Anime Cyborgs

Material Showcase

Project Vero, 2022

Scope

The scope of this asset is to provide with 2 testable anime-styled skeletal meshes of androids and a set of 6 materials to obtain an anime-like shader.

The complexity of the androids is quite simple for the purposes of showcase. They have facial expression morphs and some base blueprint functionality for blinking and looking at things.

The materials included provide 4 different types of surface shaders and 2 outline shaders.

Finally, the animations are only the basic ones as per the third person template.

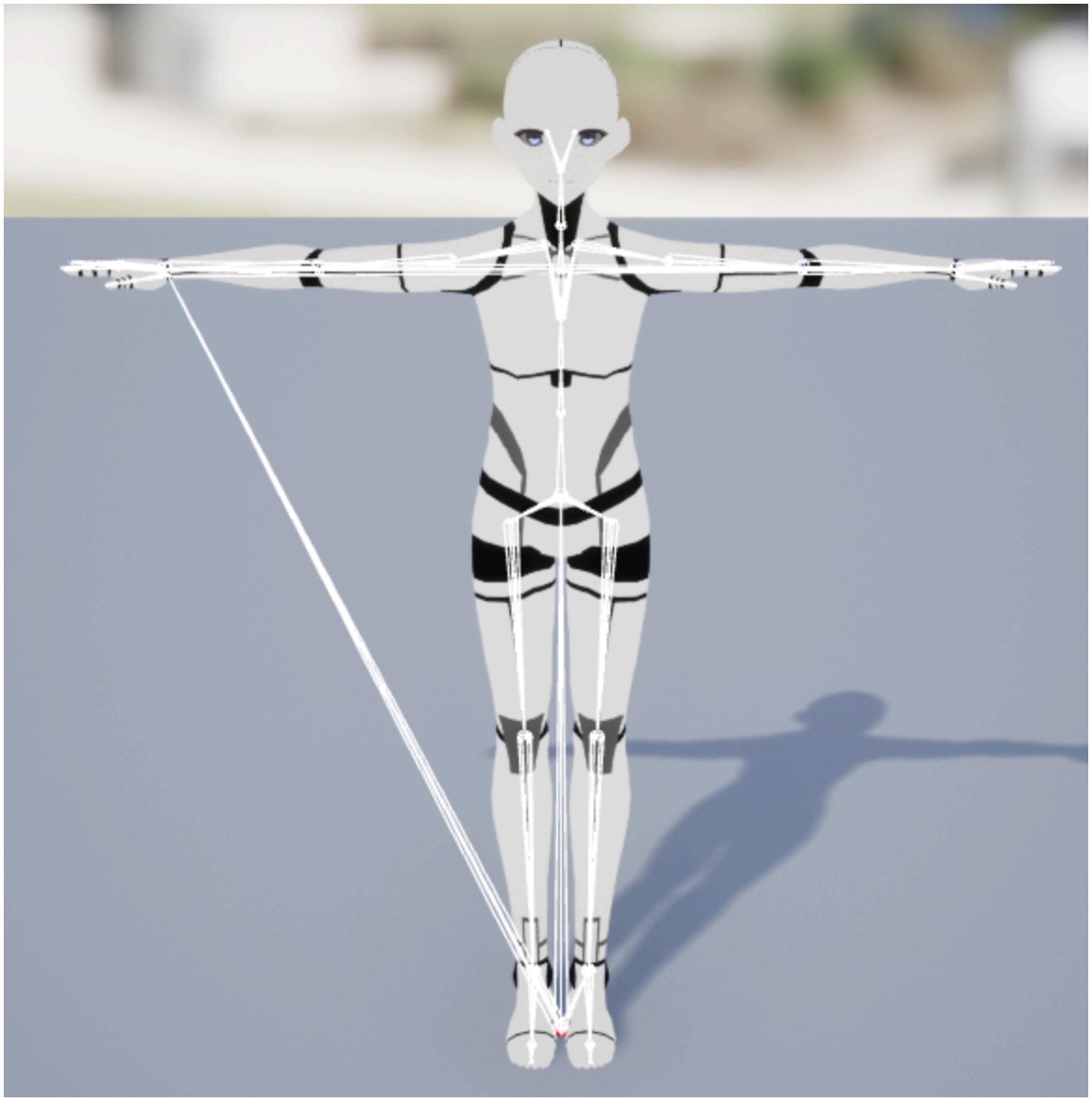
Anything not included above is considered out of scope and will not be supported/implemented in this release. This includes additional morphs, animations, or materials.

Features

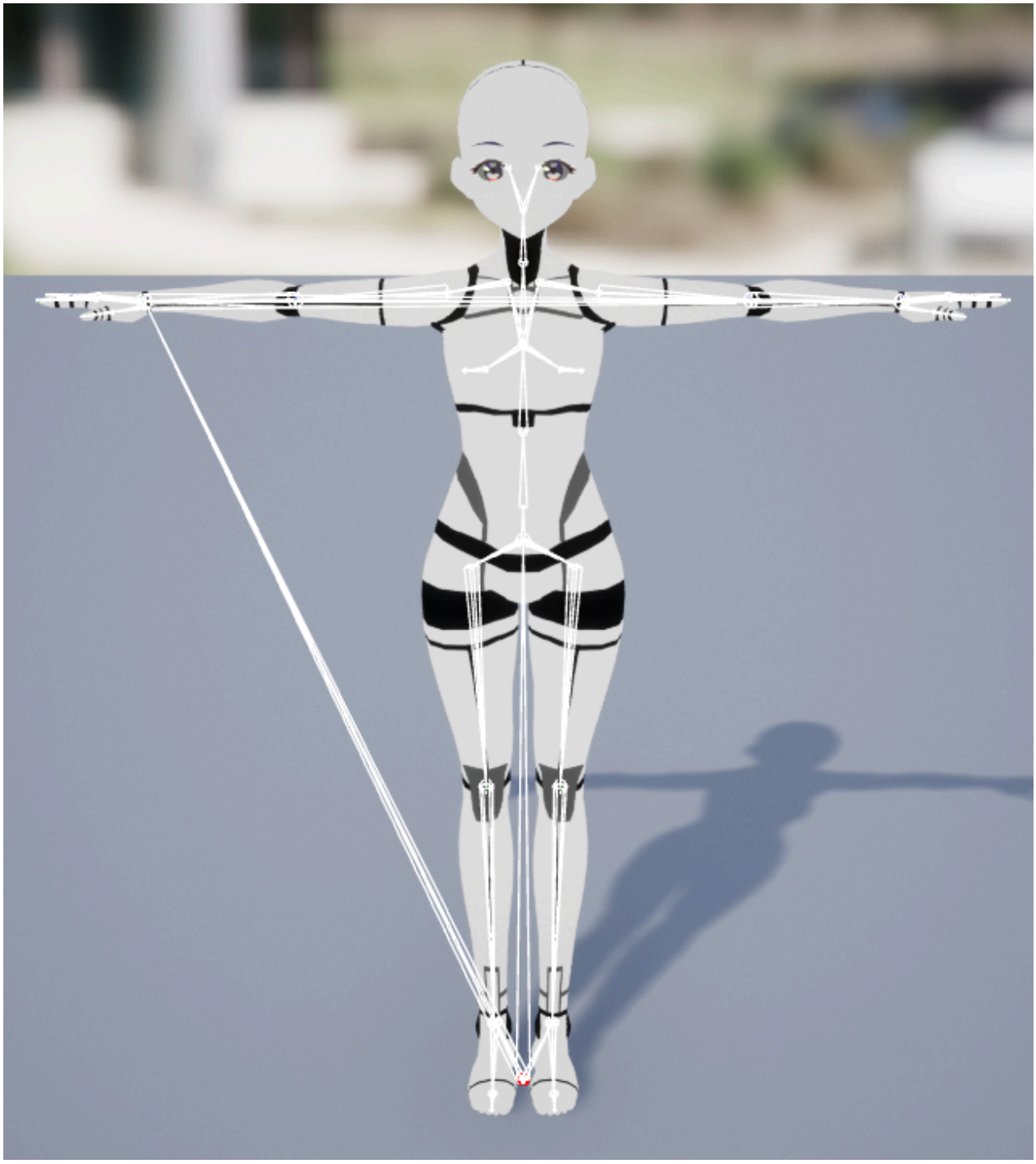
Skeletal Meshes

The male and female androids are rigged to the Unreal Skeleton and include IK bones.

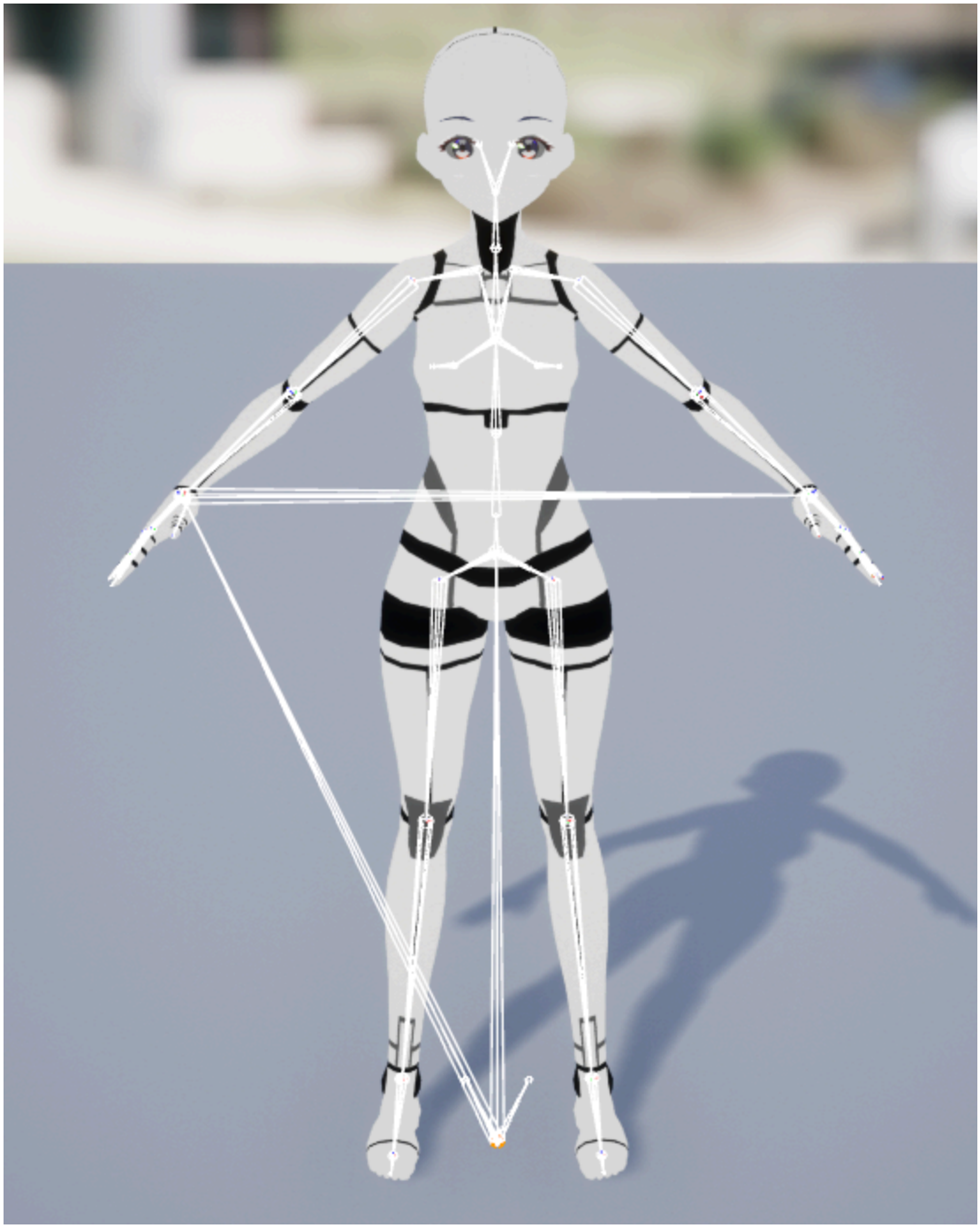
They both include additional bones for eyes and finger/toe tips.



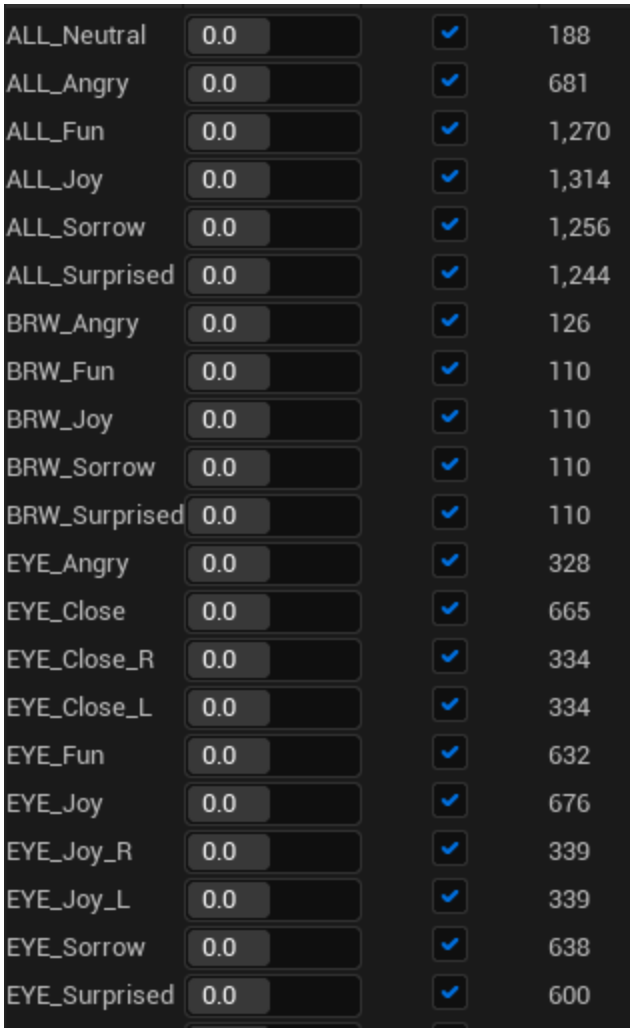
The female android also includes bones for the breasts.



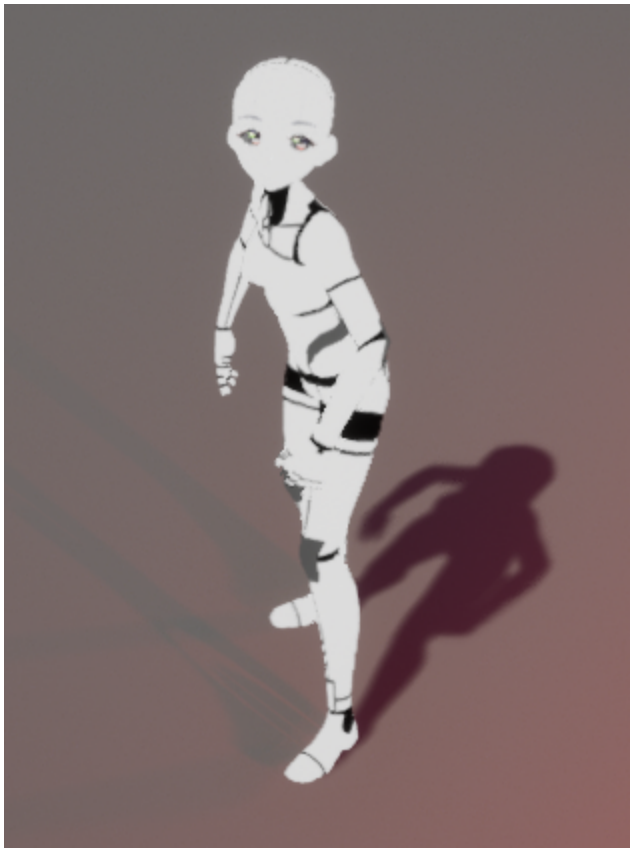
A pose asset was added to switch between T and A pose:



Morphs have been added to control facial expressions, individual facial components and teeth. You can technically use it for vocalizations though no setup is provided to do so out of the box.


ALL_Neutral 0.0 188
ALL_Angry 0.0 681
ALL_Fun 0.0 1,270
ALL_Joy 0.0 1,314
ALL_Sorrow 0.0 1,256
ALL_Surprised 0.0 1,244
BRW_Angry 0.0 126
BRW_Fun 0.0 110
BRW_Joy 0.0 110
BRW_Sorrow 0.0 110
BRW_Surprised 0.0 110
EYE_Angry 0.0 328
EYE_Close 0.0 665
EYE_Close_R 0.0 334
EYE_Close_L 0.0 334
EYE_Fun 0.0 632
EYE_Joy 0.0 676
EYE_Joy_R 0.0 339
EYE_Joy_L 0.0 339
EYE_Sorrow 0.0 638
EYE_Surprised 0.0 600

In the animation blueprint, two overrides were added and controlled through the main character blueprint to control “Looking at” things. The neck area has also been restricted so to prevent twisting of the neck.



Blinking logic is provided in the character blueprint and is set on a timer that can be turned on or off.

All animations provided are retargeted from the UE4 Mannequin and include the idle, jump start, jump, jump end, walk and run animations.

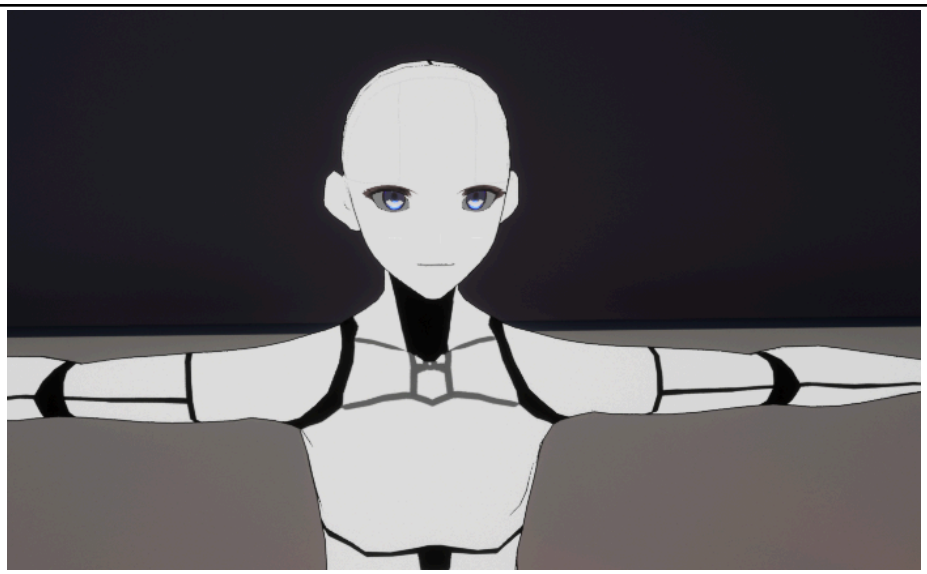
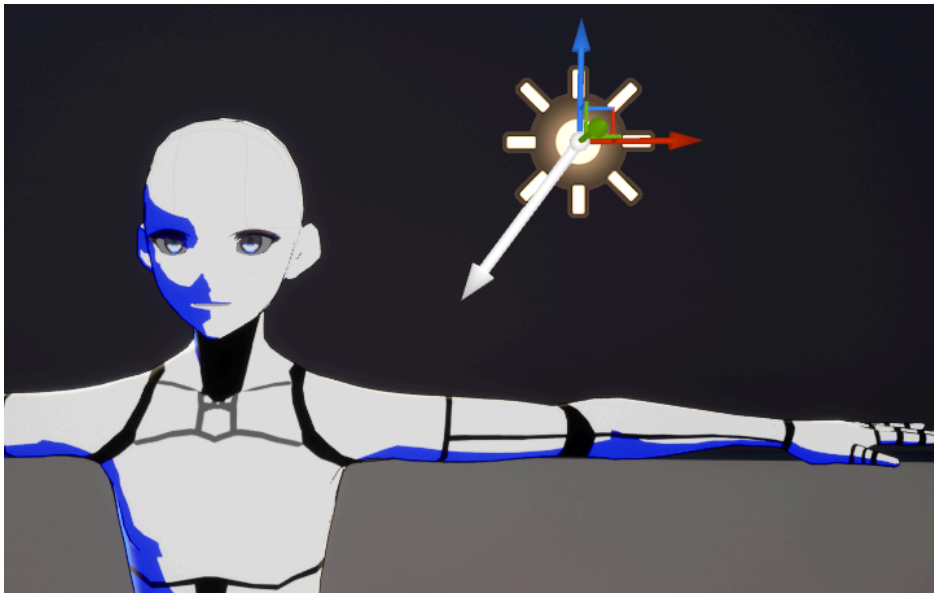
Materials

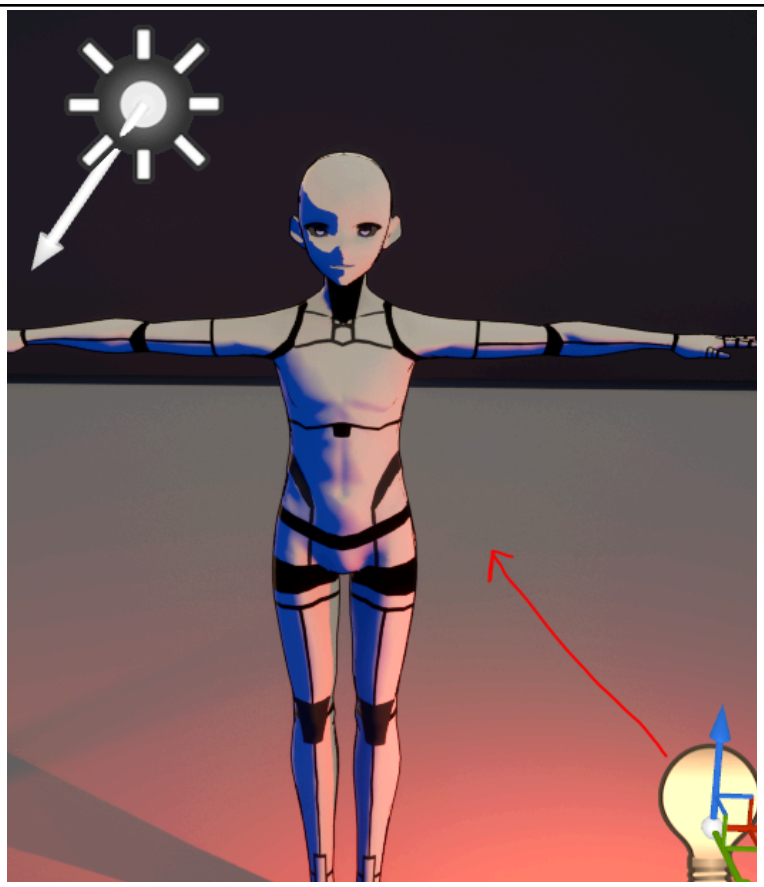
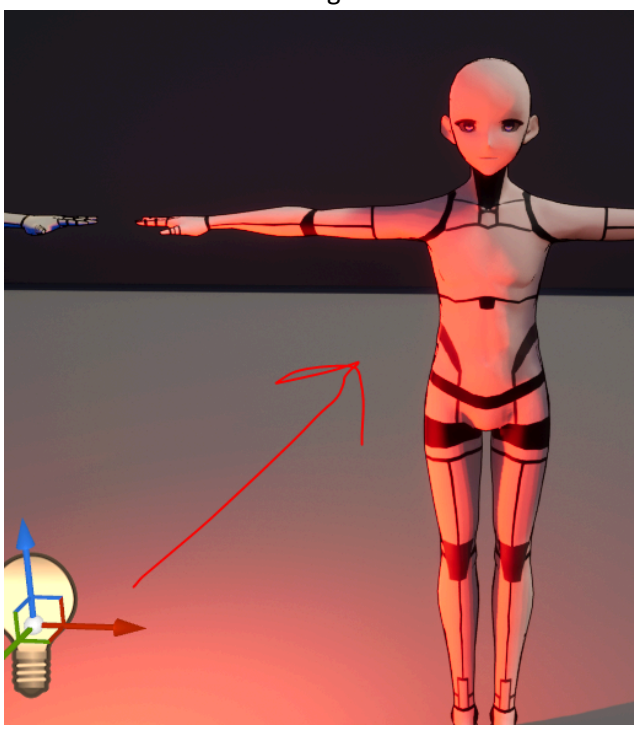
For more information on the materials, see the Materials – In depth section.

All shaders prefixed with M_ are materials whereas those prefixed with MI_ are material instances. It is highly recommended to work using the material instances for quick and easy modifications rather than modify the parent material and have it recompile each time.

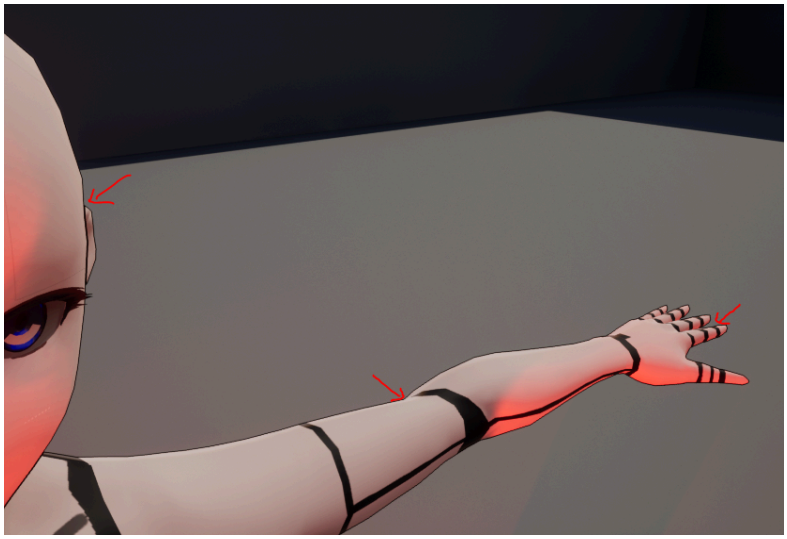
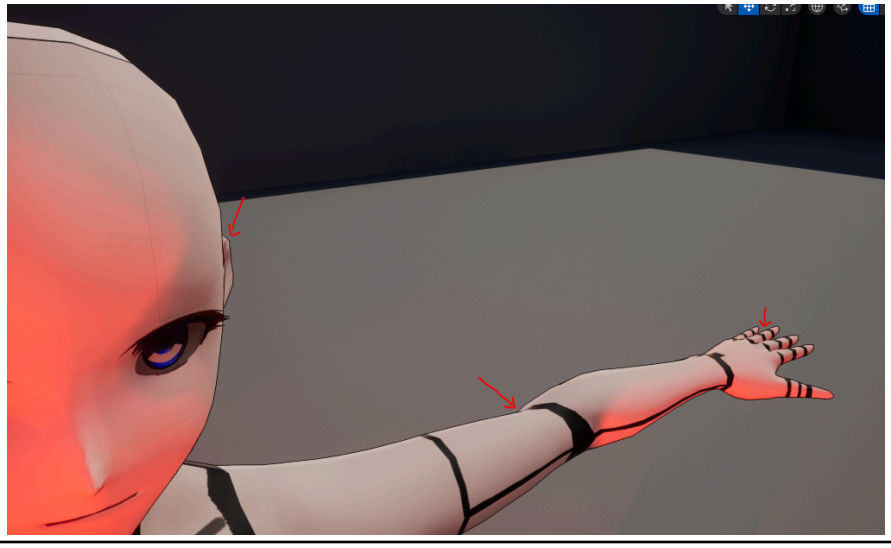
This package includes 4 surface shaders:

Name	Description
M_SimpleAnime	This is a simple emissive material that is unlit. It does not interact with light and does not have shadows.

	
M_ToonBase	This is the base for the next 3 following materials.
MI_Unlit_Base	Uses the master material to process textures and normals. This material will not interact with light. This material will generate fake shadows that are based on the directional light direction. 
MI_Lit_Base	Uses the master material to process textures and normals. Will interact with lights.

	
M_SSS_Base	Uses the master material to process textures. Will emit textures similarly to the M_SimpleAnime but will also interact with light other than the directional light. 

This package also includes 2 outline shaders:

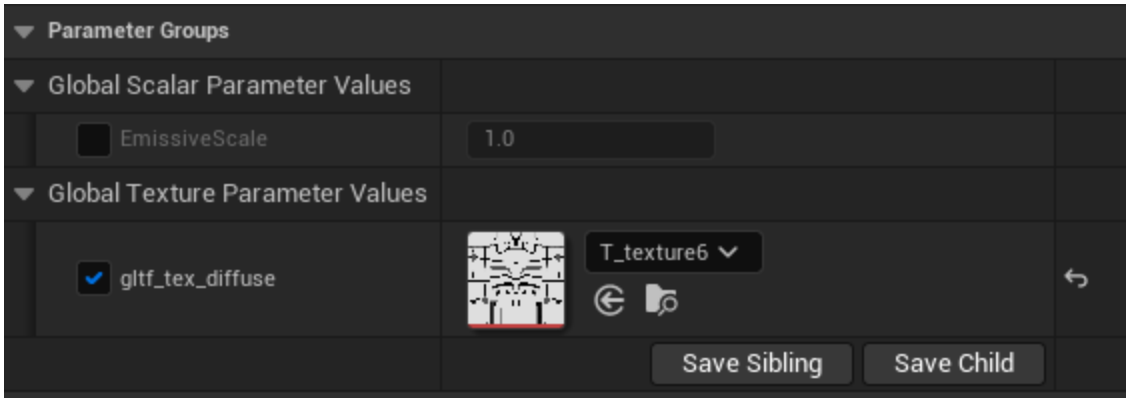
Name	Description
M_Outline_Laplacian	The outline is based on the Laplacian edge detection. 
M_Outline_SpiralBlur	The outline is based on pixel sampling and blurring to offer a fading effect. 

Materials - In depth

M_AnimeSimple

This material is as described previously. It is an unlit emissive material that will show off a texture. Normals are not computed and it does not interact with lights nor generates shadows on itself.

Example:



The only parameters you have are the emissive scale (how emissive you want it) and the texture.

MI_AS_Base

Derived from M_AnimeSimple, use the parameters to customize.

M_ToonBase

This material was inspired and improved upon from another anime-like material with considerable simplification and performance improvement. Although it can be used (as an instance) to generate just about anything, I created 3 material instances based on the master material that are setup for the lit, unlit and subsurface lighting types of shaders.

To use the material instances, you can actually make instances of instances and change the textures so as to keep the originals intact and work on the copies.

Compared to other materials, this one was severely optimized in terms of performance.

The material is quite complex and some options are most of the time switched on.

00_StaticSwitchParam

01_StaticSwitchParam_Extras

These are extra features you can use to add flair. For instance, you can use a second shade color instead of a singular shade.

02_Levels

Both options allow for higher levels of opacity and brightness.

03_ToonShader

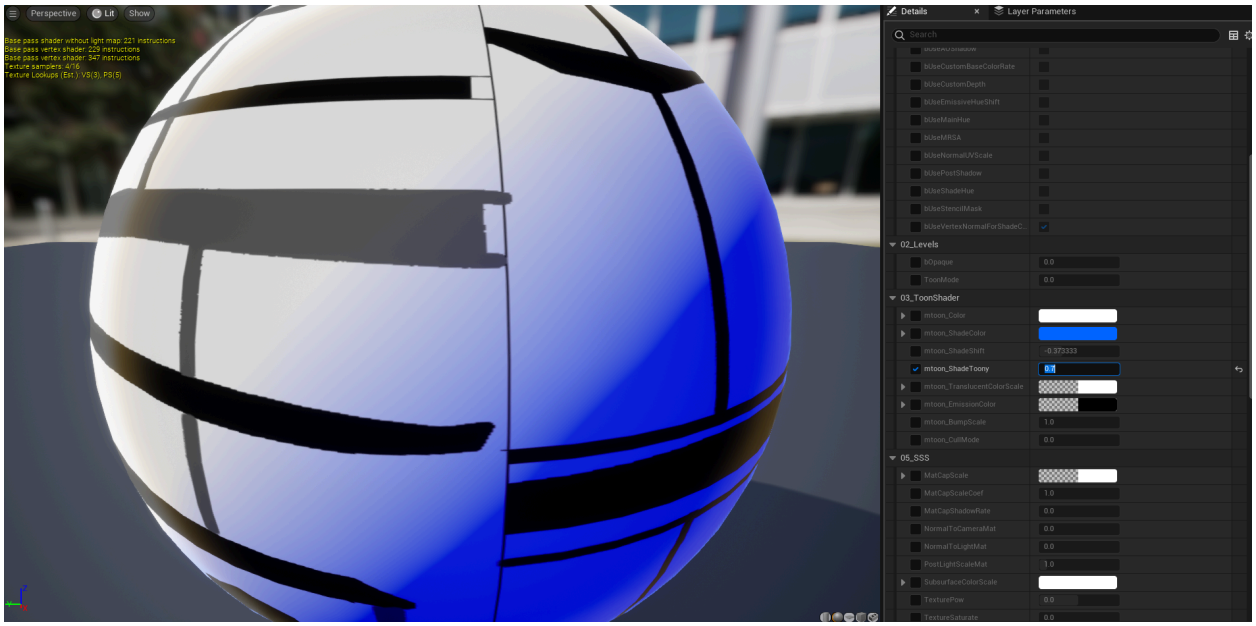
Probably the most important section.

Color will add an additional tone of color to the emissive texture color.

ShadeColor will dictate the color of the shadow.

ShadeShift determines how much of the surface is covered in shade.

ShadeToony determines whether or not you want the shade to be crisp (1) or bleeding (0).



TranslucentColorScale is as the name implies and is used for translucent materials (typically eyes or brows or hairs).

EmissionColor adds another layer of color.

Bumpscale is a coefficient for normal map scale.

Cullmode can either be 0 (two-sided), 1 (front face) or 2 (back face).

04_RimLight

All settings for rim light. This is not turned on by default.

05_SSS

This is for subsurface lighting. All settings will drive the way it looks, there isn't much to describe.

06_Shadows

There is a shadow logic if you have customized shadows however, it is not turned on by default.

Everything else

Every other Global Scalar Parameter Value is a shift, offset or alpha to play around with if you want to have minor changes incurred.

MI_Unlit_Base

This material will use the emissive channel to show the colors (texture) and will compute the directional light and normal to show shadows. You can use the shadow creep feature to dictate where the shadow starts and stops. The shadow will appear opposite of the directional light.

You can also control the level of color shading of your texture.

Finally, you can control the color of your shadow.

This material will not reflect any other lighting such as point lights.

MI_Lit_Base

Unlike the unlit variant, this material does not generate its own shadow but uses ambient lighting for shadows. It will also react to light sources.

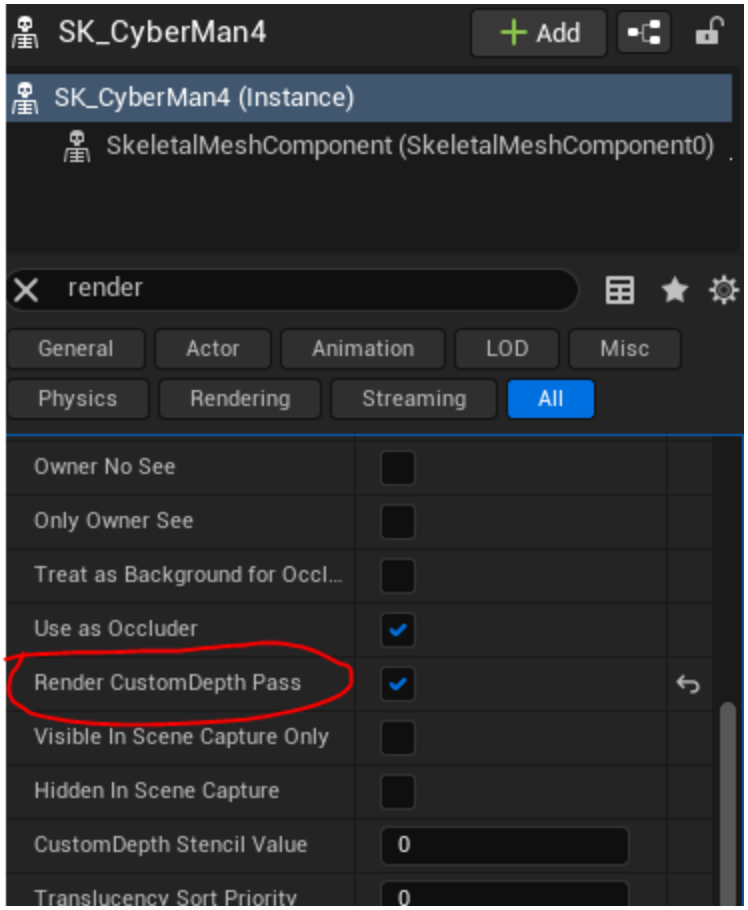
MI_SSS_Base

Subsurface lighting will act as the MI_Lit_Base material in that the textures are shown subsurface light. The model will however be able to receive light from other sources.

Outlines

Two materials are provided for outlines and they are Laplacian edge detection based or spiral blur. The spiral blur is the most expensive one but can generate blurry outlines whereas the Laplacian edge detection is very standard and low cost.

All outlines were relegated to the custom render depth pass so to enable it, select your mesh and search for “render” and check this checkbox:


SK_CyberMan4
Add
SK_CyberMan4 (Instance)
SkeletalMeshComponent (SkeletalMeshComponent0)
render
General
Actor
Animation
LOD
Misc
Physics
Rendering
Streaming
All
Owner No See
Only Owner See
Treat as Background for Occl...
Use as Occluder
Render CustomDepth Pass
Visible In Scene Capture Only
Hidden In Scene Capture
CustomDepth Stencil Value
0
Translucency Sort Priority
0