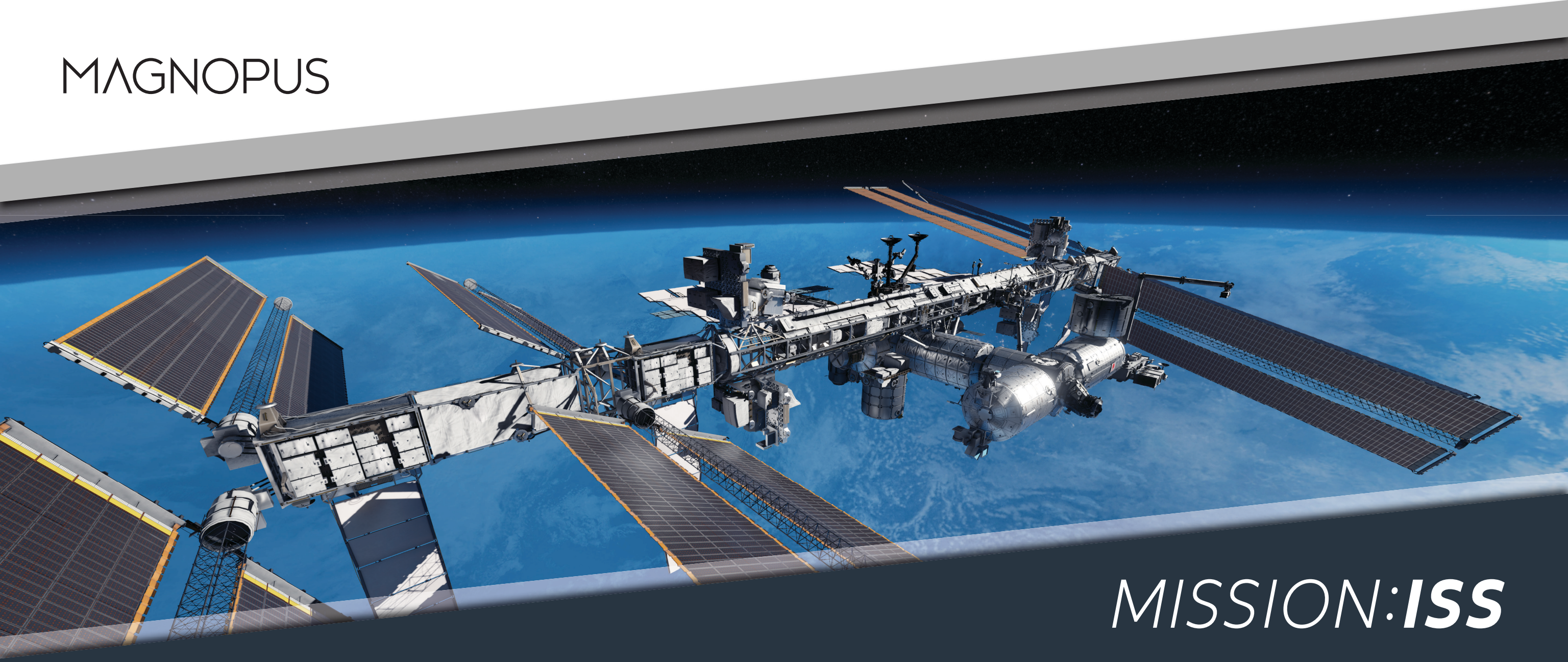


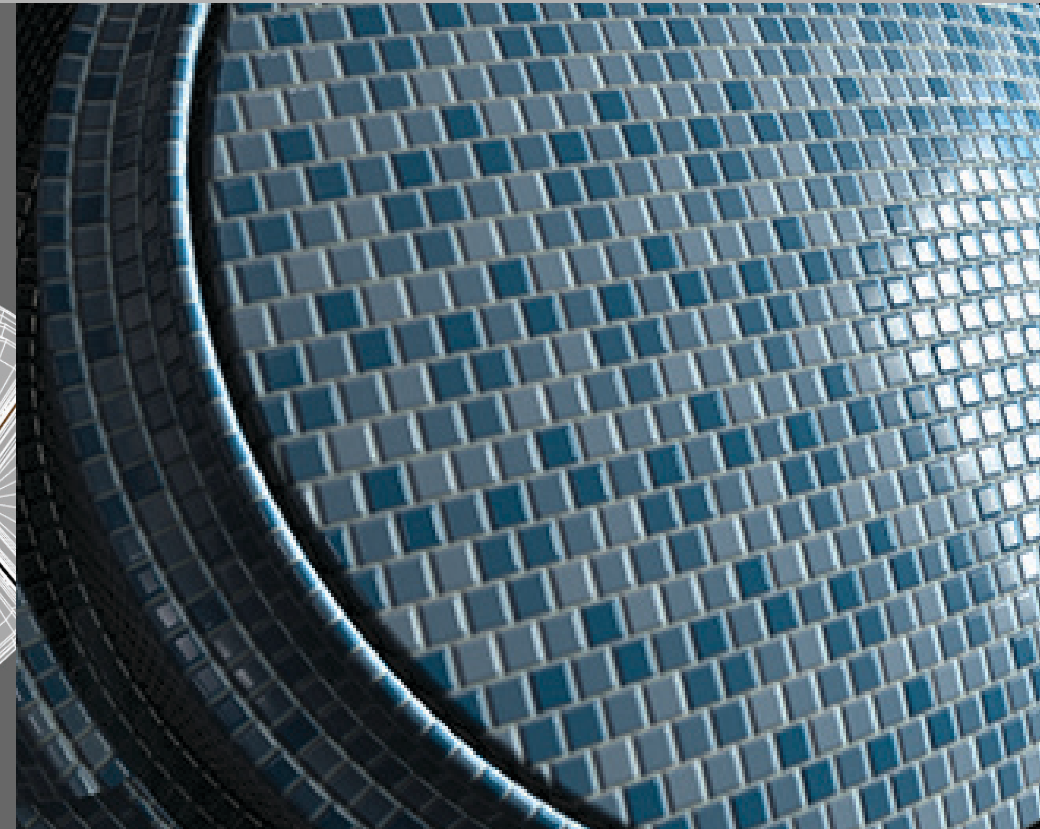
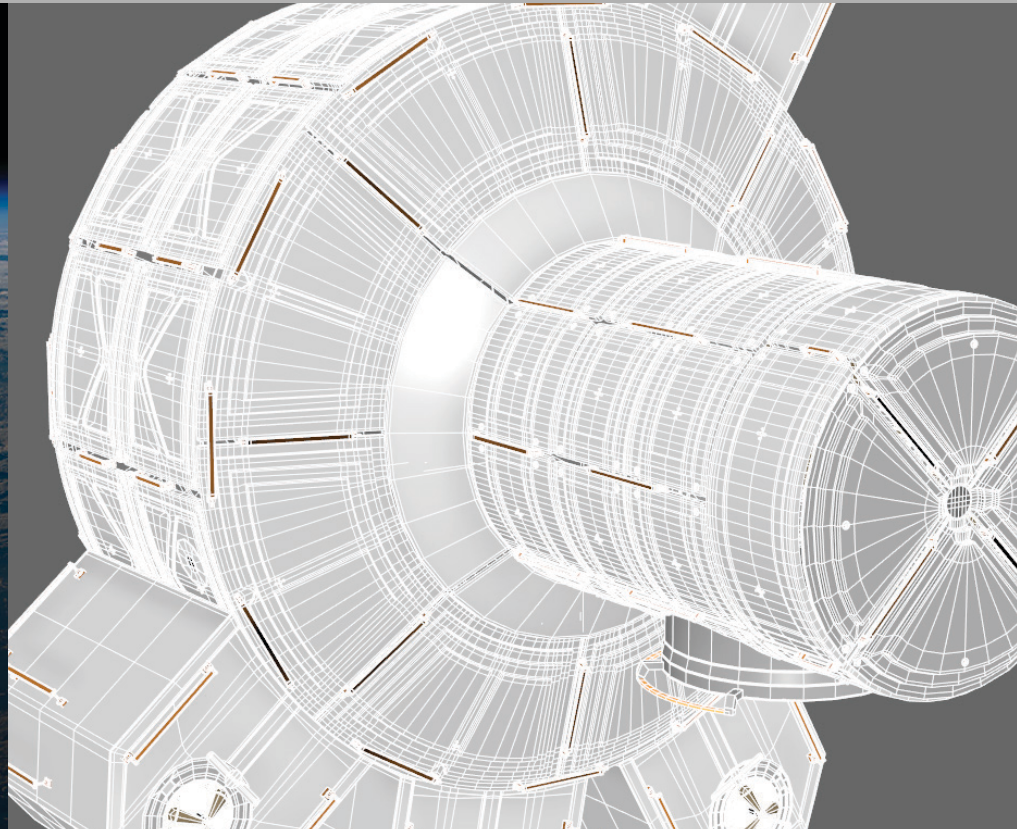
MAGNOPUS



# MISSION:ISS

THE EXTERIOR BREAKDOWN OF THE INTERNATIONAL SPACE STATION AND THE MISSION FOR QUALITY IN VR

EXTERIOR BREAKDOWN FOR VR



## PREAMBLE

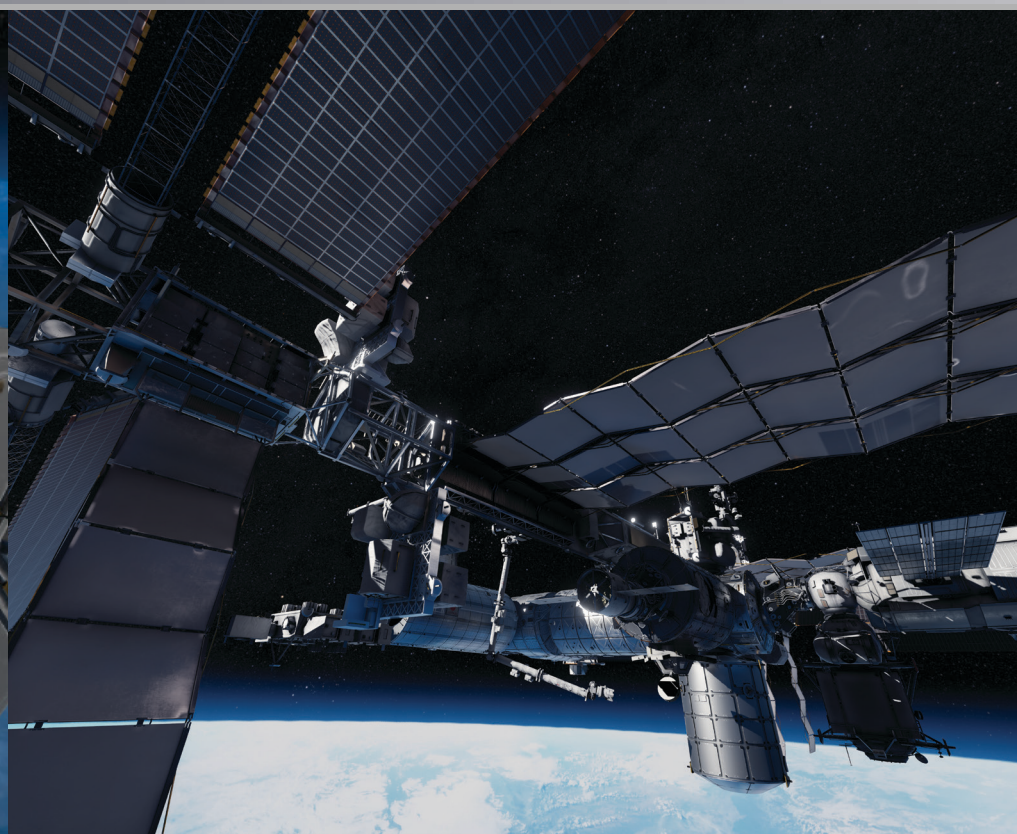
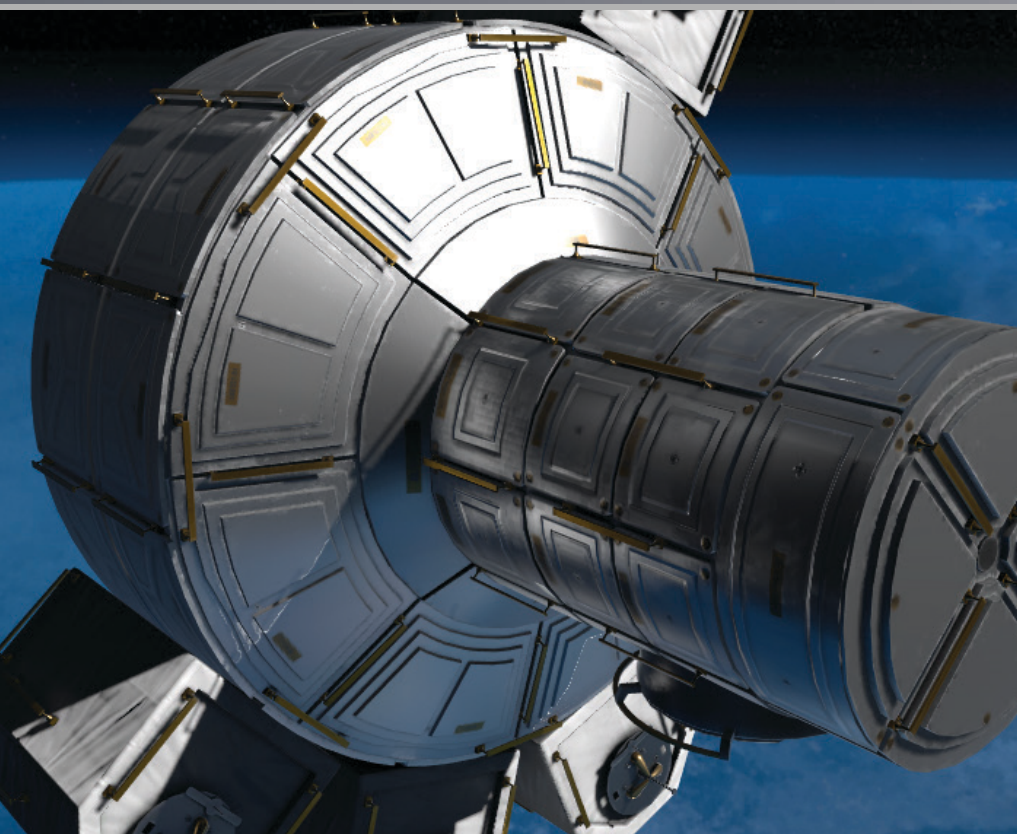
- INTRODUCTION 4
- OVERVIEW 5
- GOALS 5
- THINGS THAT DIDN'T WORK 6

## FIRST STEPS

- MODEL CLEANUP 9
- UV'S 10
- SUBSTANCE PREP 11

## TEXTURING

- BASE/SMART MATERIALS 12
- FINAL TEXTURING PROCESS 13



## UNITY

- INTEGRATION INTO UNITY 14
- MORE TEXTURES FOR LESS MEMORY 14
- MSO SHADER 15

## LIGHTING

- SETUP 16
- TEXTURE TWEAKING TO MATCH REFERENCE 20
- LIGHT BAKING 20
- POST-PROCESS 21

## CONCLUSION 22

# INTERNATIONAL SPACE STATION: FROM 3 HOURS PER FRAME TO 11 MILLISECONDS



## INTRODUCTION



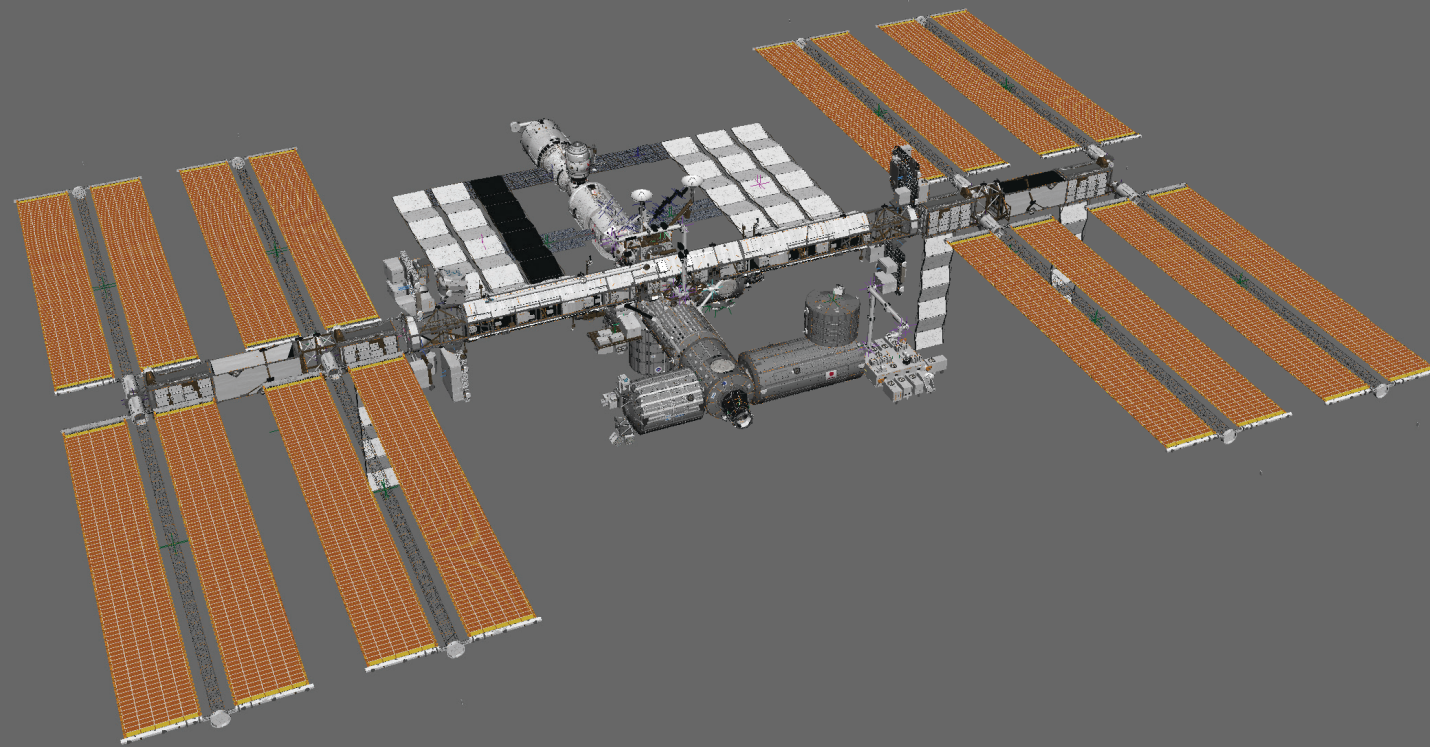
**3D ASSET LEAD**  
LUKE SCHLOEMER

Hello! My name is Luke Schloemer and I work as a 3D Asset Lead at Magnopus. I have a background in various indie game studios and specialize in Textures and Materials in Real-Time Game Engines. I was in charge of establishing the pipeline and creating the models/textures/materials for the Exterior of the International Space Station for Mission: ISS.



**CG SUPERVISOR**  
DEREK BLUME

Hi! My name is Derek Blume and I work as a CG Supervisor at Magnopus. I have worked as a 3D Generalist/ Environment Artist in the Visual Effects Industry. I set up the final look for the Exterior of the International Space Station in Mission: ISS.



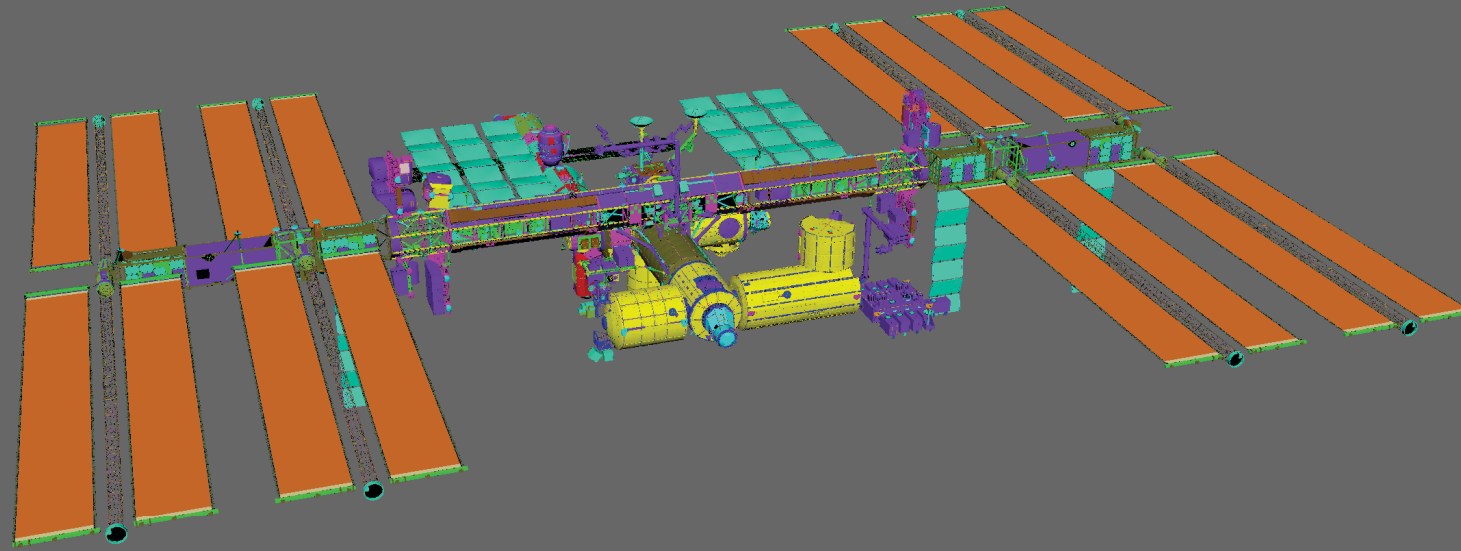
INITIAL ISS MODEL IN MAYA, 16M TRIANGLES TOTAL. 150 MATERIALS. 75 TEXTURES.

## OVERVIEW

We began with an extremely high-poly mesh from NASA of the International Space Station with the plans of bringing it into Virtual Reality. In fact, there are many great models available on the internet of this asset, but they tend to be extremely dense and modeled with SubD in mind. The one we used came with little to no textures and all basic materials. This meant aside from the base shape modeling, the entire station had to be completely rebuilt from scratch.

## GOALS

Have the highest quality International Space Station running in under 11 ms for VR. Obviously with something as complicated and as unique as the ISS, this would be a huge undertaking. We wanted to retain as much detail as possible but also keep it from feeling pristine and hot off the manufacturing line. Our target engine would be Unity as well so each mesh had to be kept under 65000 Triangles/ Vertices otherwise it would be broken apart using Unity's Mesh Importer. This tended to mess up Lightmaps and break smoothing groups. This also limited our shader production pipeline as well since any fancy material we might need has to be set up by an artist or a programmer who has time (which was not the case during 90% of production).

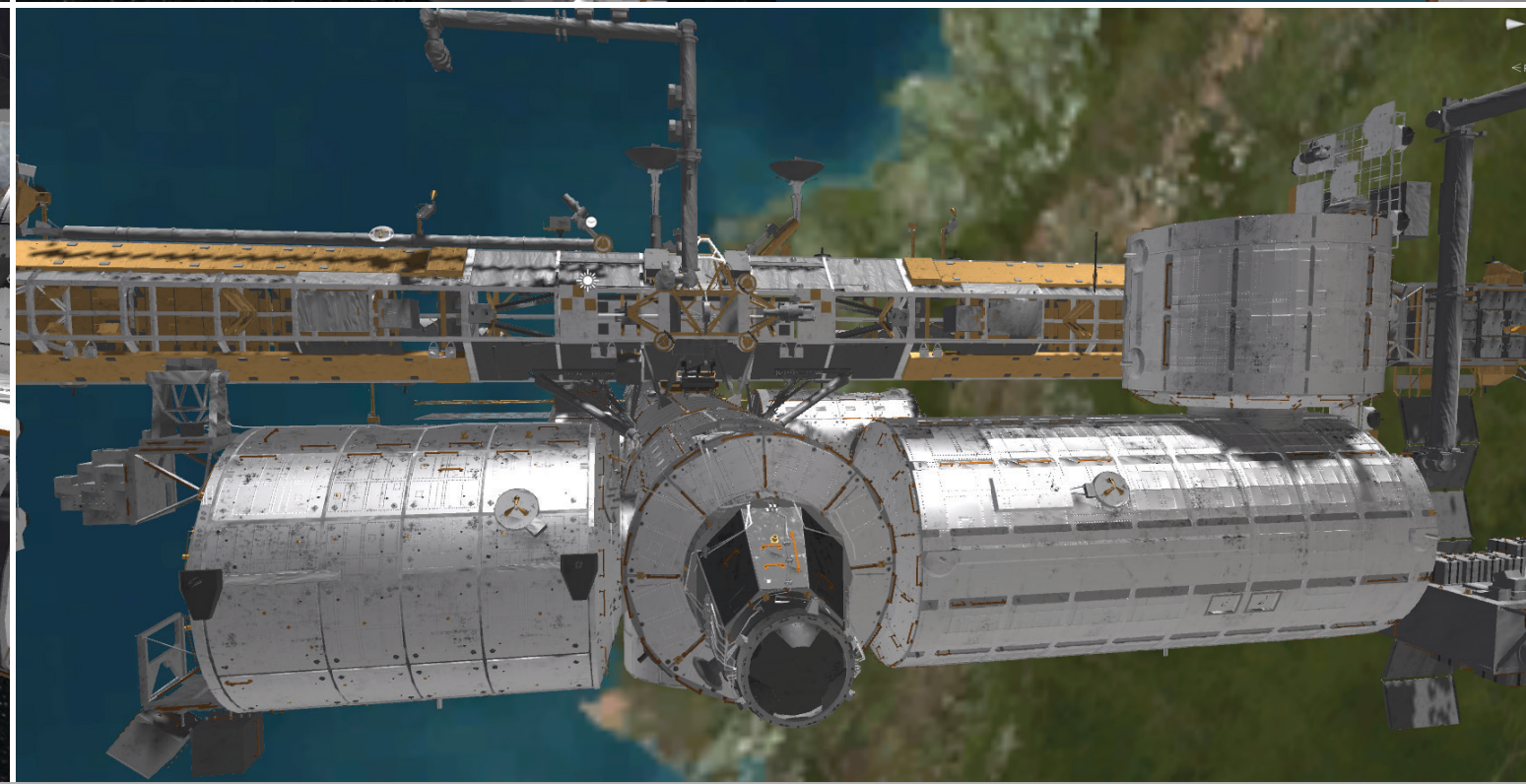
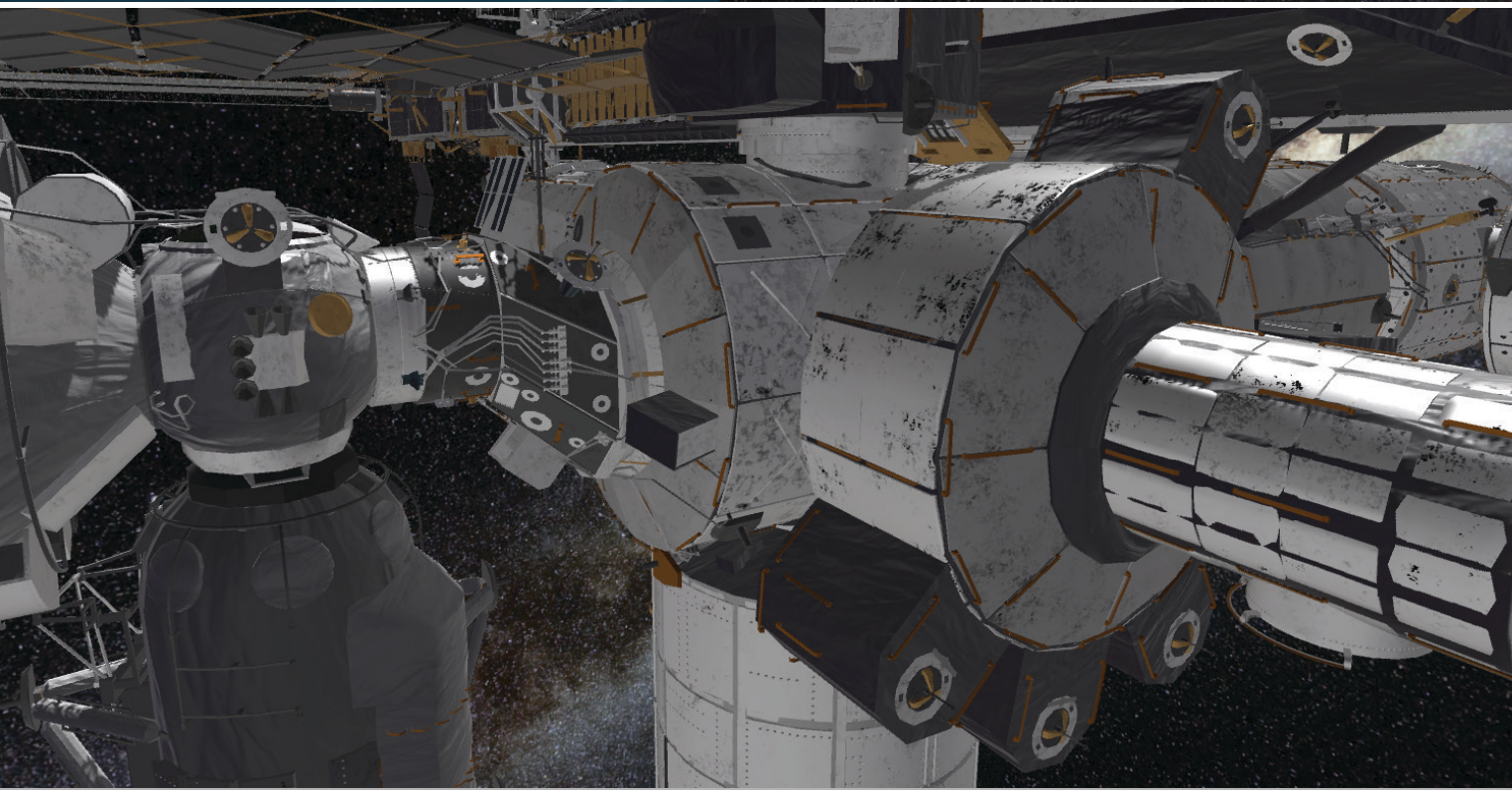
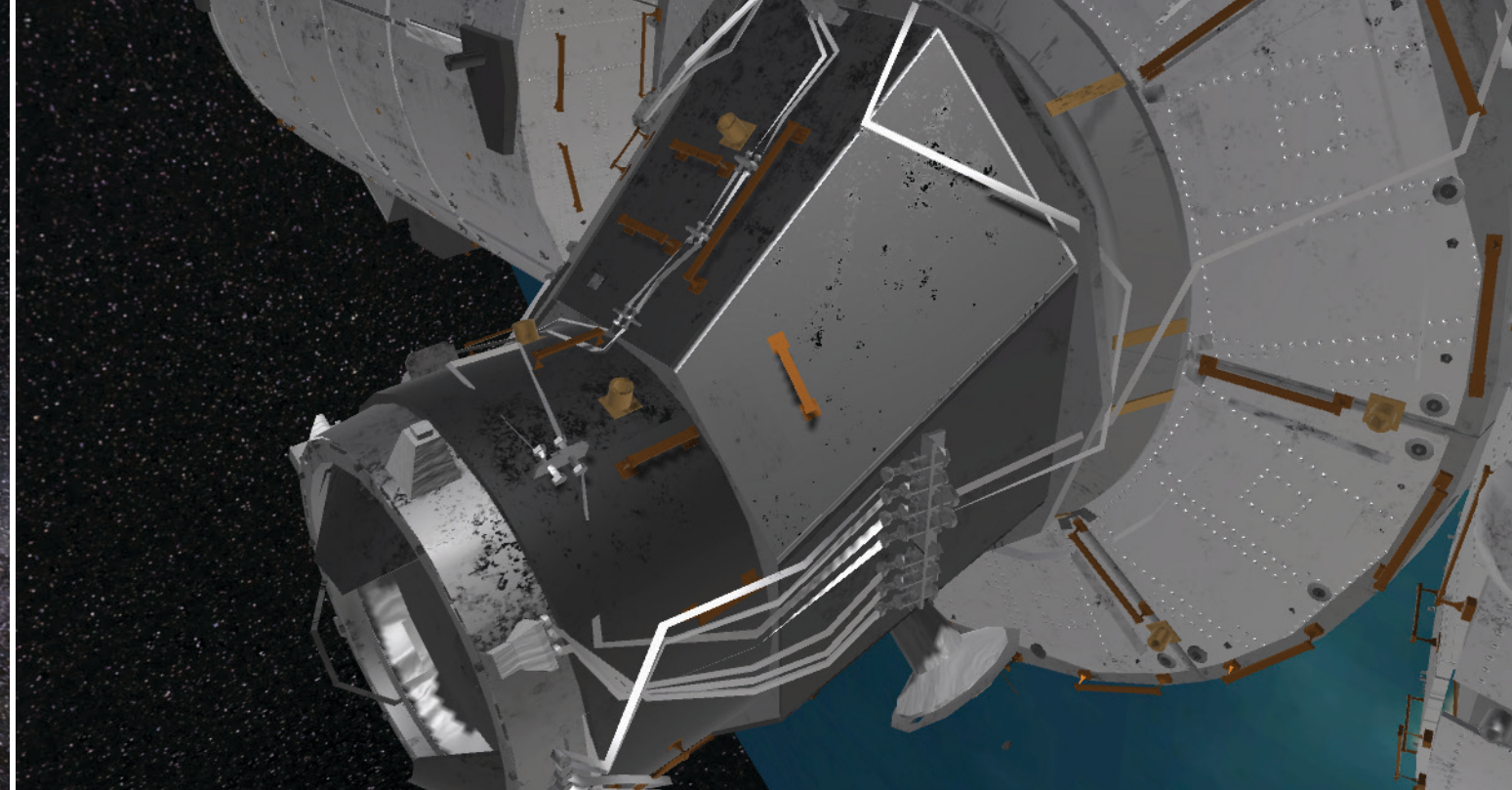
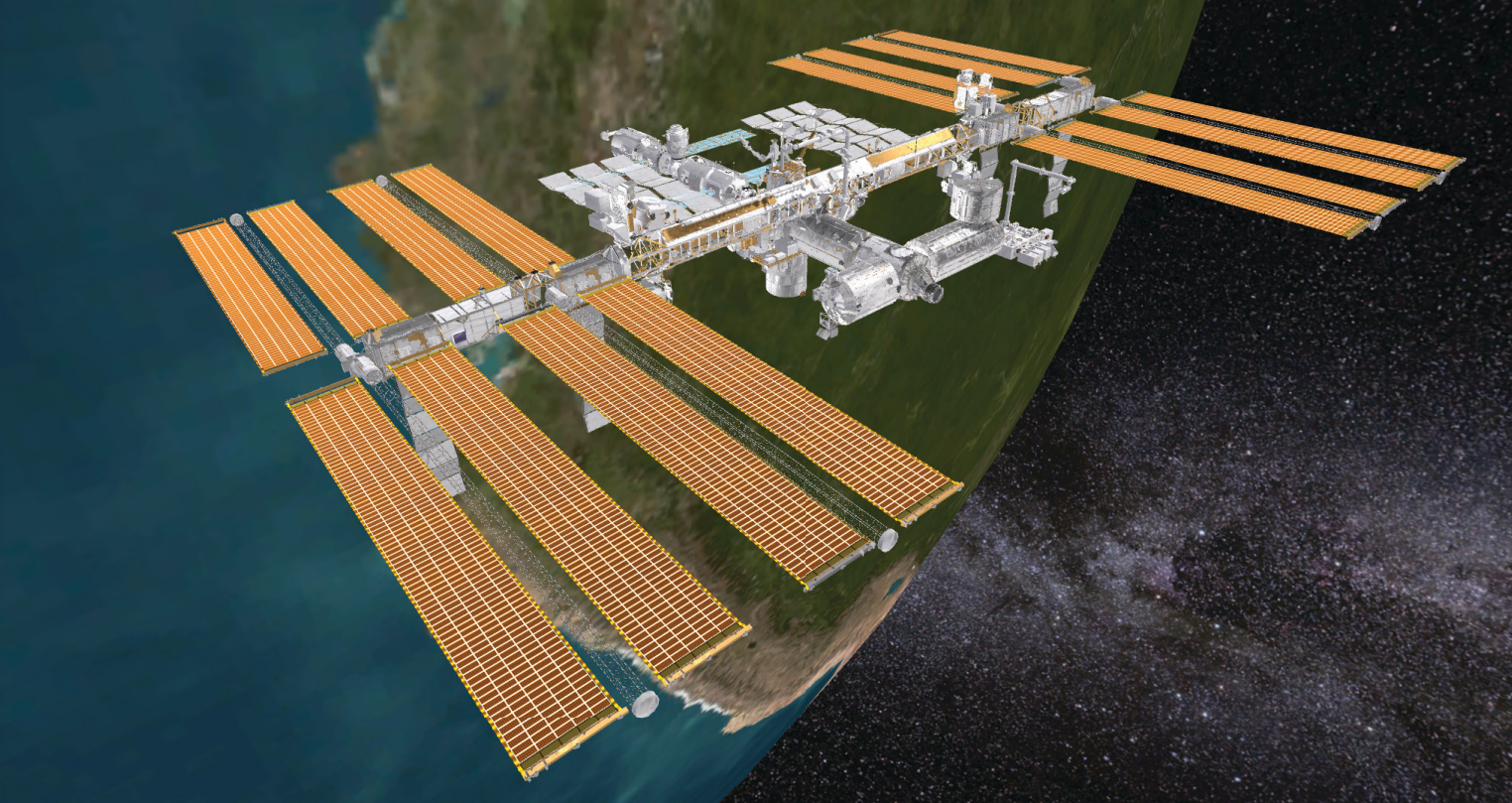


930K TRIANGLES, 25 MATERIALS, 14 TEXTURES. COLOR CODED TO DESIGNATE SHARED MATERIALS.

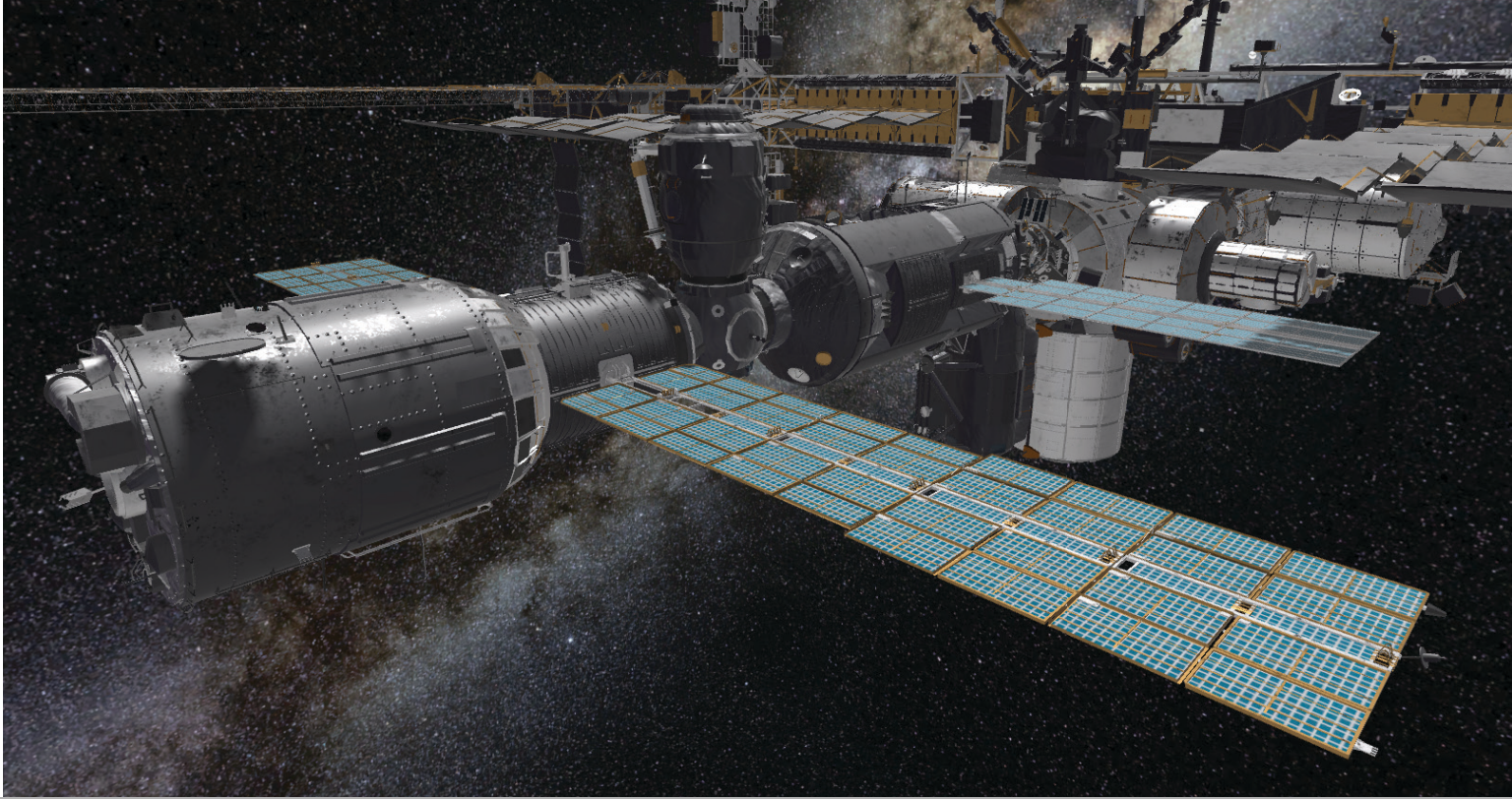
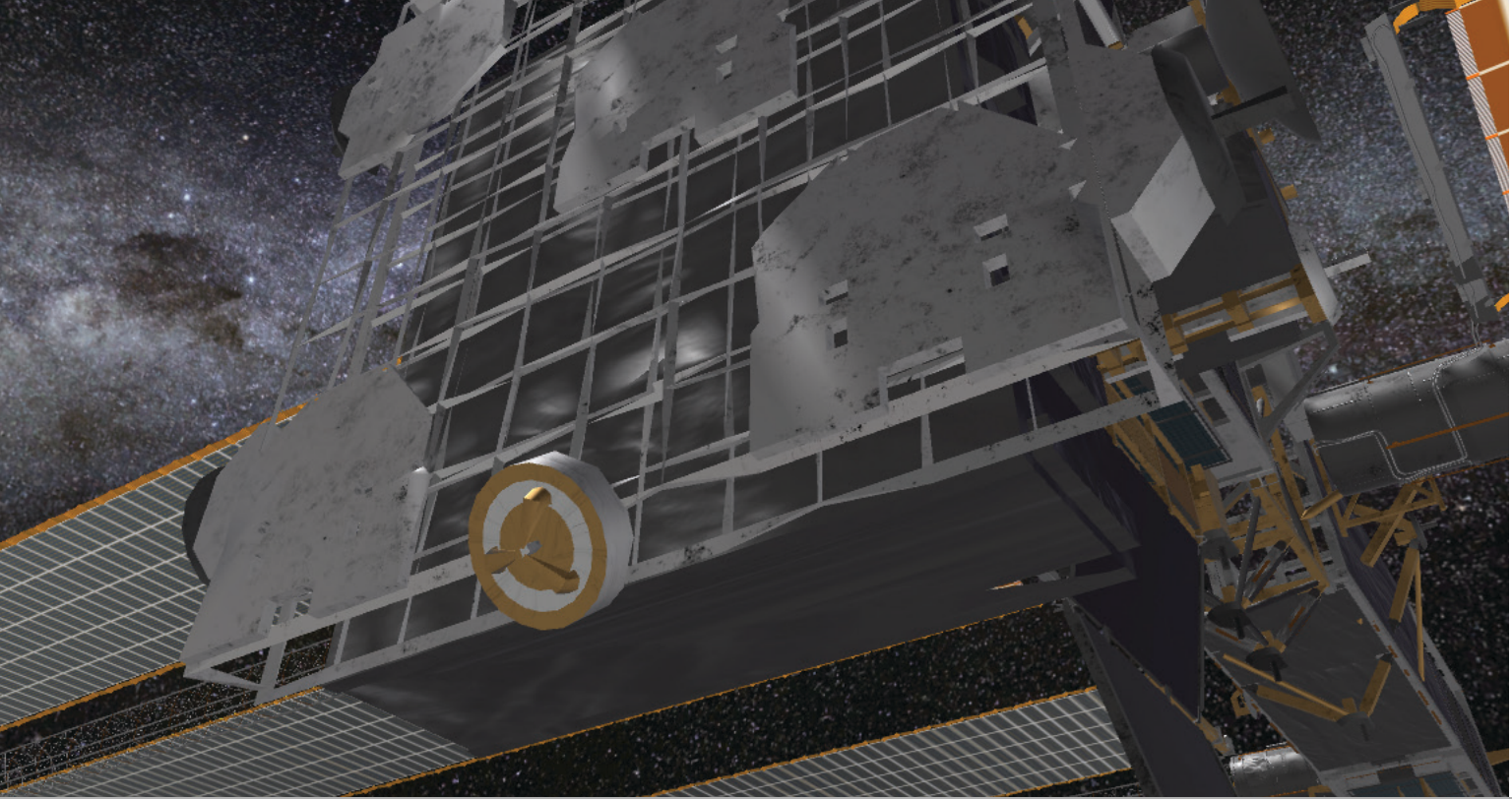
## THINGS WE TRIED THAT DIDN'T WORK OUT

It's all hard surface right? So all we need is the silhouette and tileable materials! The first iteration was going through each piece and using Maya's Polyreduce or Max's ProOptimizer to get it to a more manageable size. Then we combined and condensed as many materials as possible based on color and type. We created a few tileable base materials out of what came with the Turbosquid model. Exported into unity, set it all up, and did a quick and dirty light bake. The initial results were promising. We had a version of it running in VR. From far away it looked great. However, we wanted to move along the ISS up close and personal.

Unfortunately, it looked far less impressive upon closer inspection. The auto-reduction process has messed up smoothing, light baking errors, as well as pieces of geometry that naturally got completely screwed up with the process. It got us in VR sooner than we thought and was enough for our programmers to start working with it. It overall looked very unpolished and we were missing that "lived in" feel that we knew we had to accomplish. The level of quality was definitely not acceptable to ship with, so we started the task of doing every piece the right way with no automated functions (for the most part).



FIRST ITERATION IN UNITY

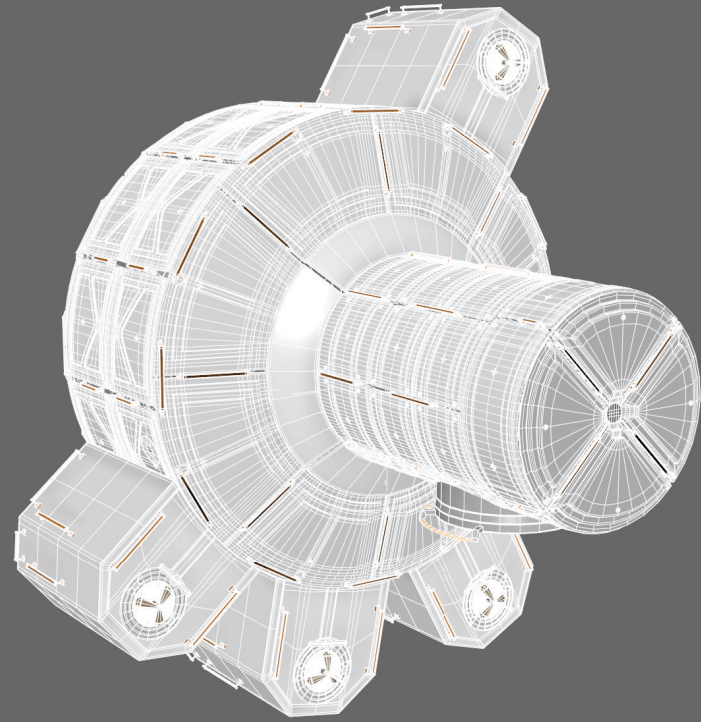


FIRST ITERATION IN UNITY

Obviously we could have taken this much further and cleaned up any issues we found, but it almost seemed like working in the wrong direction, especially since we had such a strong clean model to begin with.



## FIRST STEPS



QUEST AIRLOCK SOURCE ASSET. 450K TRIANGLES. 15 MATERIALS. 5 TEXTURES.



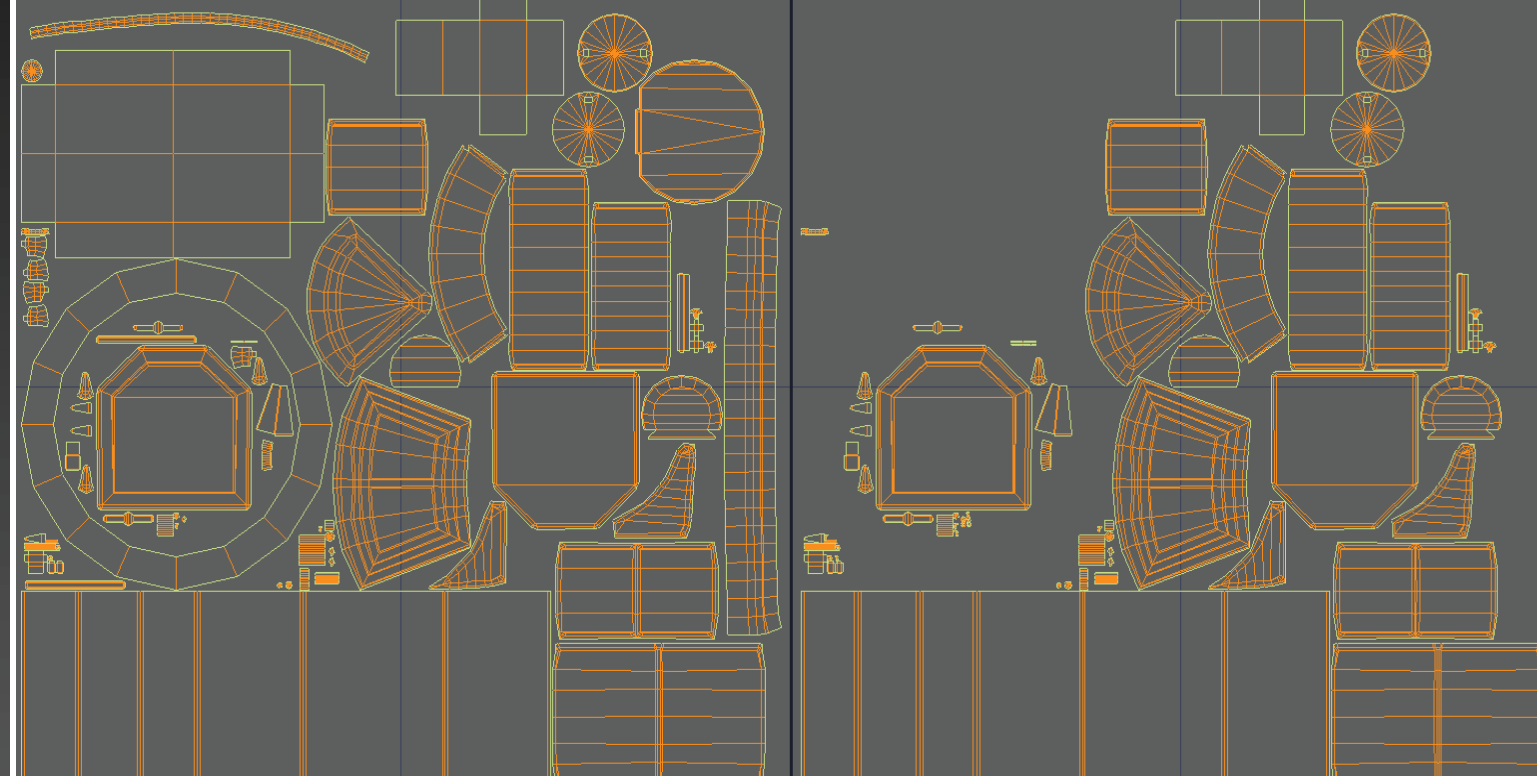
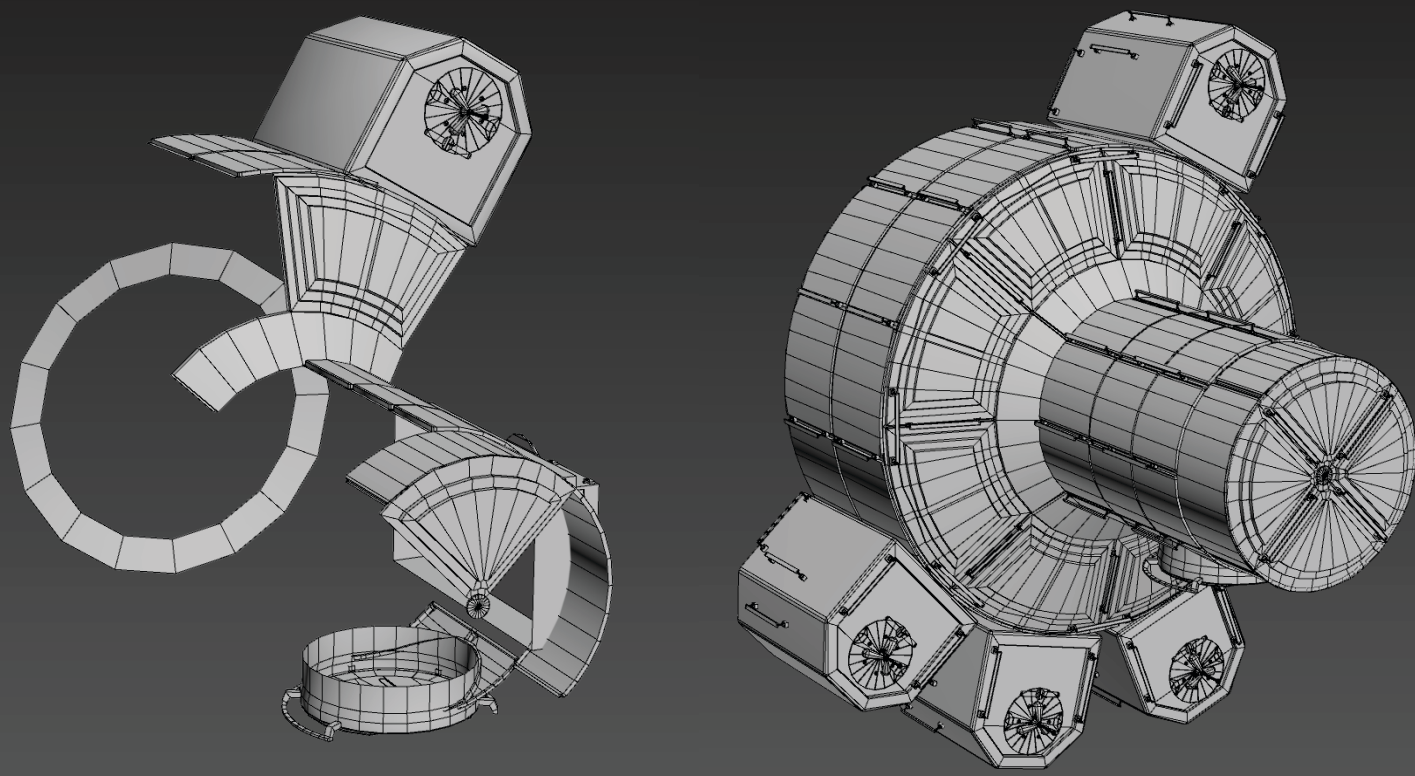
ISOLATED REPEATABLE PIECES OF THE QUESTAIRLOCK

## MODEL CLEANUP

Since the scope of the task was so huge, naturally the first step I wanted to do was develop a look for a single polished asset, as well as gauge how long it takes to clean up a module properly. We had a decision to make. Should we just focus on cleaning up all the loose ends of the current version? Or should we start over from scratch? Since the ISS was usable enough to prototype with, I set out to establish the potential workflow for creating the ISS 2.0. The biggest benefit we had for these assets is that they had several repeatable pieces. We could theoretically get a very polished “hero asset” look with only a few pieces unwrapped and baked in a clever manner.

To make my life easier, the first step I did was isolate all the repeatable pieces of geometry on the module. The wonderful thing about all these modules is pretty much all of the pieces are symmetrical in some form. Through clever placing of pivots, I could quickly repopulate the assets around the module to make the full asset once the cleanup phase was finished. I wanted to preserve the silhouette as much as I could, but obviously make the pieces as low as possible. Each single vertex needed to be used as efficiently as possible on every single asset. It was an extremely methodical process.

The model cleanup was relatively straightforward overall. Any cloth element generally preserved the beveled edges. Metal + Plastic pieces were using hard edges and are rarely beveled.



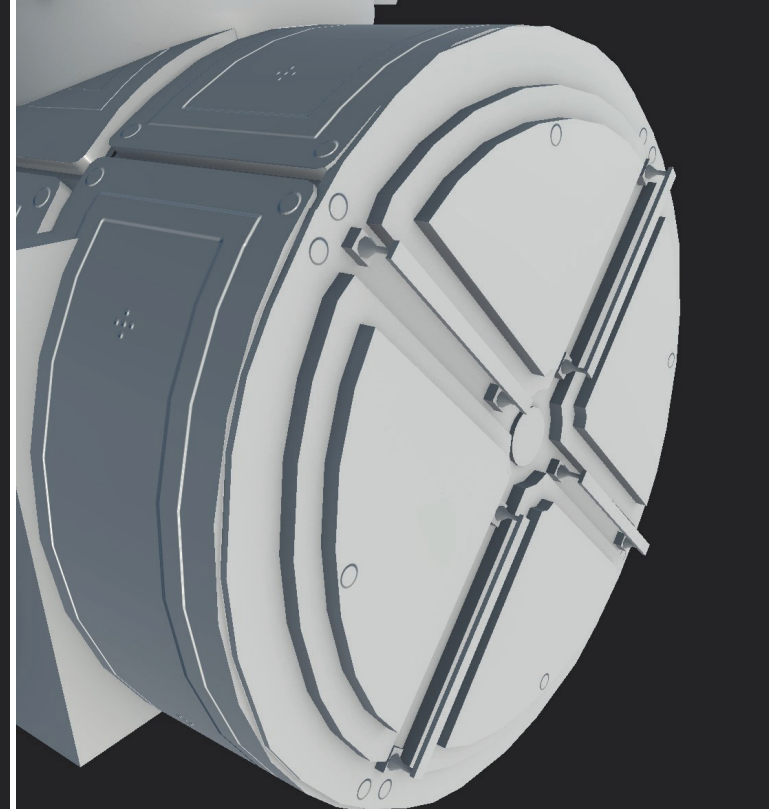
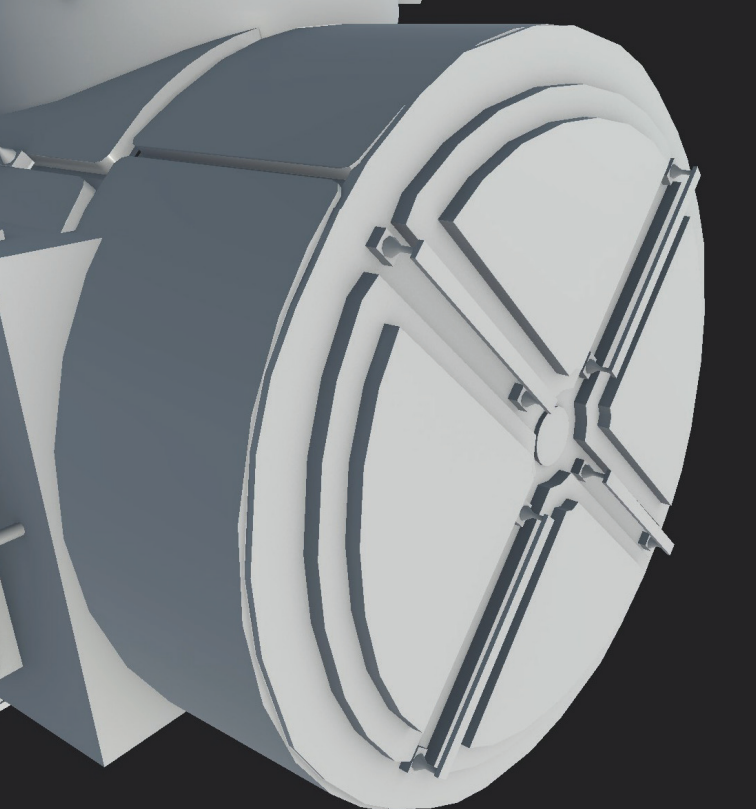
CLEANED UP GEOMETRY OF REPEATABLE PIECES. TOTAL OF 3.5K TRIANGLES. FINAL GEOMETRY WITH OVERLAPPING UV'S. 28K TRIANGLES TOTAL.

FINAL UV UNWRAP

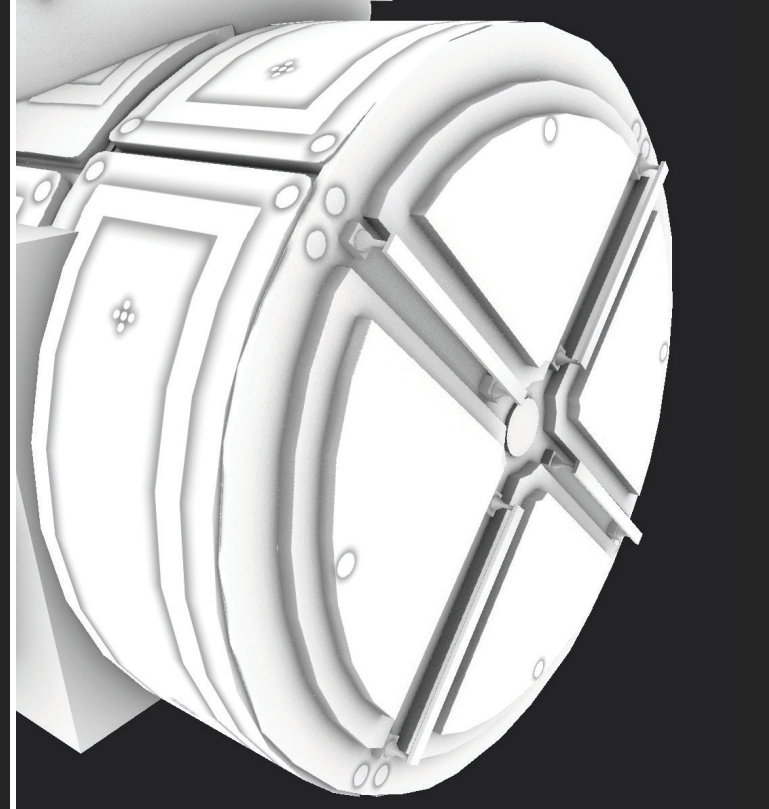
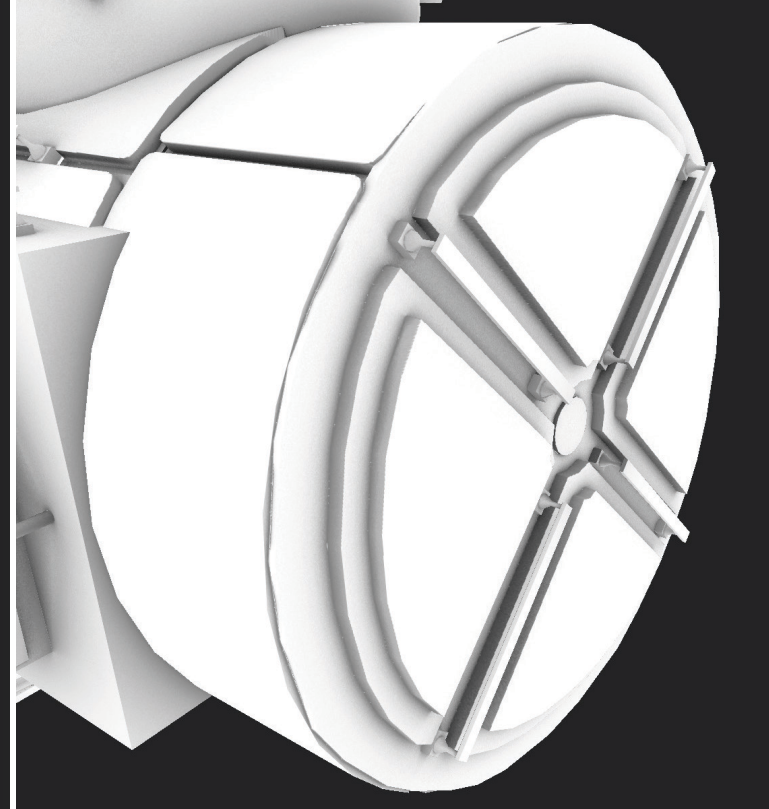
## UV'S

Once the geometry was cleaned up; I started unwrapping each piece individually. I tried to sew as many UV edges together as possible to maximize my texel density. I wanted to strike a balance between realistic Ambient Occlusion bakes, and overlapping pieces.

For my final unwrap, I unwrapped each piece individually. Then I packed the pieces together in a single map. After that I copied pieces over to finish the asset. In order to get a proper Ambient Occlusion bake across the board, I needed to put all but one of the overlapping UV shells into the 1-2 UV space.



BEFORE AND AFTER OF NORMAL PAINTING IN SUBSTANCE.



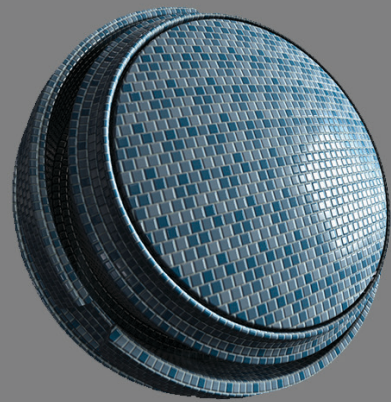
BEFORE AND AFTER AMBIENT OCCLUSION WITH NEW HEIGHT/NORMAL DATA

## SUBSTANCE PREP

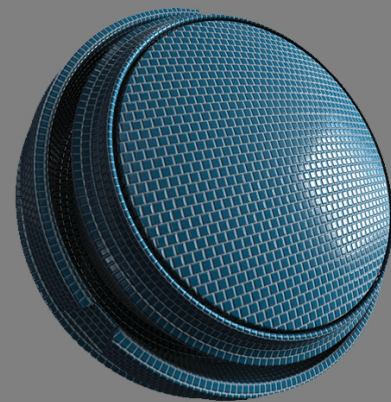
To save time on the baking process, I opted to paint my normals in Substance instead of relying on the usual high poly to low poly projection workflow. I baked my base maps in Substance and made sure that there were no Ambient Occlusion errors from my overlapping UV shells.

Once I was happy with my new normal map, I used this plugin (<https://share.allegorithmic.com/libraries/1994>) to composite my newly painted height data into the Ambient Occlusion and Curvature Maps.

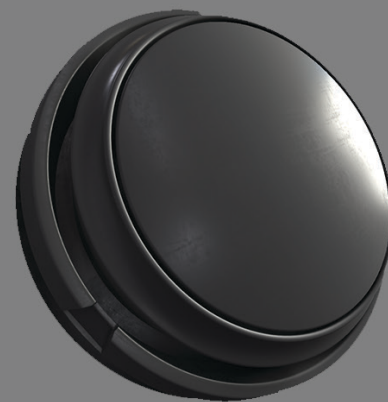
# TEXTURING



Solar Panel Varied



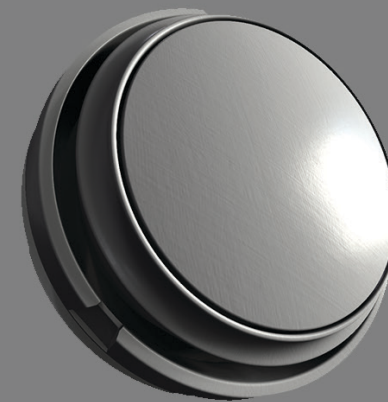
Solar Panel



Dirty Plastic



Fabric



Plastic

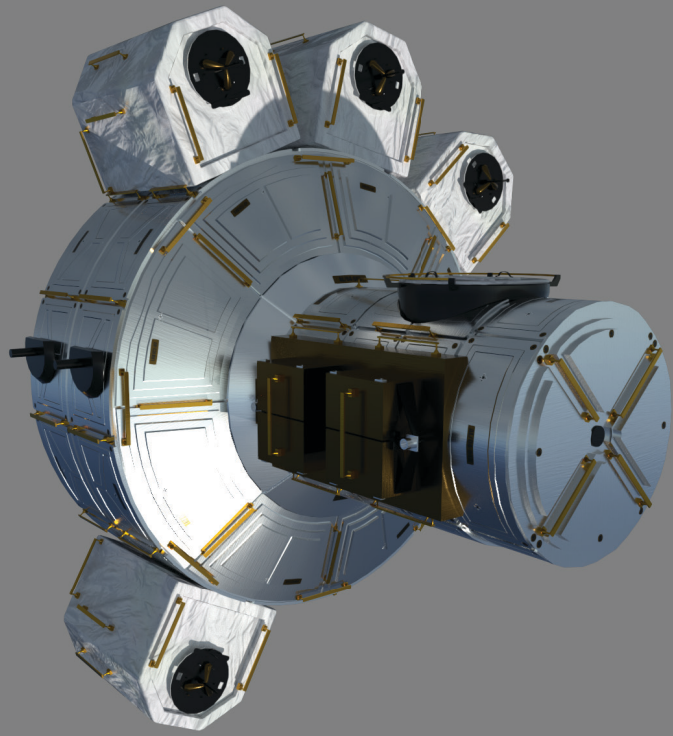


Aluminum

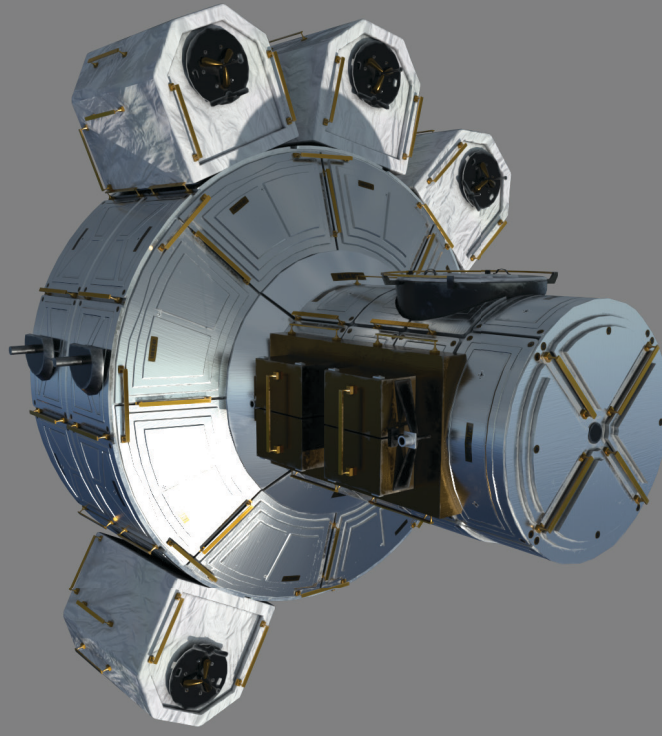
BASE MATERIALS CREATED IN SUBSTANCE

## BASE MATERIALS

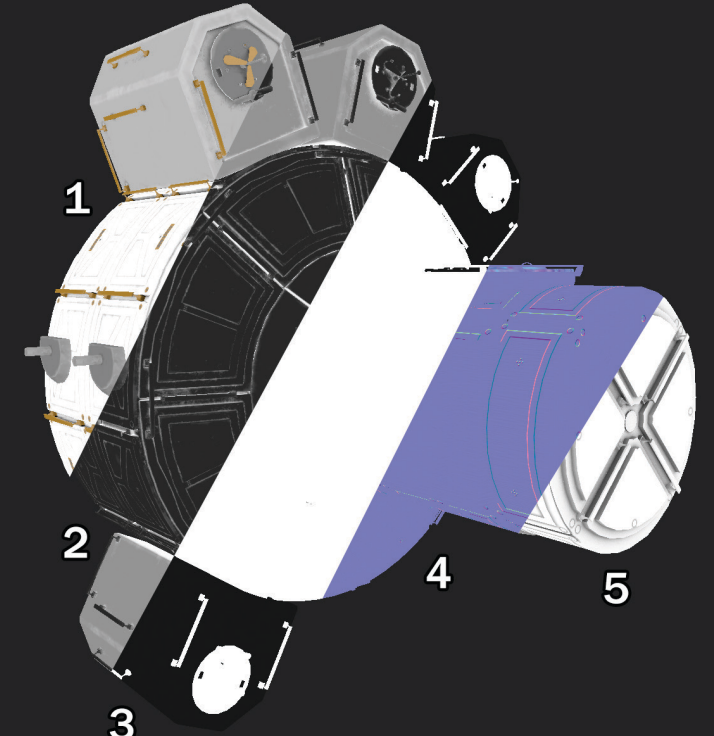
To ensure consistency across all assets, I set about creating some base materials in Substance. I was mostly concerned about how the materials reacted to light and less concerned about the basic colors. After analyzing the space station materials, there were lots of different colored metals, solar panels, and pseudo-plastics. Once I had these base materials, I can populate any asset and ensure every asset's material behaves the same way when it came to lighting.



BASE MATERIAL PASS



ISS\_GRUNGE SMART MATERIAL APPLIED



DIFFERENT FINAL TEXTURES; 1 = COLOR, 2 = ROUGHNESS, 3 = METALLIC, 4 = NORMAL, 5 = AMBIENT OCCLUSION

## FINAL TEXTURING PROCESS

The first step once I had all my materials and bakes set up was to assign it to each appropriate element. I tried to match the reference model and pictures as closely as possible regarding base color and type of material. This step was relatively quick and painless for the most part.

Now that I had all the base materials set up correctly, the next step was achieving the “used” look. It had to be subtle enough to not make it look like it’s an incredibly worn ISS, but still unique enough to give it the lived- in look. To do this I set up a ISS\_Grunge Smart Material that applied a dirt pass and scratch pass across the entire mesh. The dirt pass primarily used the Ambient Occlusion map to generate the mask; while the scratch pass used the curvature to make the metal edges look a bit more worn.

# UNITY

## INTEGRATION INTO UNITY

Each module was exported in world space that way all we had to do was zero out the transforms and everything snaps into place. Before importing into Unity I created a custom lightmap set just packing the current UV's into the second channel; this removed all overlapping UV's but kept the same UV shells. Some meshes were over the 65K triangle/vert limit, so I had to split up the asset ahead of time to prevent asset corruption from Unity's auto-splitting. Initially, there were issues with gray scale metalness and the final baked lighting. It caused colors to be washed out and the specular lighting to look incorrect. This seemed to be an artifact of Unity's Standard Shader combined with our lighting setup; still not 100% sure what caused this issue. To prevent this artifact we simply got rid of all the metalness on the grunge layers. The impact to the look we were going for was minimal and removed the issue outright.

## MORE TEXTURES FOR LESS MEMORY!

One thing we quickly started to notice was Unity's Standard Shader is not well optimized for texture memory. This part will get a little technical but should make sense for anyone who has messed with Photoshop and the "channel" function. First let me talk about what comes out of the box with Unity's Standard Shading Model.

Let me illustrate the problem by talking about what a full PBR setup in our pipeline needed. We have a Base Color, Metallic, Smoothness, Normal, and Occlusion Maps. Unity packages the Metallic and Smoothness together by putting the Smoothness into the Alpha of the Metallic map. So by default at 4k resolution a single RGB texture takes up 10.7MB of texture memory. A single texture with RGBA



FINAL LIGHT-BAKED ASSET IN UNITY

costs 21.3MB of texture memory. Normal Maps are also compressed differently to preserve the red and green channel's details. This causes a Normal Map's memory footprint to be the same as in image with alpha, 21.3MB. Using Unity's Standard setup, that gave us a memory footprint as listed below:

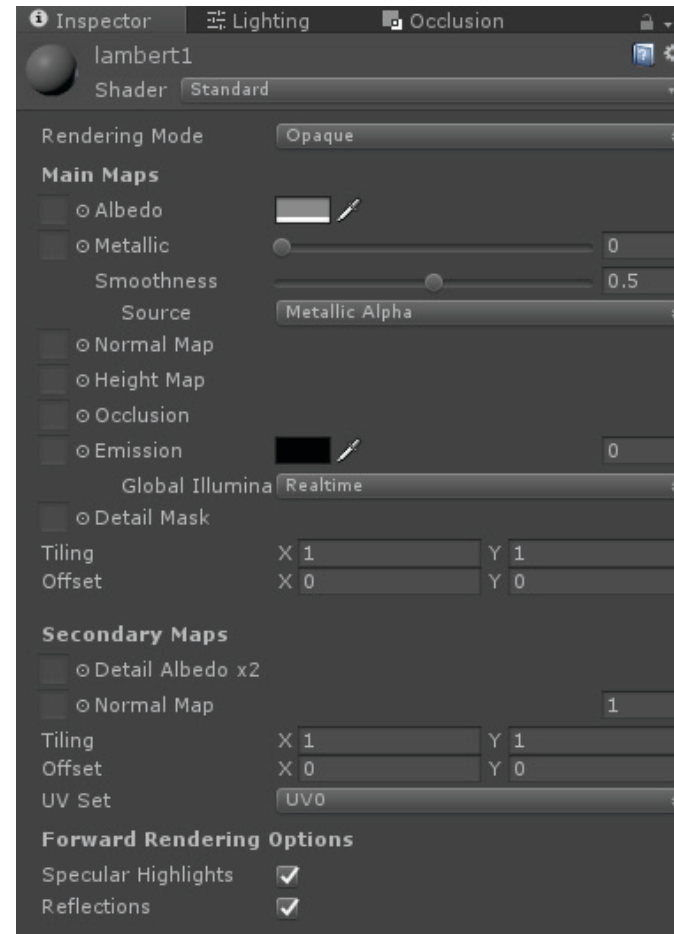
4k Base Color - RGB - 10.7MB                      4k Occlusion - RGB - 10.7MB  
 4k Metallic/Smoothness RGBA - 21.3MB      Total Texture Memory @ 4k: 64MB  
 4k Normal Map - RGBnm - 21.3MB

## MSO SHADER

For those familiar with Unreal; you can easily solve this problem by packing greyscale maps together into a single RGB images. Thus, the first idea of an MSO shader was born. MSO is an acronym for Metallic, Smoothness, and Occlusion. This would pack all three maps into a single RGB compressed image. The first iteration was made using the Unity plugin "Shaderforge" but at the time of production, baked "Directional Specular" lighting was unsupported by the plugin. This rendered the concept of what we had done usable, but implementation would need someone more tech-heavy to custom write a version of Unity's Standard Shader. Luckily for us, we had someone with just enough time (1 day) to make this version of the shader. Implementing this shader reduced our memory footprint to the following textures:

4k Base Color - RGB - 10.7MB                      4k Normal Map - RGBnm - 21.3MB  
 4k Metallic/Smoothness/Occlusion              Total Texture Memory @ 4k: 42.7MB  
 RGB - 10.7MB

Implementing this shader saved us 33% across the board on texture memory with NO performance hits and no changes to our visual look



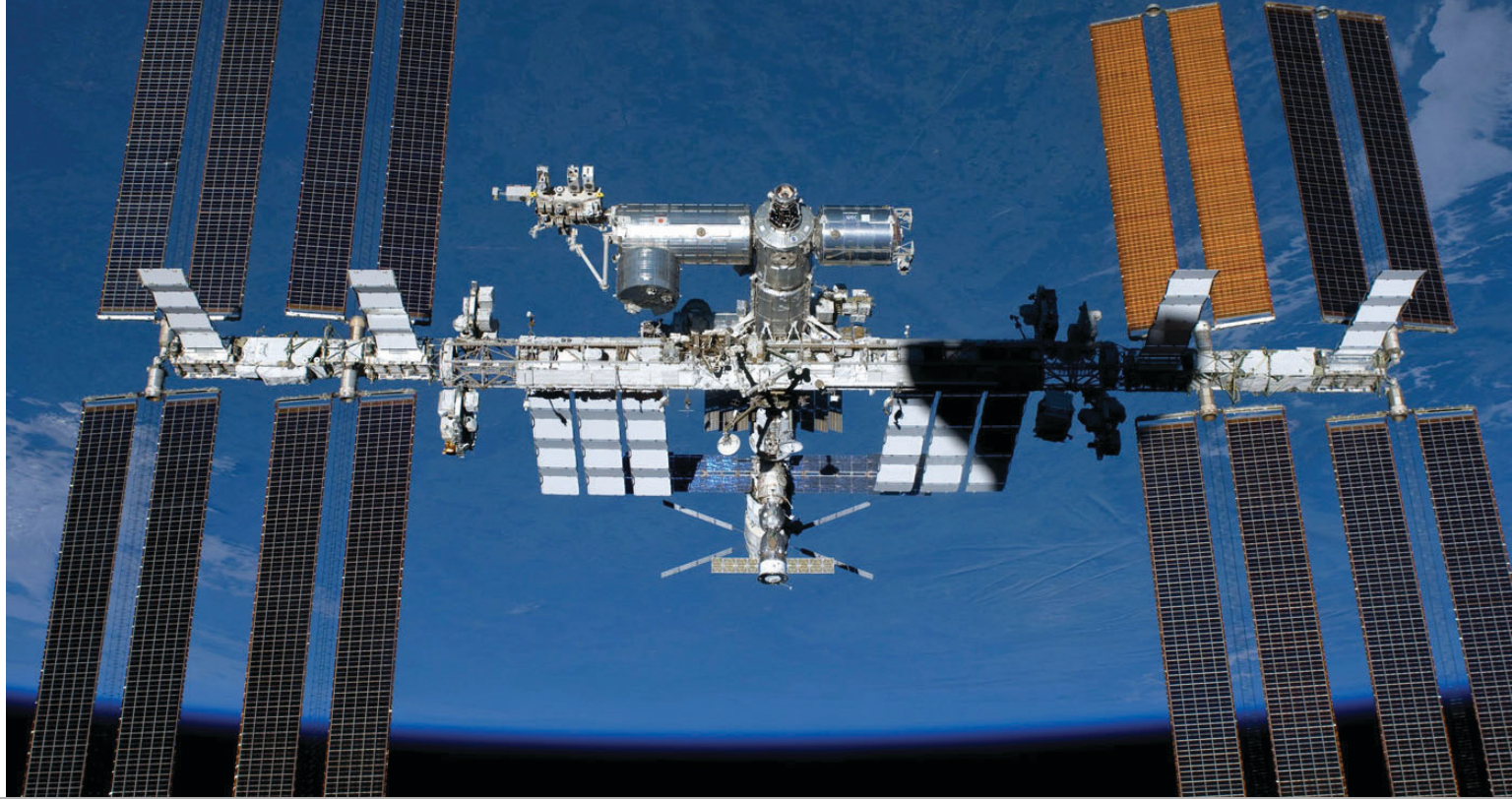
DEFAULT STANDARD SHADER IN UNITY



FINAL SHADER PARAMETERS OF MSO MATERIAL

as well. Near the end of production, we were hitting a cap on texture memory for the International Space Station interior. One of our developers wrote an editor tool script to process ALL the interior materials and assign the correct shaders. I created a Substance Designer batch file that processed all the textures and put them in the right place. This again saved us 33% memory across the board on the entire interior of the space station. It also saved us time packing our textures this way when we started compressing all the textures at the end of the project. All the specular properties of the material could be compressed at once since they were all part of the same texture. In the end, it was great having 4K source textures in our project and then just clamping them down to the minimum required. That ended up shaving about 500MB of Texture Memory at the end.

# LIGHTING



## LIGHTING SETUP

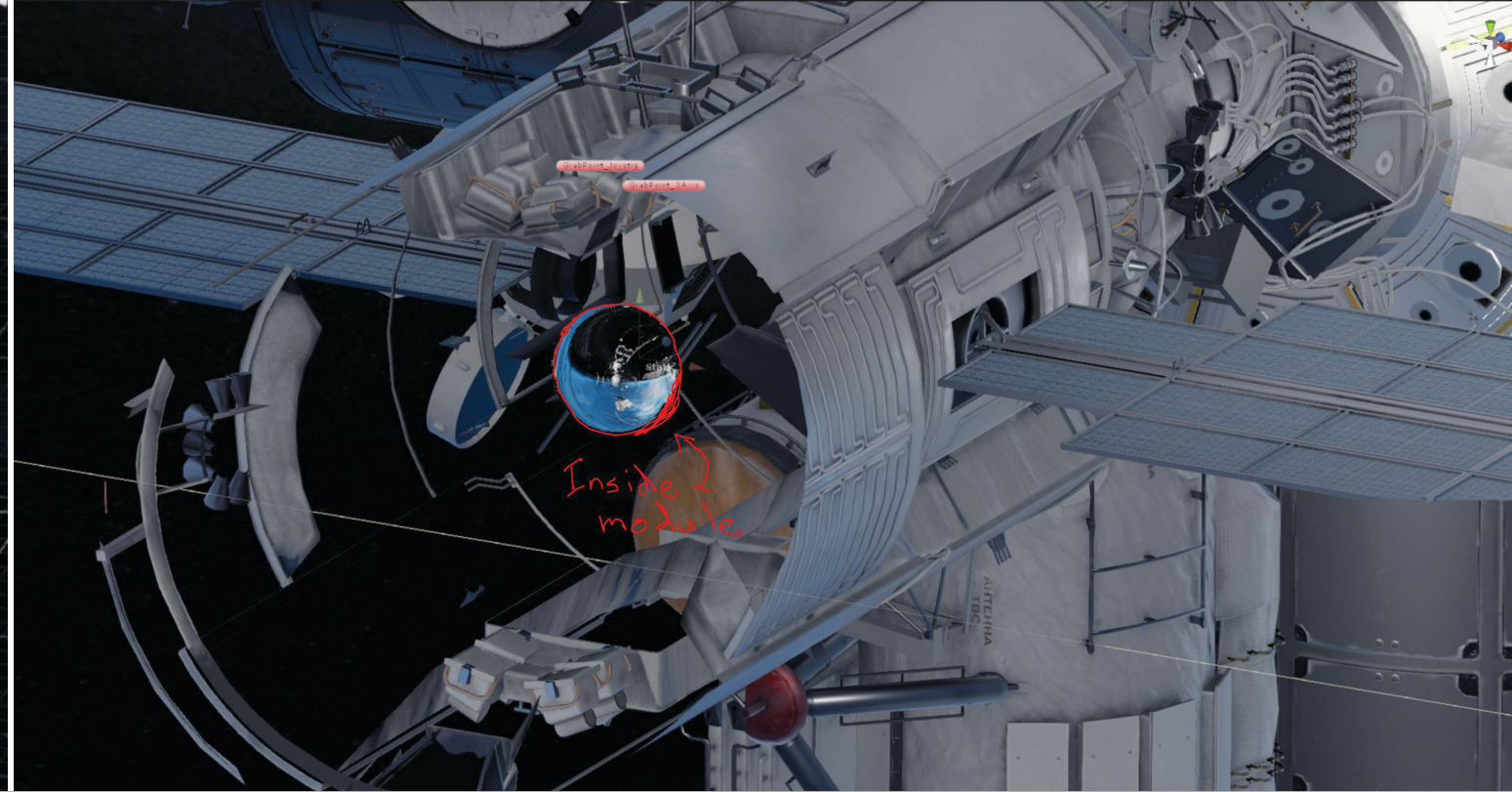
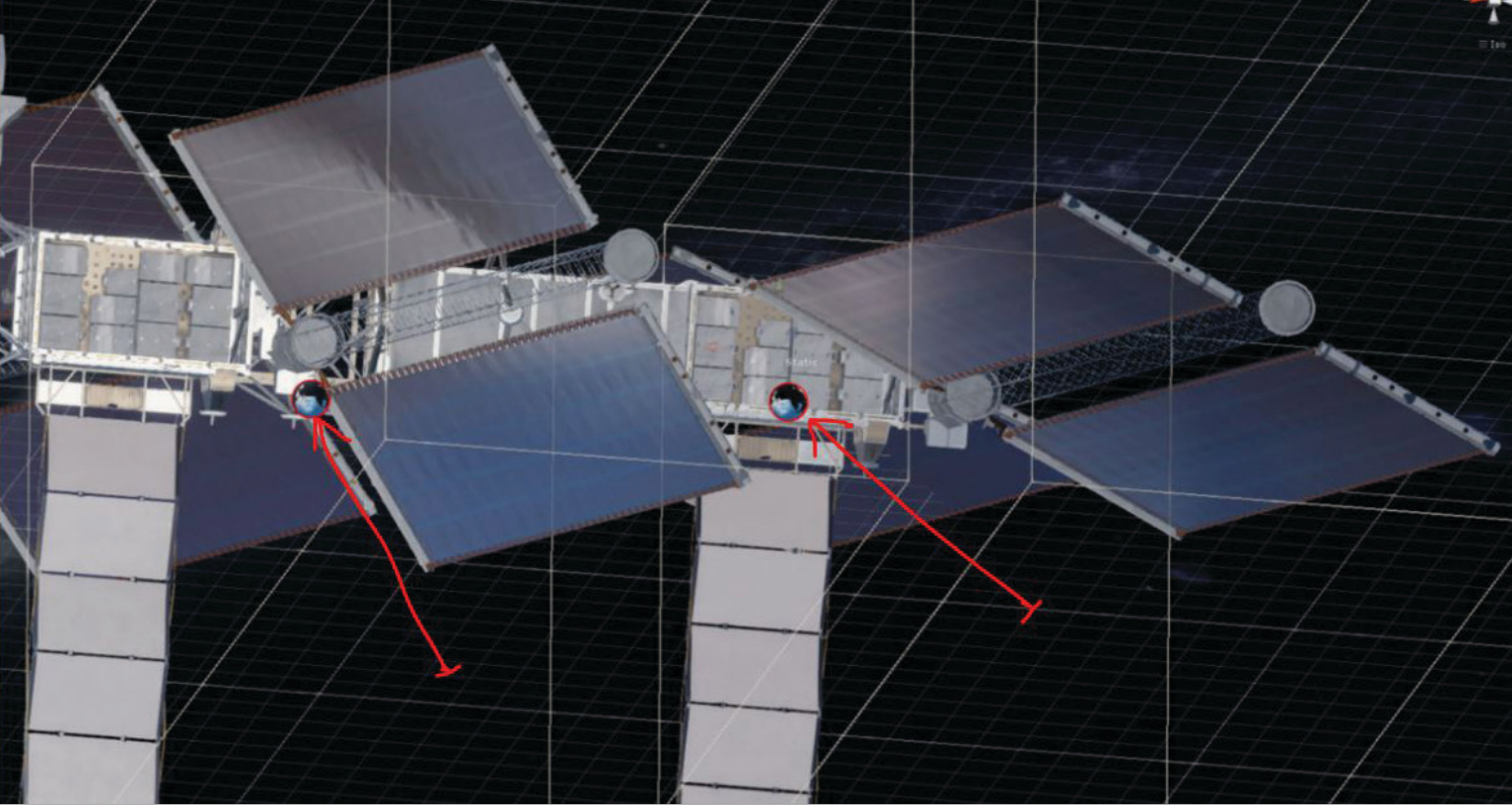
The awesome new challenge that lighting for VR offers is to design lights that will be interesting to the player from any angle available. In this case, all angles are available to the player as an astronaut floating wherever they want. This rules out the 3/4 back lit approach that works so well in the framed shots of a film. Our favorite reference images for the ISS always had extreme light angles from the sun raking across portions of the station. The finer shadows helped to define the smaller nurnies and greebles while the larger shadows kept the station from feeling flat.

We tried to mimic the feel of these images and find a location for the sun that produced interesting, angular shadows that are visible across portions of the station no matter what angle the player viewed it from. Many of the reference images felt like 1 point light setups with a super strong key light and intense shadows.





We wanted keep those extreme light values but never push it so far as to lose detail in the station. The reflection of the earth helped to reduce the blackness of the shadows but we still struggled with keeping detail in the less reflective parts. The earth ended up being used not just for reflections but by adding a directional light to create a subtle blue fill. Our final setup ended up with a 3 point system but kept the fill light and back light subtle enough so it added dimensionality up close but did not overwhelm the 1 point dramatic feel from a distance. We adjusted the light values so the user could see detail in the brightest whites and darkest blacks.



## REFLECTION SETUP

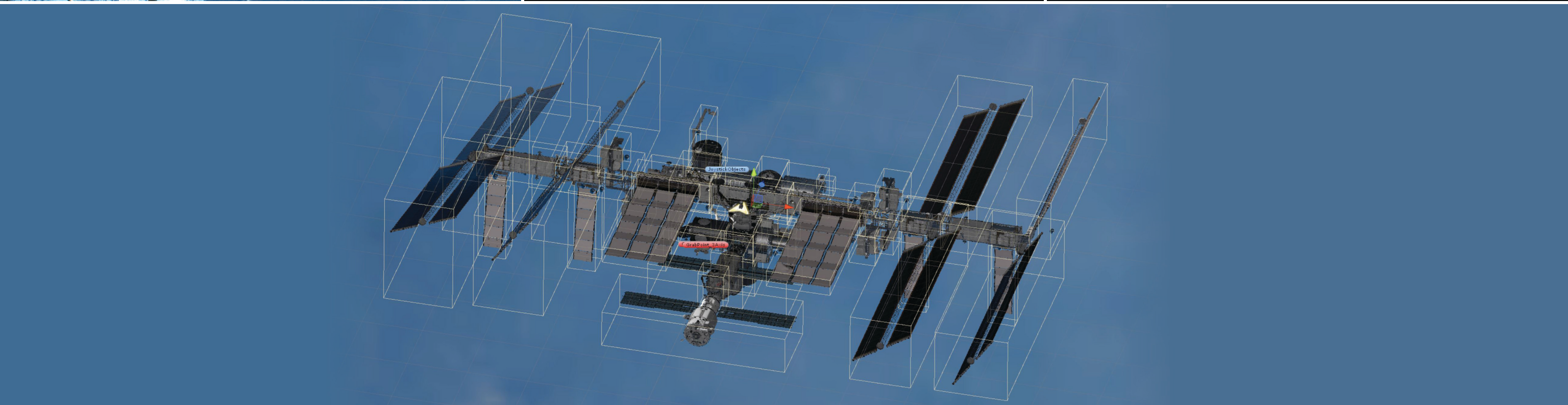
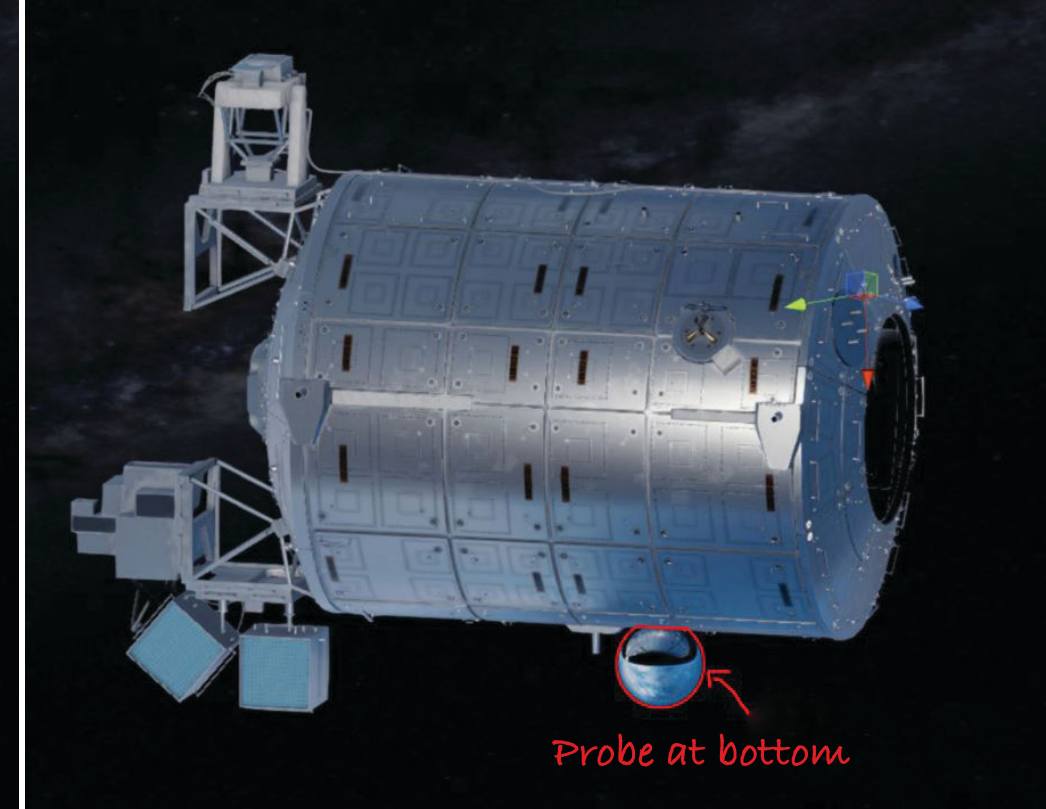
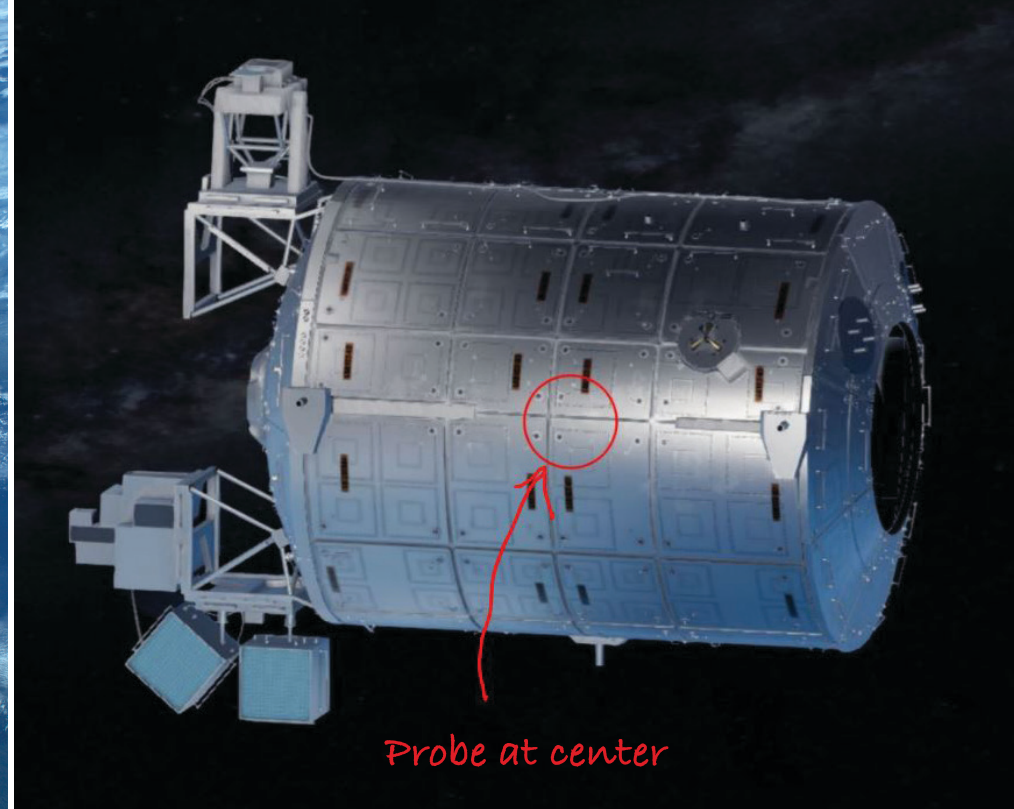
The blackness of space does not make for interesting reflections. More than half of the reflection environment is starfield, so we placed the reflection probes in spots that would emphasize the reflections from other parts of the station. For the solar panels, we placed the probes close to the truss to pick up more of those shapes.

For most of the modules, we took advantage of the single sided mesh and placed the reflection probes dead center in the middle of the module to get a relatively accurate reflection.

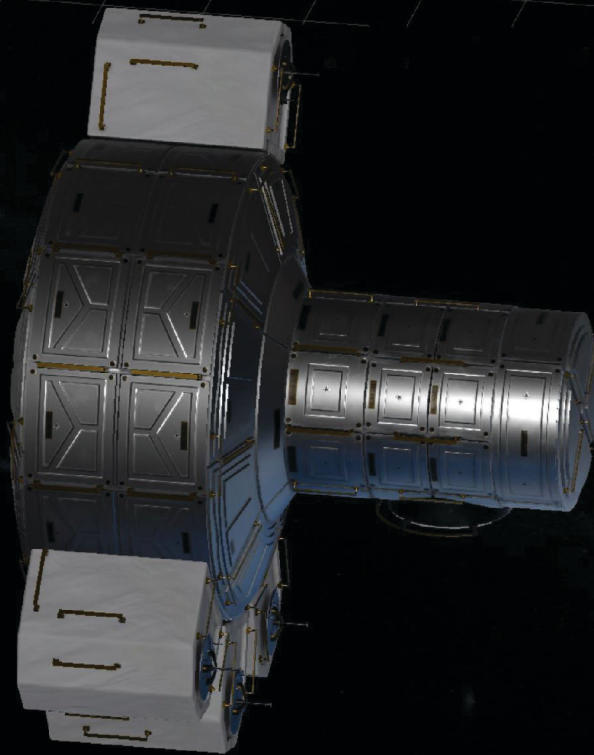
This technique did not work for the extra shiny modules. The reflection felt flat compared to the the references.

Some reflections required moving the probe between the earth and the module; giving a bit more life to the assets reflection in the parts facing starfields.

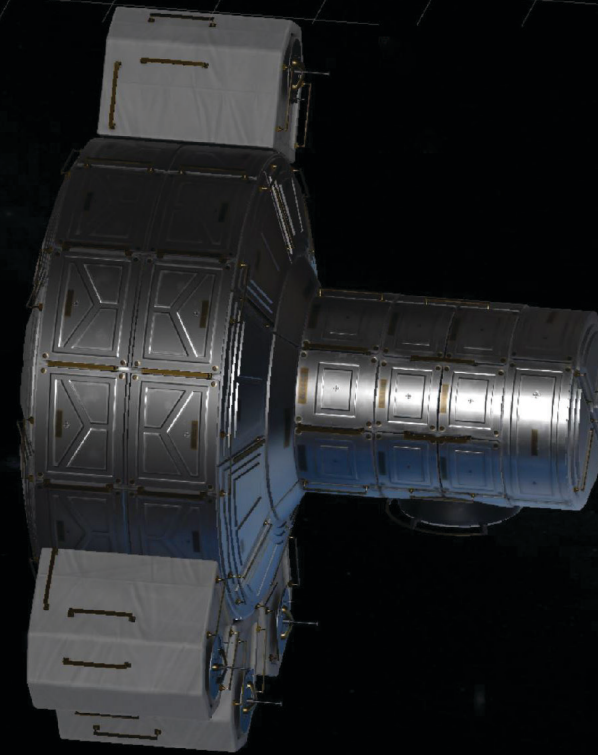
In the end, we ended up giving most of the modules their own reflection probe.



Base  
Material  
Setup



Final  
Material



## TEXTURE TWEAKING TO MATCH REFERENCE

For the most part, the maps generated in Substance were solid. There were just a few tweaks required in the Smoothness/Metallic to bring out some extra life and some adjustment to the albedo to match reference images more closely.

Across the station, each cloth element required individual attention to avoid feeling tiled, especially along the truss. Adding subtle rotations and scale to some of the cloth instances to help break up the pattern. The Normal Maps were reduced to half their strength to reduce shimmering in the small bolt details.

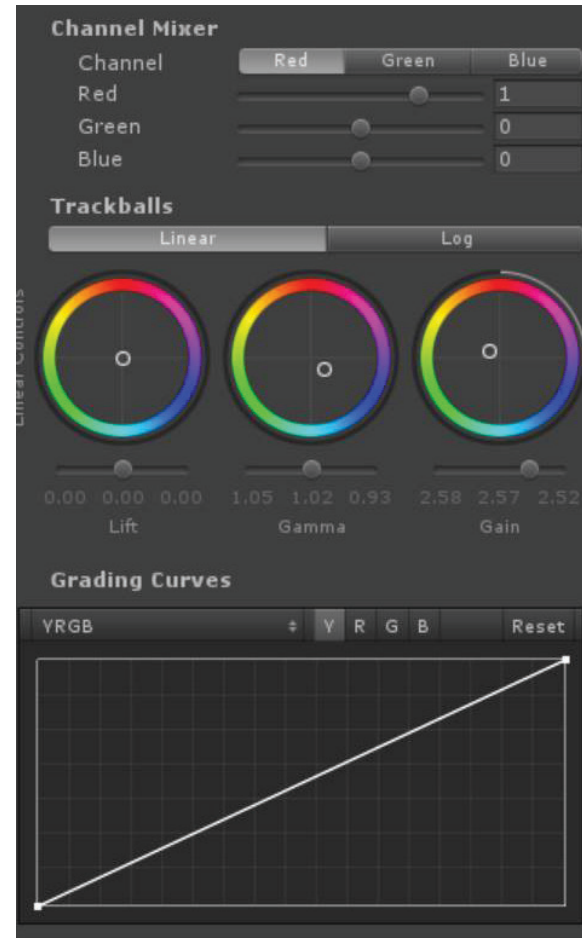
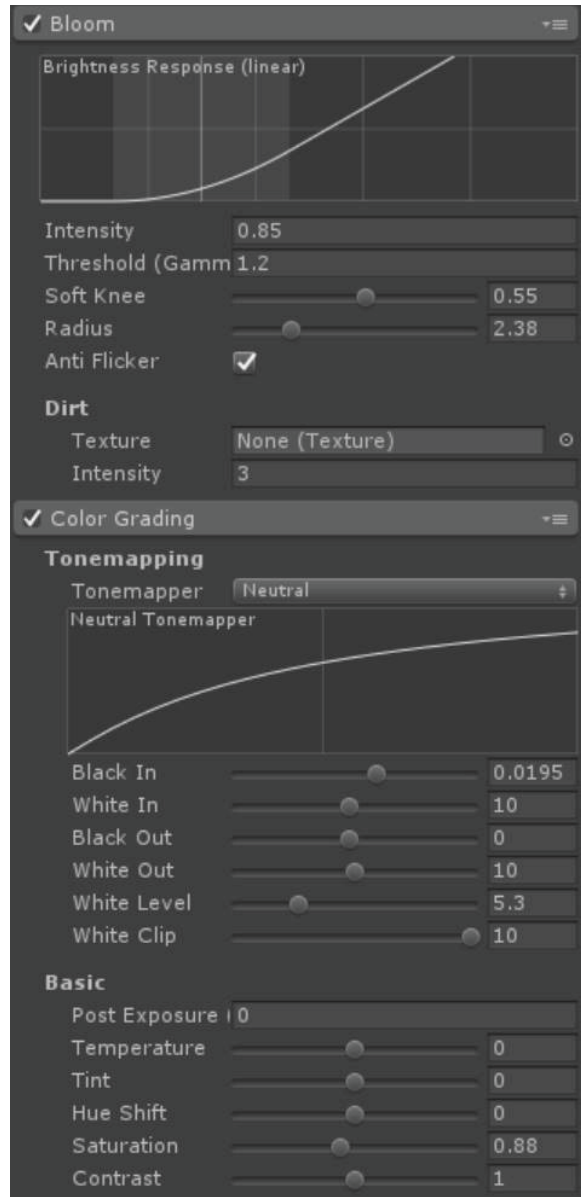
## LIGHT BAKING

A light bake for the entire station quickly started becoming an event that took multiple hours. We were forced to make our bakes very modular in checking the lighting, deactivating all but one section of geometry and reflection probes to make the bake faster. Occasionally we would model out boxes as stand ins for the deactivated portions of the station to improve the accuracy of the reflection. This was by far the most tedious part of the process. It was further compounded by the issue we found in Unity that occurred with very large bakes. For some reason unbeknownst to even our brightest Programmers, light baking data could never successfully transfer from machine to machine. This meant that all Version Control solutions were impossible when transferring lighting for a scene. Our programmers were testing with corrupted light bakes for the majority of the project.

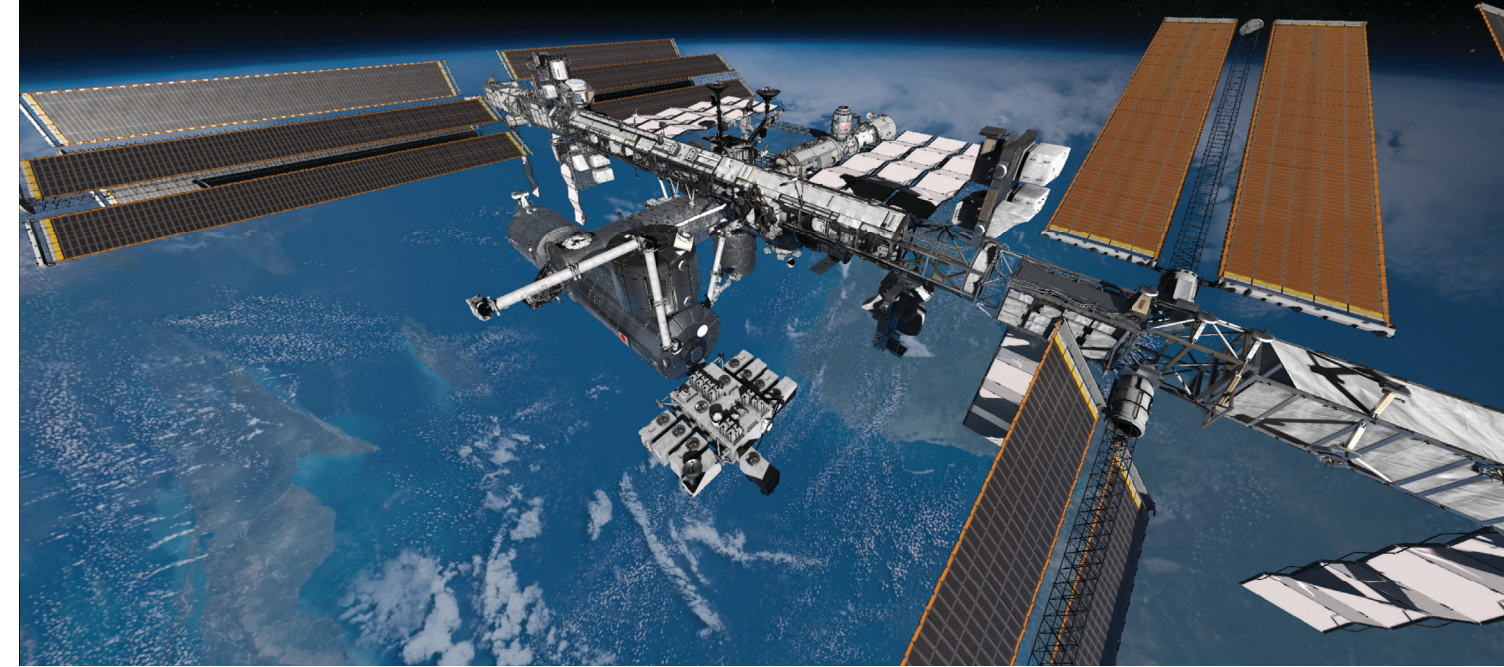
While this was an incredible headache and set back; eventually we settled on using a single machine for baking. This resulted in a GI Baking Cache of over 200GB! Our final version shipped from a single computer that had lighting data from all of the scenes.

## POST PROCESS

Post Processing was used sparingly in the final product to keep the project running at 90 fps. We ended up using the Fast Approximate Anti-Aliasing, Bloom, and some Color Grading. As much as we loved the Post Ambient Occlusion, it was just too heavy and the Screen Space Reflections would have required us to switch to a deferred rendering technique. These are the final post settings we used.

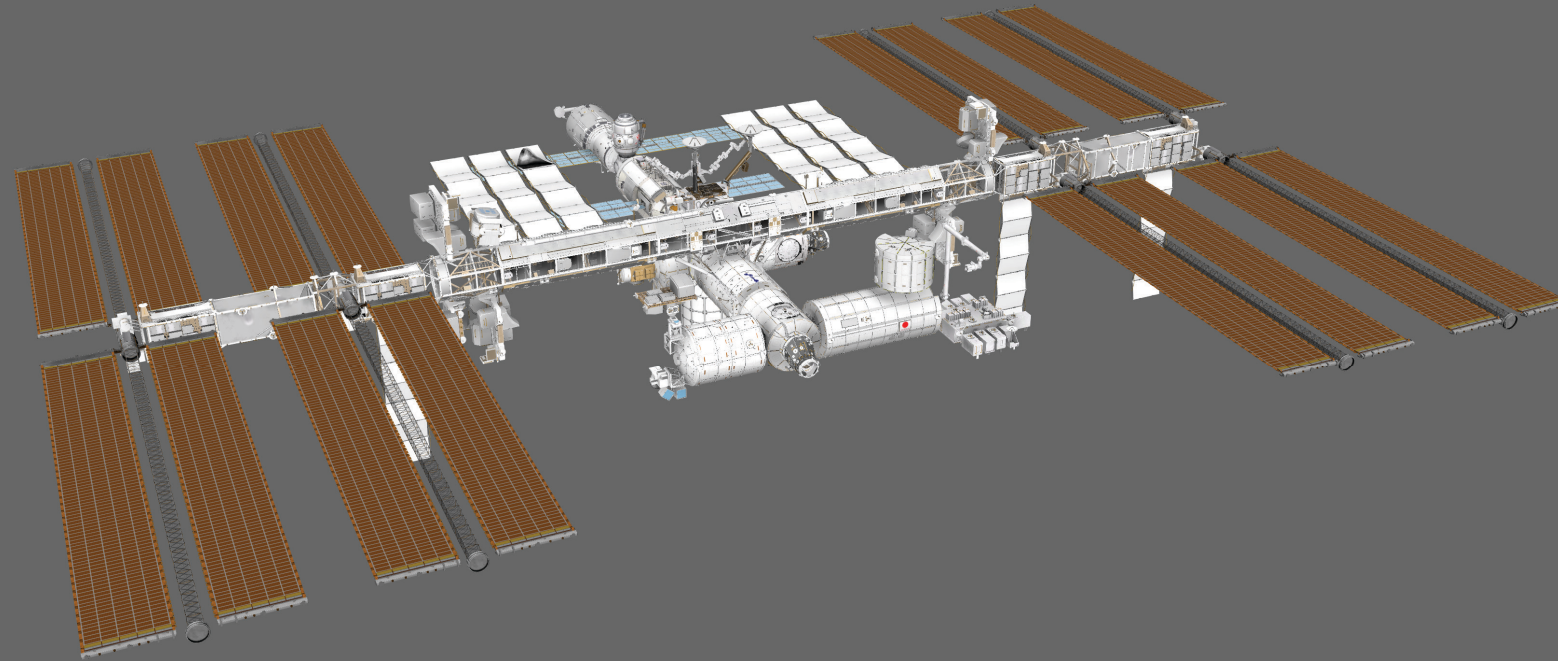


Before Post Process



After Post Process



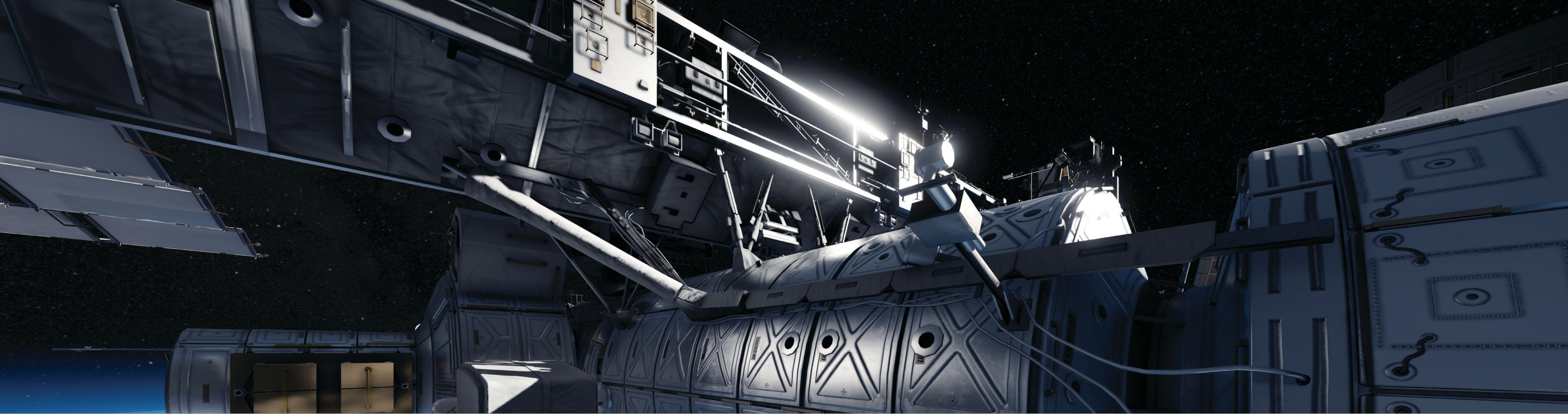


FINAL ASSET IN MAYA - 1.7 MILLION TRIANGLES, 30 MATERIALS / TEXTURE SETS

## CONCLUSION

Looking back, there are probably a few things we might have done better. Ideally, we probably should have explored some sort of hybrid approach between the two ISS examples. If we had used global tile-able textures in conjunction with unique grunge/scratch maps for each asset, our texture memory footprint would have been much lower with better quality. For lighting, it would have sped up iteration time if we had established lighting and visual look ahead of time with simple blockout meshes. The final look would have been set in stone with quick iterative light baking, then just expanded and tweaked once we got our final meshes in.

Some key takeaways: Definitely don't be afraid to think outside of the frame for lighting in VR. Try to create shadows that add dimensionality and stay interesting no matter what direction you view them from. Don't take reference too literally. Matching a photograph from a certain angle under certain conditions might shackle your experience to only working properly if viewed from a certain place. Don't be shy to give yourself more flexibility with some bounce lights and fills here and there to make each viewpoint impactful. Remember that you're lighting for things to work in every direction, for an audience that can travel around from every side. So make bold broad strokes, but then go check the impact from various vantage points. Finally, don't be afraid to "cheat" to get a look in VR. If you didn't get there the conventional way it's not that important. Sometimes all that matters is what it looks like to the user.



The production of Mission: ISS was under a year, with full production only happening in the last six months or so. Our team size also tripled over the course of development. It was incredibly fast-paced with little time for error. We strived to be as efficient as possible, but were still encouraged to experiment and see what worked best. VR after all is a constantly evolving industry with many techniques changing my the minute. We feel we pushed Unity to the absolute brink more than few times. We look forward to creating exciting and new experiences that change your perception of what can be done in VR.

**Mission:ISS is available now for free on the Oculus Store!**  
**For more information contact [ISS@MAGNOPUS.COM](mailto:ISS@MAGNOPUS.COM)**

**MAGNOPUS** - <http://www.magnopus.com>

**DEREK BLUME** - <http://www.derekblume.com>

**MISSION:ISS** - <http://missioniss.magnopus.com>

**LUKE SCHLOEMER** - <http://www.LTS3D.com>

**WWW.MAGNOPUS.COM**

Downtown Los Angeles | 523 W. Sixth St. Suite 1216 | Los Angeles, CA 90014