author: Kostas Gialitakis

kostas.gialitakis@gmail.com

http://www.kostas.se/shader

http://www.linkedin.com/in/kostasg

The head model, the head model's diffuse texture and the head model's normal map are those of Lee Perry-Smith. A big thanks for sharing these with the 3D community.

For some of the example images using a Cube Map I've used a few HDR probes from Keith Burns site. Thanks Keith.

When making this guide I searched for a screenshot application which would include the mouse cursor. I found a free application called "Prt Scr" which did the job well. I didn't expect to find such a potent application for free so it gets a mention in this document.

Shout out to everyone on the polycount boards!