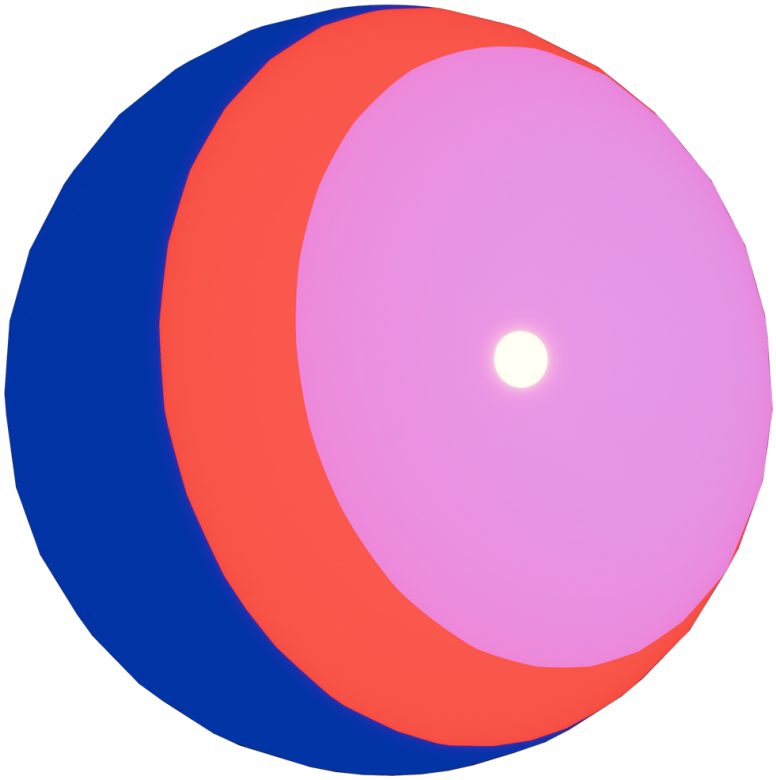
Instructions
Finathai's Toon Shader Collection for Blender Goo Engine

Requires Goo Engine 4.1+ (https://www.dillongoostudios.com/gooengine)

This is the standard clean toon shading you can get, using the core shader. The positioning of the shadow and mid tone is adjustable, as well as the specular effect.



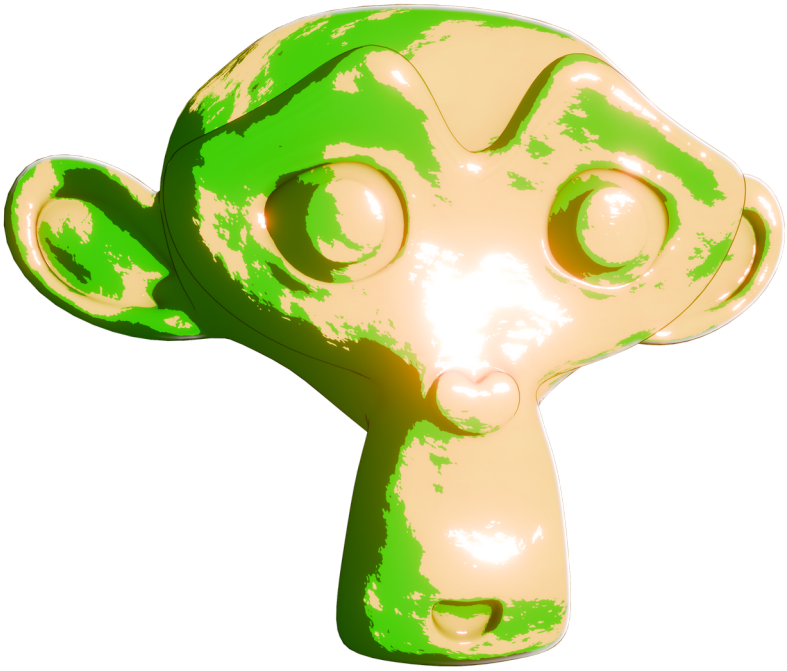
Something to note here is the shadow and mid tone colors, which get multiplied with the base color. This can result in the actual shadow and mid tone colors to not adhere to your set color.

For example, in the above image, if you set the shadow color to green instead of blue, you get a black shadow.

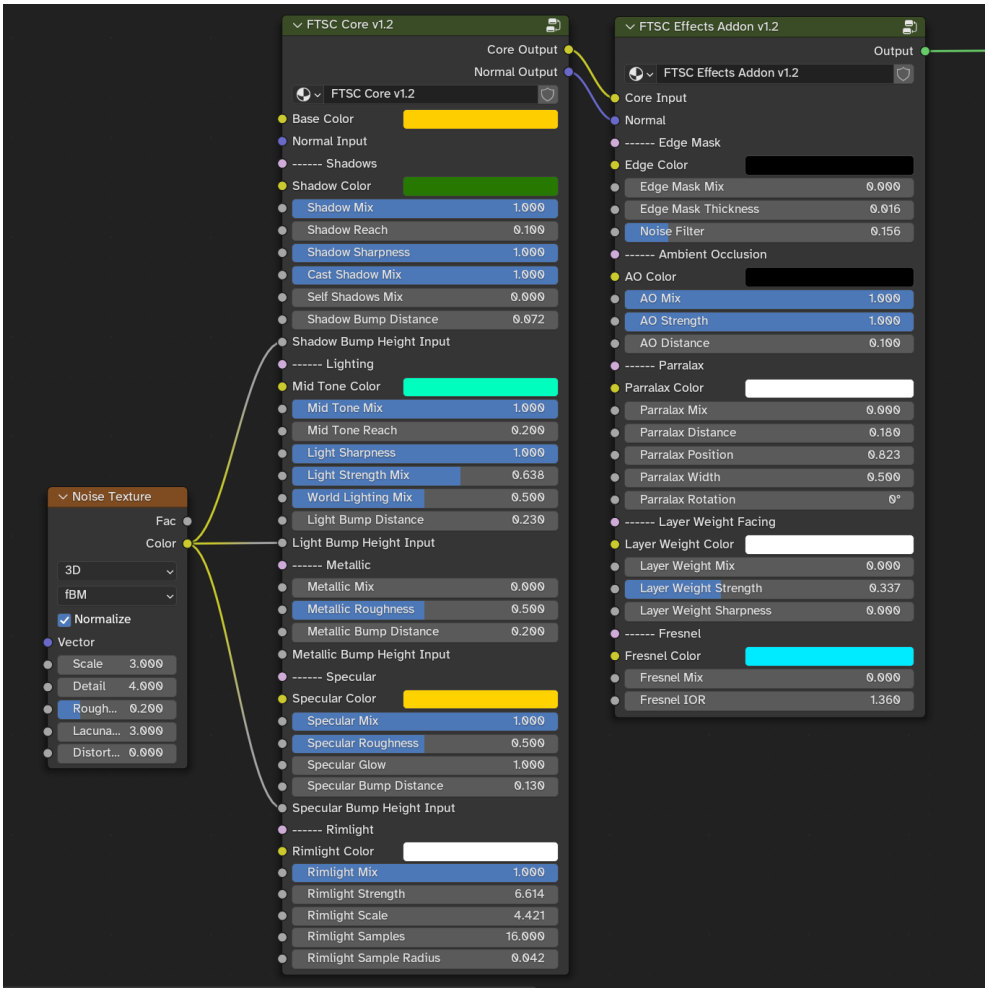
Please let me know, if there is a better way to handle shadow and mid tone colors.

The sharpness of these areas can be freely adjusted to mix in realistic lighting.

You can achieve a painterly effect with the use of a noise texture.

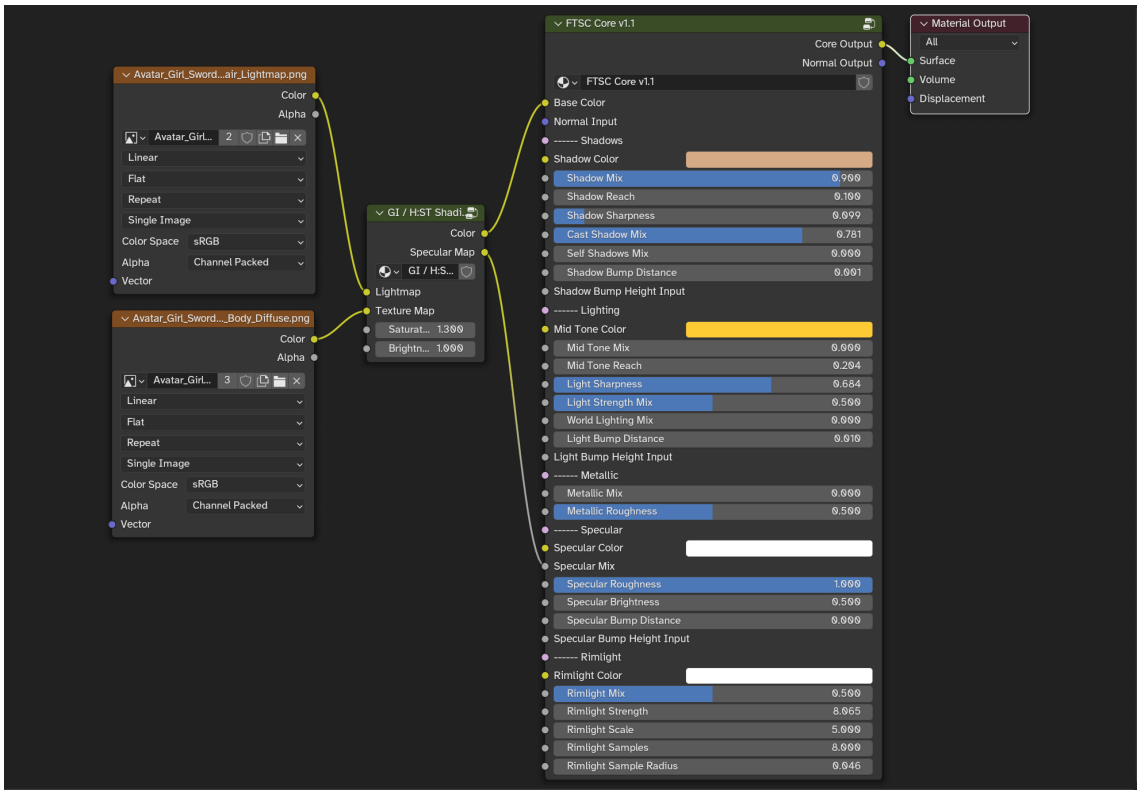


The 'Effects Addon' gives you control over general effects. Included in there is an interesting parralax shader by Xetirano (https://www.youtube.com/watch?v=g3PR4uzbXjs), which can give you fake reflections for glass panes or whatever you like parallax to do, like depth illusion and such.



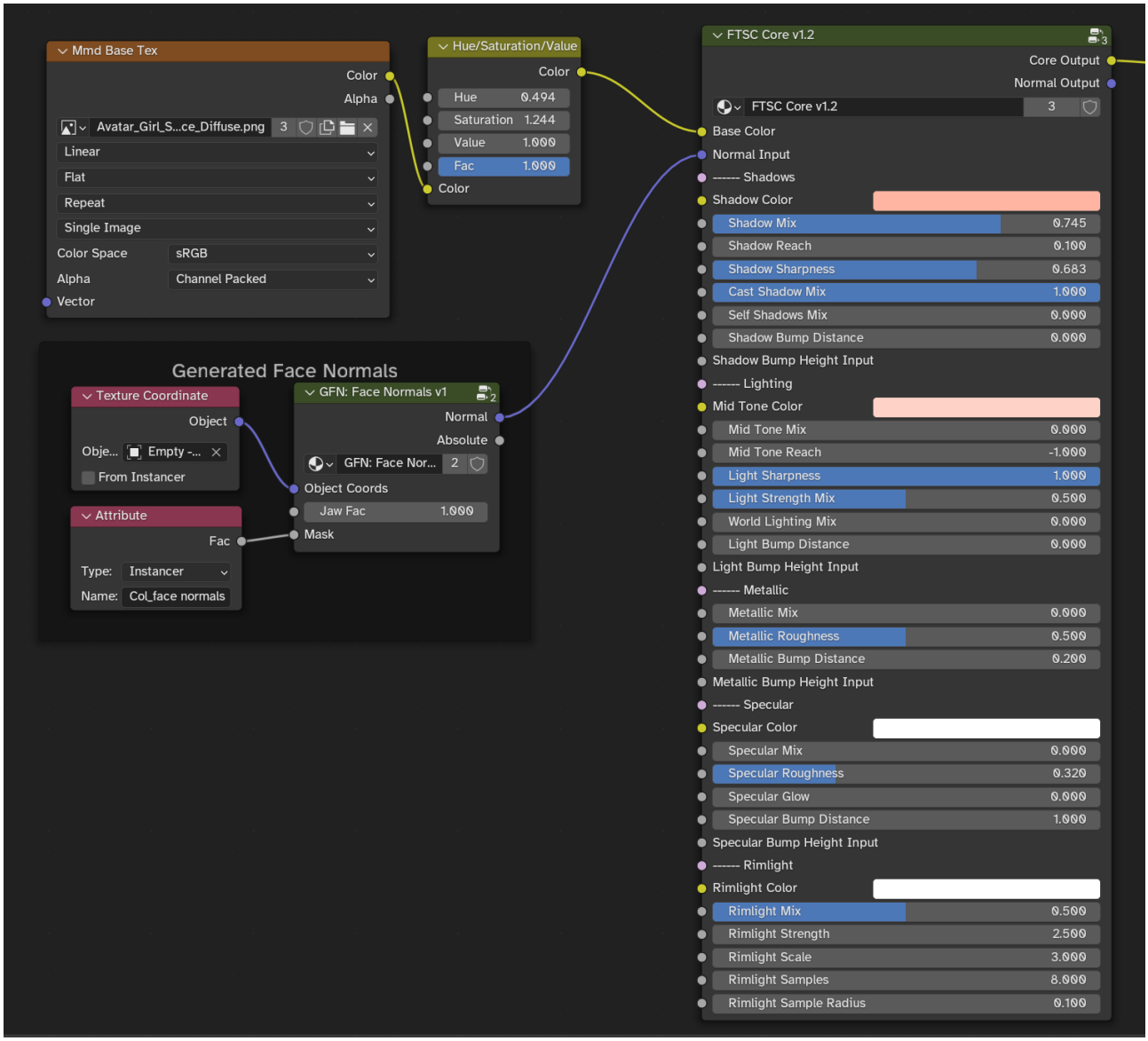
Character Shading for Genshin Impact, Honkai: Star Rail and Zenless Zone Zero

You can achieve this look by simply using my core shader, like in this example of the hair shading:



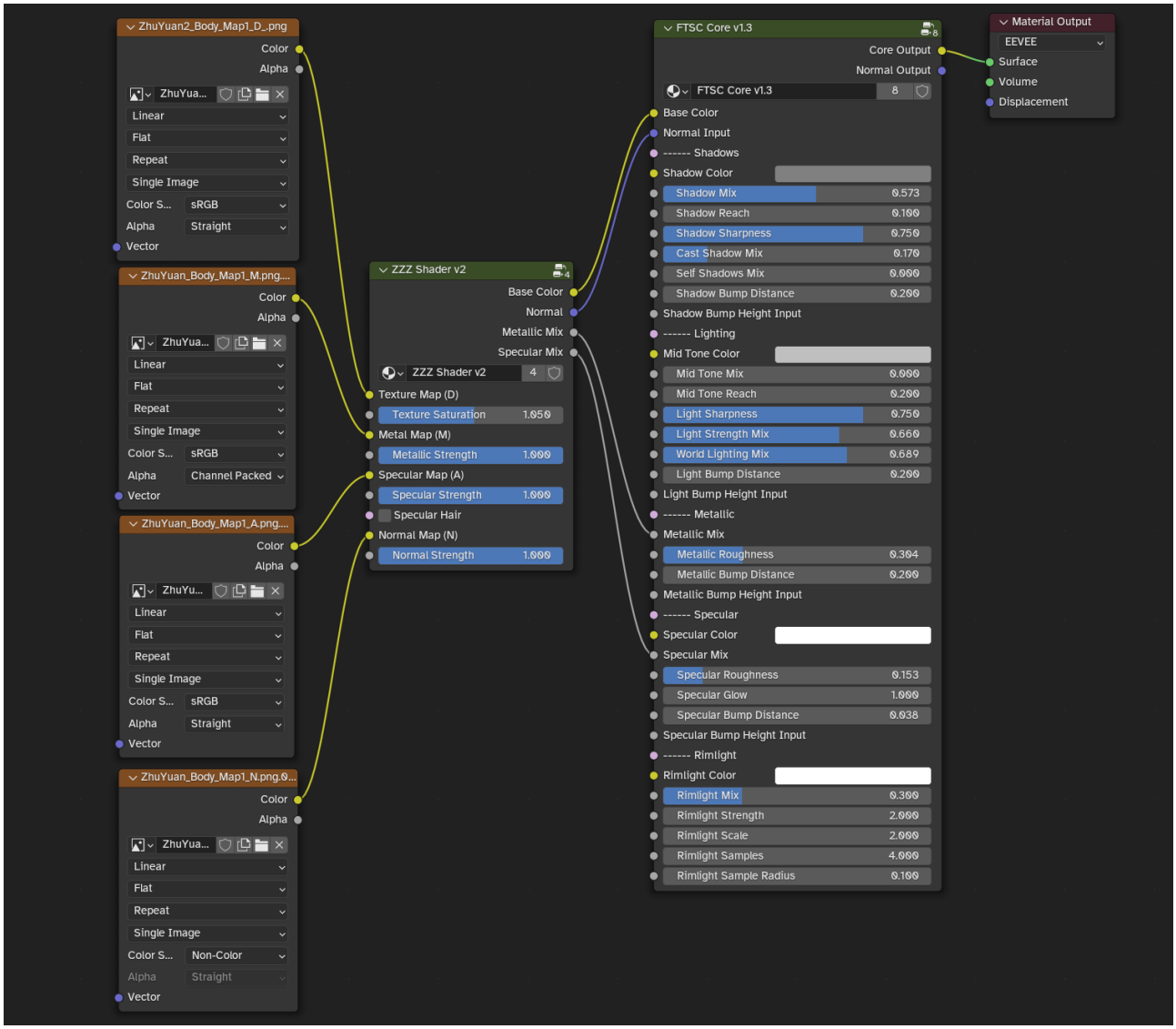
Lightmaps, which get split into the RGB color channels make the highlights on the hair texture shine, as well as metallic bits on clothing.

For clean face shading, you can also follow this guide by aVersionOfReality: https://www.youtube.com/watch?v=sQW2wqltB0A



Just use the 'Normal Input' for this.

A node group for shading Zenless Zone Zero characters is also included:



Make sure to set the normal map to ‘Non-Color’. This setup converts the red and green channel normal map from Unity to a Blender compatible OpenGL normal map.

The Genshin Impact / Honkai: Star Rail compatibility node group contains some leftover node setups from Dab_neko to make character shading look correct.

Considerations:

- Use 'Check Self Shadowing' in material settings to filter out self shadows.
- Shadow and Mid Tone colors get multiplied with the base color, resulting in the actual appearance of these colors being heavily dependent on the base color.
- Specularity is affected by light strength, but not 'Light Strength Mix'.
- 'Shadow Mix' and 'Shadow Reach' need to be both turned to 0 to deactivate shadows, same with the Mid Tone.
- There is always going to be a slight shadow with both 'Shadow Sharpness' and 'Light Sharpness' turned to 1.

- The 'Light Strength Mix' adds subtle light color.
- When plugged into the material output, keep in mind that this results in an Emission with the strength set to 1. You can turn the overall material brightness down by using an Emission shader after the node groups.
- Performance seems alright, only use the 'Effects Addon' when you need it.

Notes

I have included additional node groups like the 'Get Light Colors' which just outputs the color of the lights and you can choose between standard normals or object normals. Could be useful.
There is also the 'Edge Highlight' group which uses Goo Engine's Curvature to get Edges which might not actually be useful at all, but it's there.
Lastly, there is a simple 'Transparency & Emission' group.

The insides of all the node groups are well ordered and labeled so you can get to understand the inner workings, if you like. If you find some improvements, I'd be happy to implement them.

The project file includes my usual settings for Blender, like the compositor. I also set up Freestyle outlines that work quite well for replacing the solidify modifier outlines for every object. This increases viewport performance at the cost of render time. The compositor is set up to mix bloom with the lineart, making it look pretty good. Freestyle lines also avoid the problem of solidify outlines clipping into eachother and you get actual paint strokes which looks great. It's a bit finicky to use, play with the settings to get your desired line look or deactivate it.

Place objects in the lineart exclusion collection for them to get excluded from Freestyle lines. You can also use a second view layer with Freestyle turned off to control what gets lineart.

Feel free to tweak everything to your liking, this setup is likely not perfect, but I'm very happy how it all came together.

For in-depth discussions or help, please join my Discord server (NSFW, link provided separately).