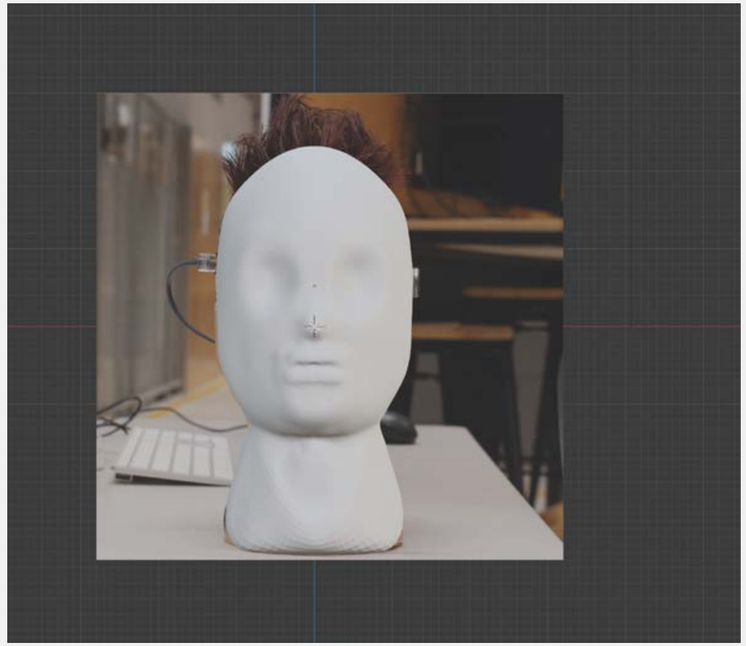
Cyberthug Delta
Rationale

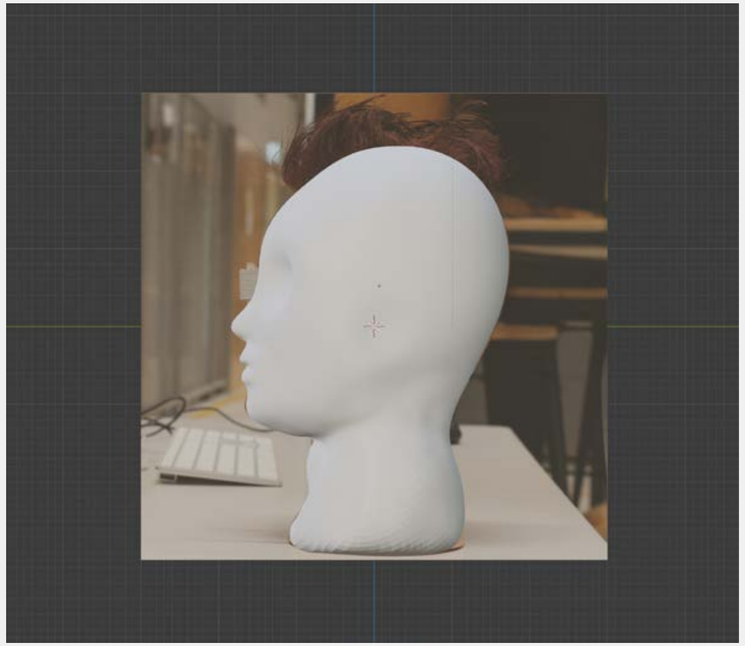
Although it wasn't required for the assignments, I wanted to try to recreate the Cyberthug from semester 1 in Blender (later Maya), just as a side project/supplementary material.

Something about having both the physical model from sem1 and the digital remake next to each other seemed really appealing, and I wanted to see how much detail I could add. I find I'm more motivated to learn software when I have a project to apply the skills to.

The cool thing about this was that I was able to have both the character and the artefact together in a seamless integrated CGI environment. It's an industry that's higher up in my list of interests, so this could possibly be extended into my major project for next year's Bachelor degree (perhaps a Cyberthug short film).

As with the Cyberthug Core, I decided to use Pixar's RenderMan engine, mainly out of brand loyalty and familiarity.

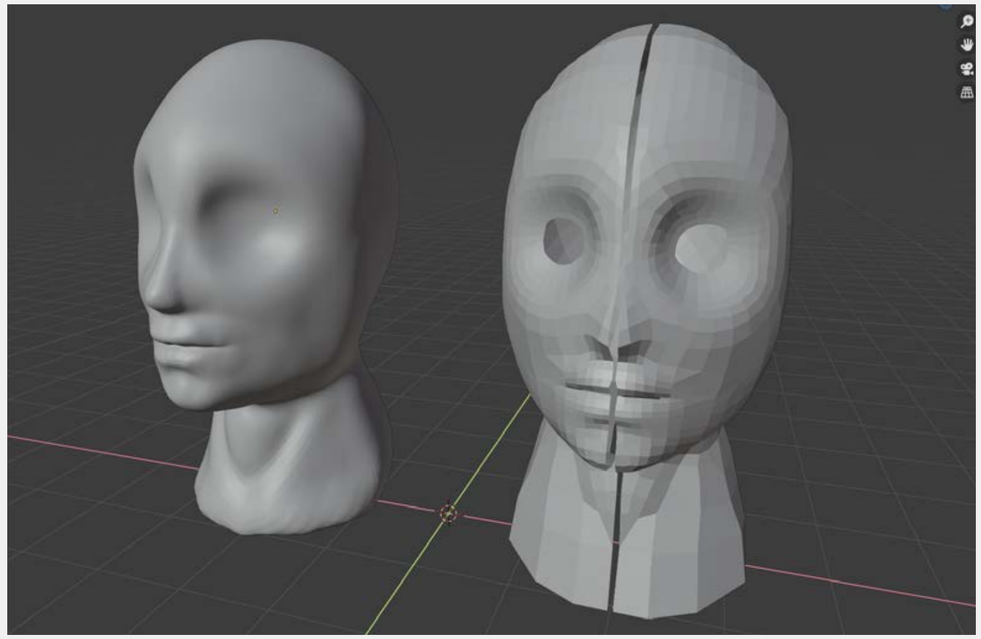


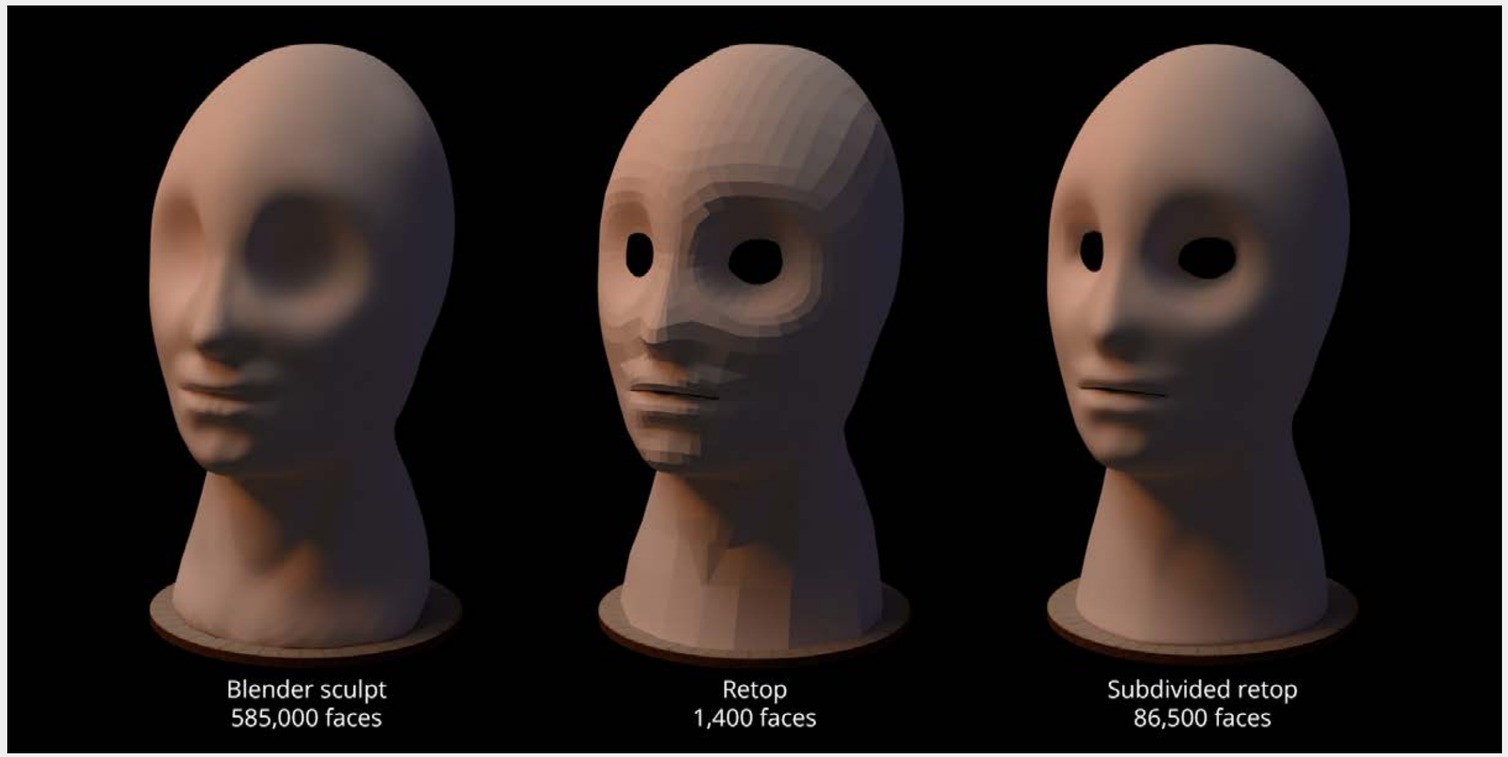


Cyberthug Delta
Sculpting

To begin, I took photos of the physical Cyberthug model from as far away as possible to get a flat, orthographic perspective. Then I made sure they lined up properly, before bringing them into Blender.

I created a sphere and used Blender's sculpting tools to shape it to the reference images. Once most of the shape was formed, I had to rotate around it to fix a few things. I made sure that I had mirroring along the X axis enabled to keep the symmetry.



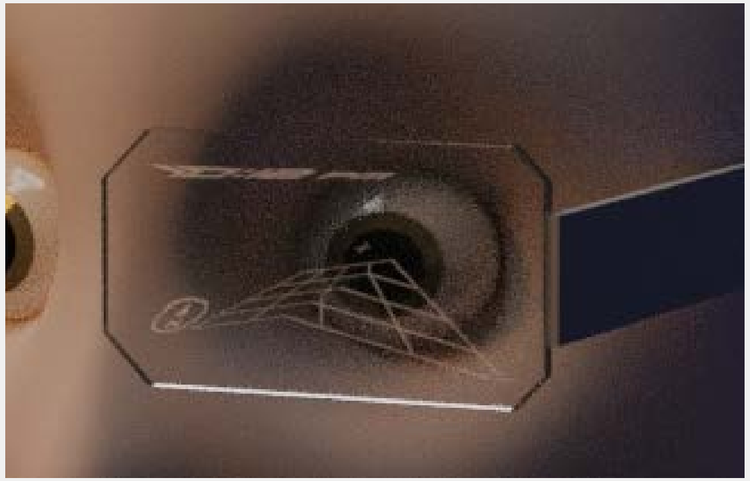


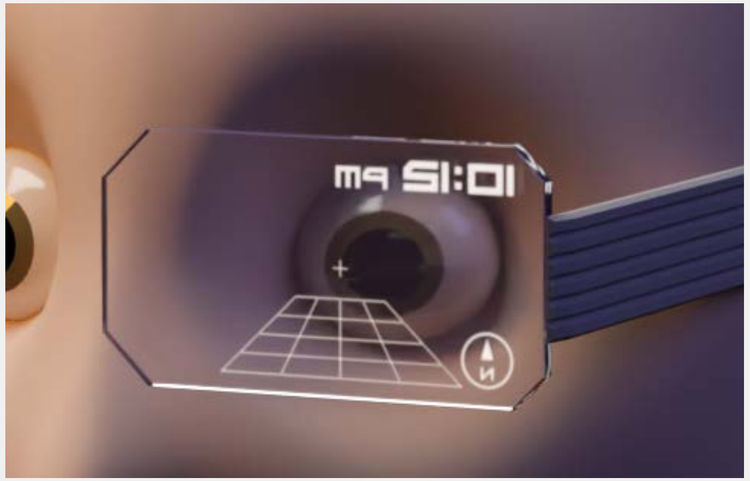
Cyberthug Delta
Retopology

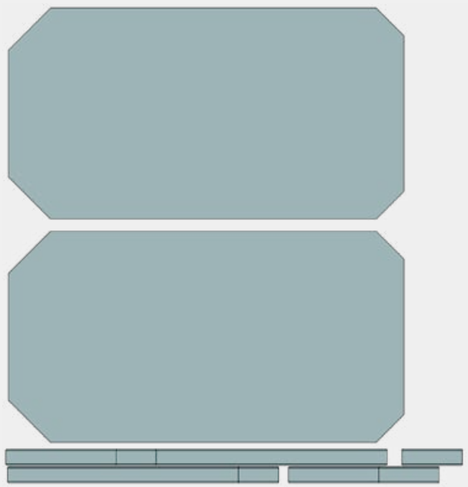
An important step in the modelling process is something called retopology. This is where you use a sculpted mesh as reference (which typically consists of a unnecessarily high polygon count) and go over the surface of it with quads. This is a subjective process and involves a bit of guesswork when it comes to where the quads should sit.

I only needed to do one half of the character's head, then used the mirror modifier. I thought it would be best to leave a gap in the middle so the sides wouldn't be perfect halves.

I bridged the middle edge loops and enabled smooth shading.





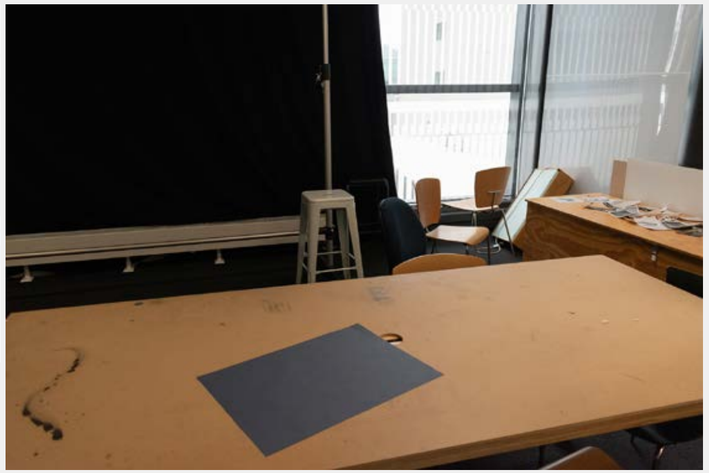


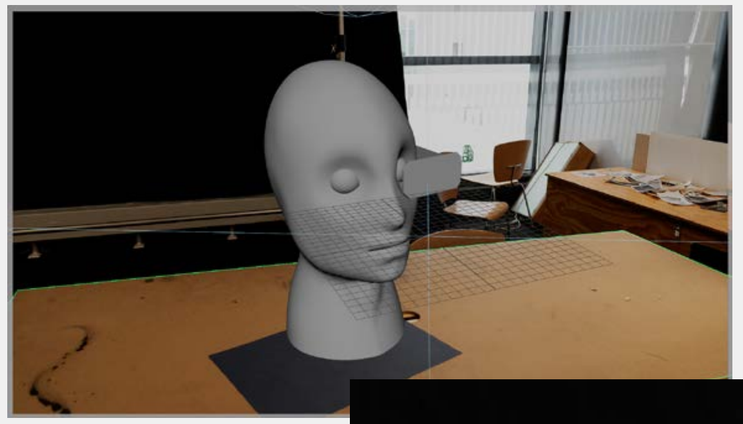
Cyberthug Delta
Texturing

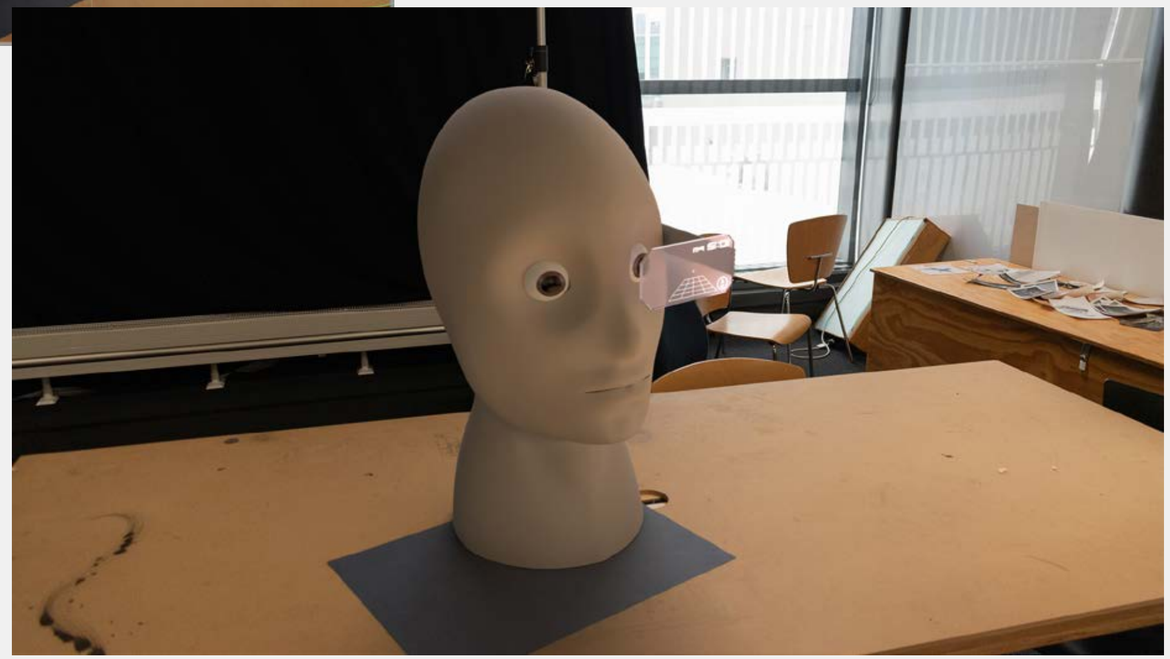
Texturing is obviously a crucial part of the CGI process, and the workflow I used in my previous animations wasn't exactly good practice.

At the top is the HoNeDi texture applied to the corresponding model before and after correcting the UV map. This was my first time UV mapping something so I'm pretty impressed with it. An emission map was also applied to make the text emit light.

Below that are the images for the UV, diffuse and emission textures. The UV map isn't technically a texture, but rather a layout or guide to use when making the others.





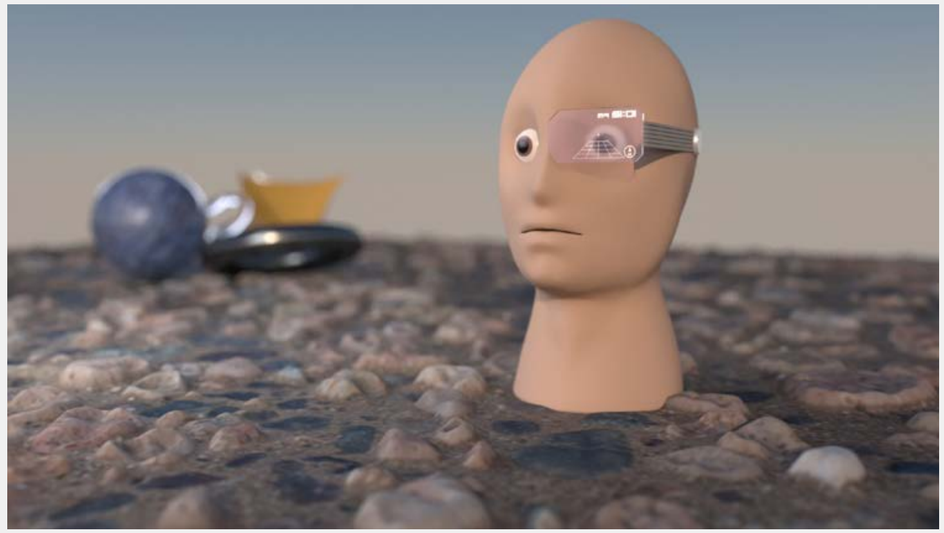


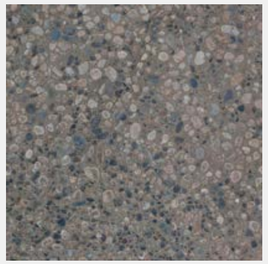
Cyberthug Delta
Environment Integration

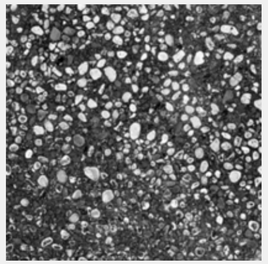
While it turned out to be a tedious process, I wanted to try having the CGI Cyberthug appear to be in a physical setting. To do this, I took a 360 degree panorama with my DSLR's HDR bracketing enabled. This resulted in about 400 shots, which I then converted to an HDRI in Lightroom (the warped image in the top-left). I set this as a spherical dome light in Maya.

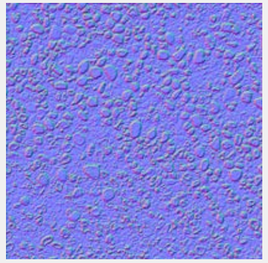
I then created a plane under Cyberthug Delta which matched the table in the reference image. I applied a RenderMan holdout to this plane which I think made it render the shadows but not the colour itself (the engine took the lighting in the HDRI into account for realism).

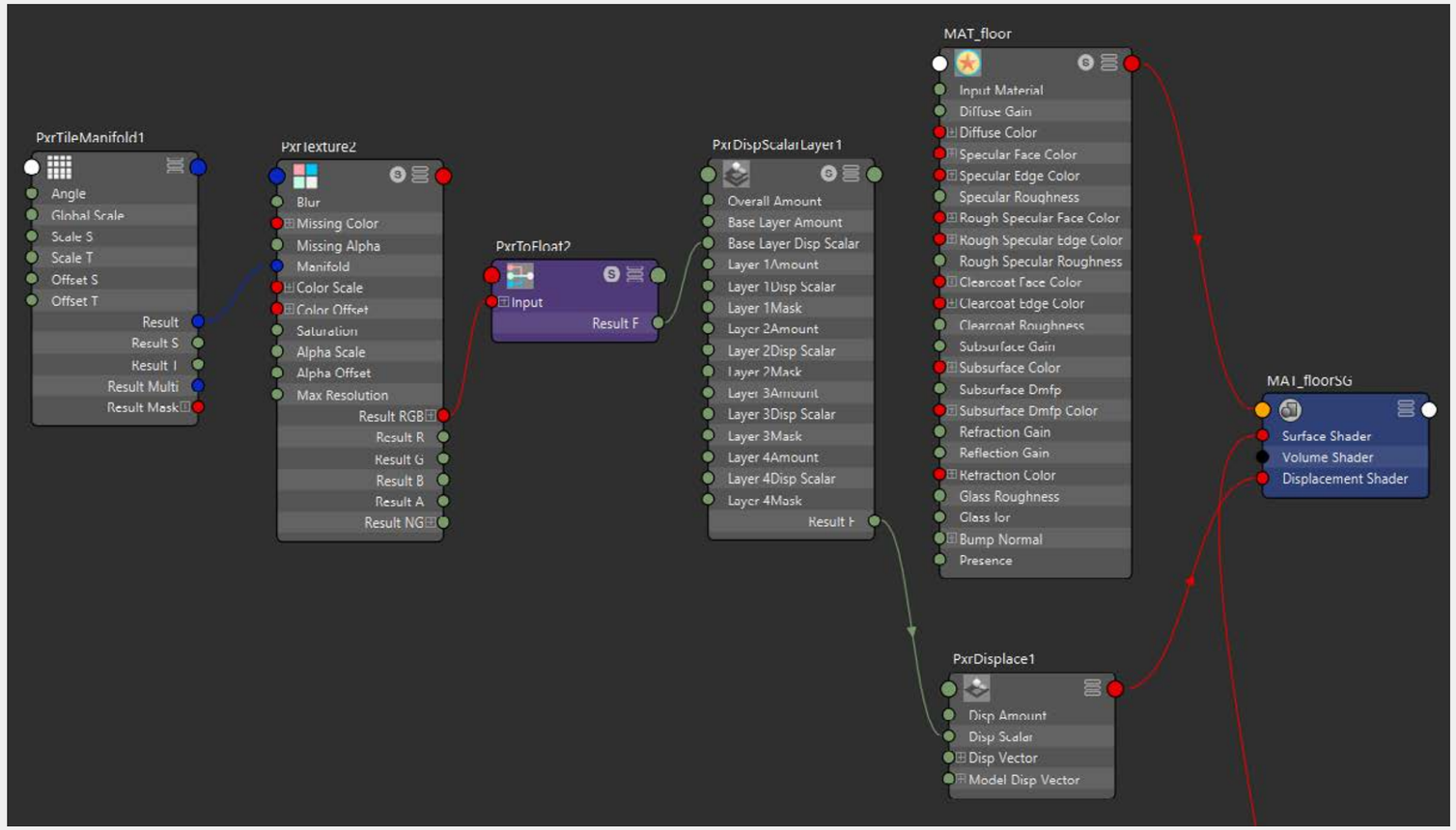
I had a brief go at this in 2017 at Yoobee, but that involved TheFoundry's Nuke software, which is expensive. I forgot the process for that, but for this simple render, I was able to just layer the stuff in Photoshop.











Cyberthug Delta

Texturing and Rendering Notes

I followed a tutorial written by the Pixar team about colour management in the CGI environment as I wanted to ensure my renders would display consistently across Maya, RenderMan, Photoshop, After Effects etc. The render in the top-left is Cyberthug Delta added into that sample scene :)

It was then that I learnt about the ACES standard, but that seemed overkill with terms like OpenColorIO and various plugins that had to be installed, so I sticked to the default color space.

The graph at the bottom shows Maya's Hypershade window, with nodes connected for the pebble material. The way I understand it is the PxrTileManifold node handles tiling and placement, the PxrTexture node processes the texture map itself, the PxrFloat node converts it somehow, and the PxrDispScalarLayer node handles how much it displaces the geometry.

The sample project used a normal map in place of this, but included a bump map. Turns out bump and displacement maps are interchangeable as they're both greyscale.

https://renderman.pixar.com/color-management