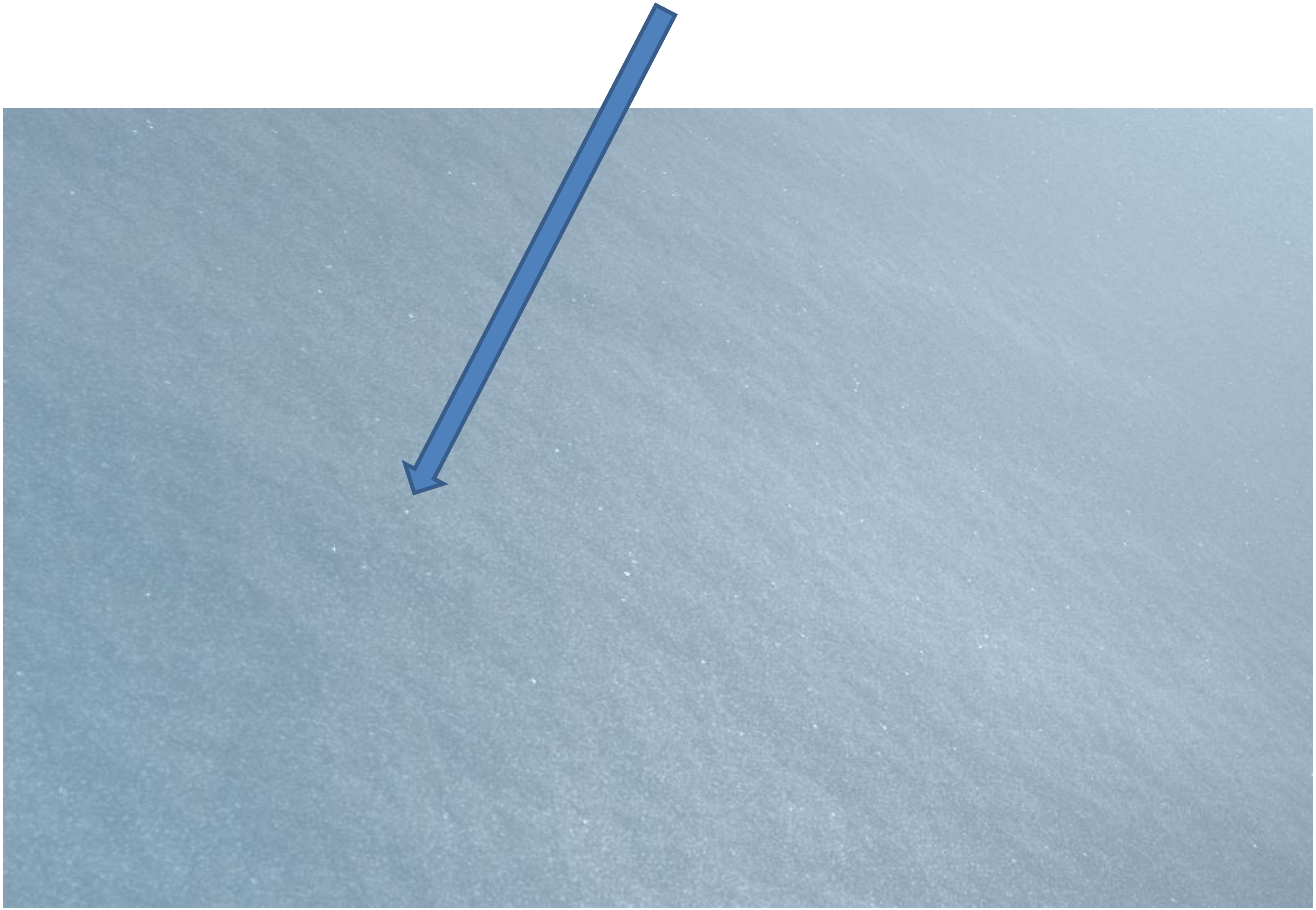
Snow Shader R&D in UDK
I3 DLC

Key Aspects:

These are visual elements aside from correct physical values I used to create my snow

- Sparkles
- Detail Normal Noise
- Detail Specular noise
- Large Tiling Shapes

Sparkles (Glints)



Sparkles (Glints)

What it is:
Small shimmering white spots representative of pieces of snow at glancing angles

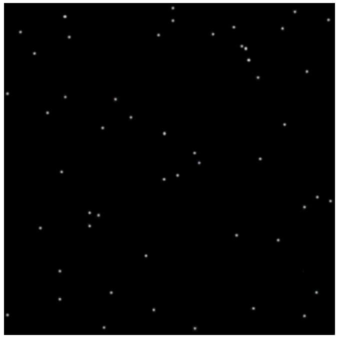
They were required to:

- Retain visibility most of the time (when not in super strong specular highlight)
- Exist at all view angles that were not in shadow
- Had to shimmer when the camera moves, and slightly when it wasn’t
- Had to be visible at far distances (not mip out)

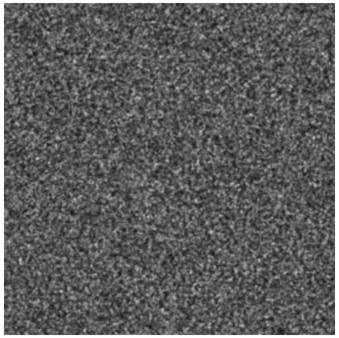
What I did:
-For visibility I added them to the specular map via a custom texture which I could control the intensity separately. This solved the first problem. Once I added environment mapping, it solved the second. This also allowed for custom tiling rate.
-I then used two (slowly) panning blurred noise textures, who's uv’s I added the x and y channel of my reflection vector too, as a mask for the sparkle texture. This meant as the camera moved, the mask moved creating the “shimmer” effect. It also meant that when standing still the panning noise textures slowly masked the glints in and out.
-Finally, since UDK doesn’t have support for custom mips, I blended different scaled UV’s of the glint texture based off of the pixel depth. This allowed the glints to stay present even far away.

SP Solution:
-The panning masks and reflection vector offset for the glint UV technique should still work here. Keeping it as a separate texture works well to give artists control of the glints. Adding those values to the final specular value should still work well.
-For the fidelity at a distance problem, custom mips can solve that problem.

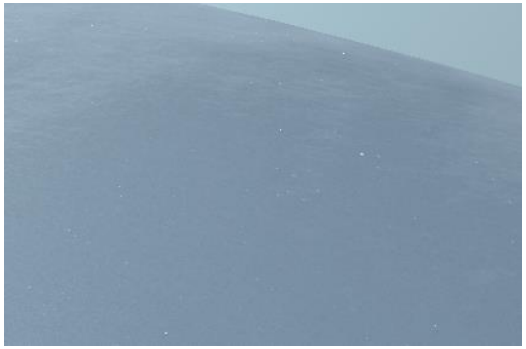
Glint Texture

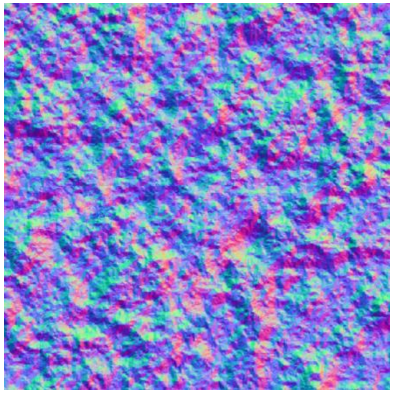


Blurred Noise

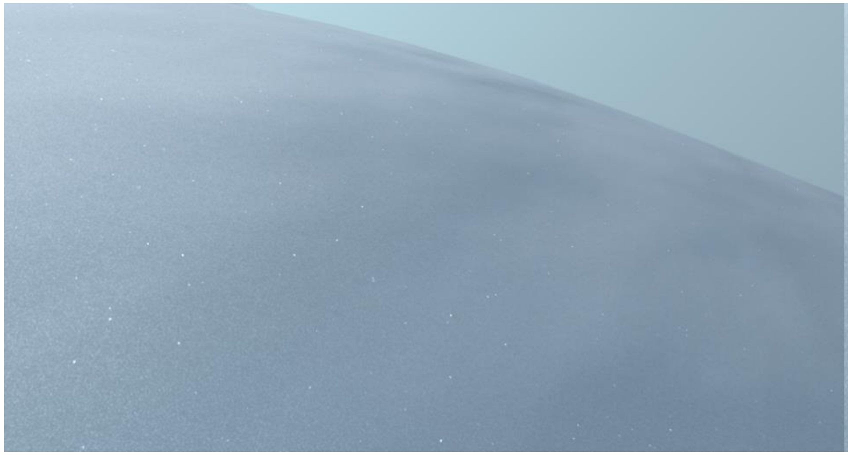


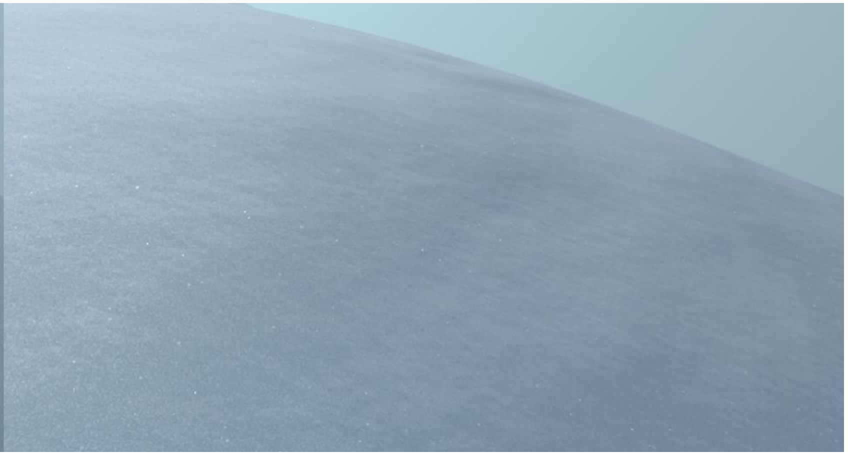
Detail Normal





These texture was used to really add to the “soft roughness” feel that snow has. I ended up turning down the intensity to a very low level, and also ended up with a relatively low tile rate. The final results made a big difference, especially where the light falloff starts.





Specular Noise

What it is:
Mini fidelity added to the specular reflection (both light, and environment source) that really sells the surface as detailed

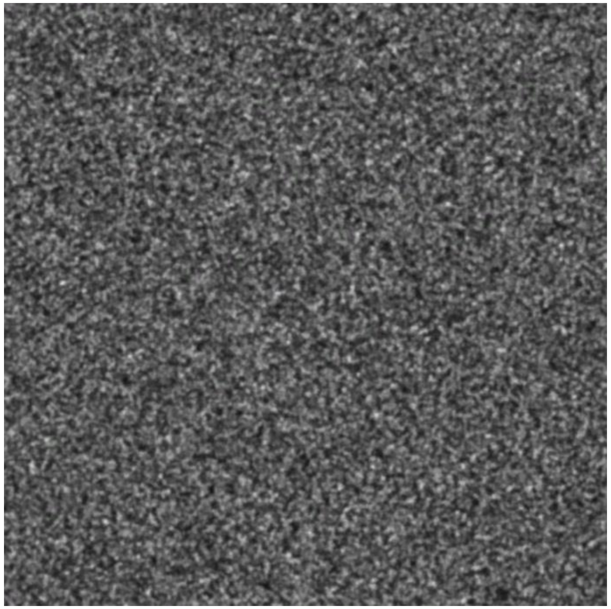
They were required to:

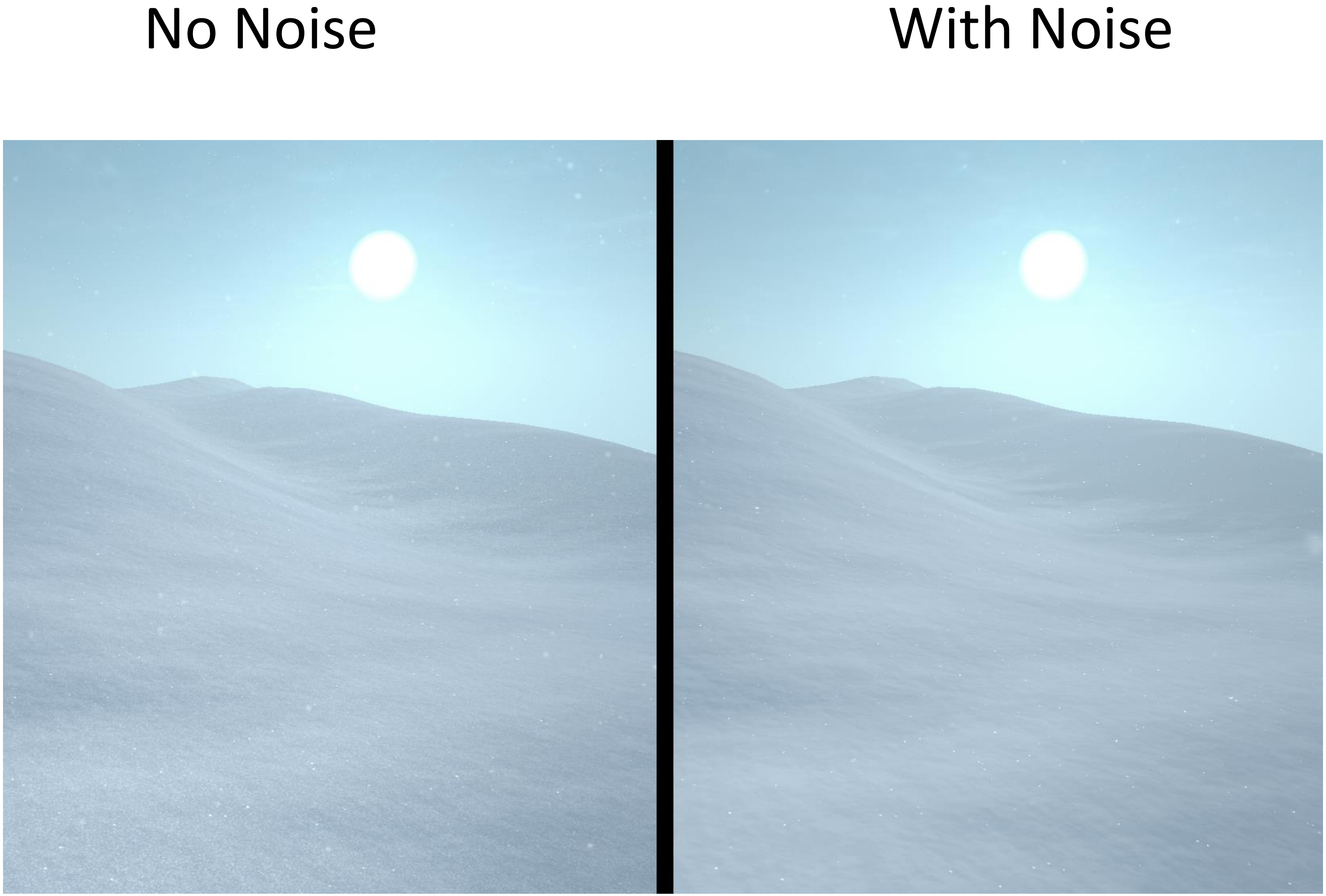
- Retain detail (ie stay visible) both at near distances, and at far distances
- Have controllable contrast with how it interacts with reflections

What I did:
-Similar to the glints, I faked custom mips by changing tile rate at different pixel depths.
-I multiplied my noise texture by a control variable, added that to 1 (so the noise range was at least one), and finally multiplied my specular map for the overall snow by the result. I went with this technique so that the base "spec" of the material could always be controlled with a variable or a texture, and the noise could be controlled 100% separately from that. It also meant that you could scale the intensity of the noise before having it affect the base specular value. This prevented the noise from removing the base spec in any area creating strange over contrasted noise areas

SP Solution:
-We can replace my custom tile control with proper custom mips to keep the detail visible at range
-The rest should be doable without much problem

Other Notes:
-The Journey team ran into Anisotropic Filtering problems with their implimentation of this. Check out their work here:
Journey Sand Rendering
-The reason I added the noise to the specularity rather than the roughness (which would be correct technically) was twofold. The first reason, was it allowed even the brightest and most solid highlight to have noise. If it was done with gloss, then the center highlight would blast out the noise in the tightest spots. The second was that having the noise in the gloss would have meant adjusting the behavior of the light with the material overall, potentially lessening the effect or adjusting the final visual result in non highlight areas.

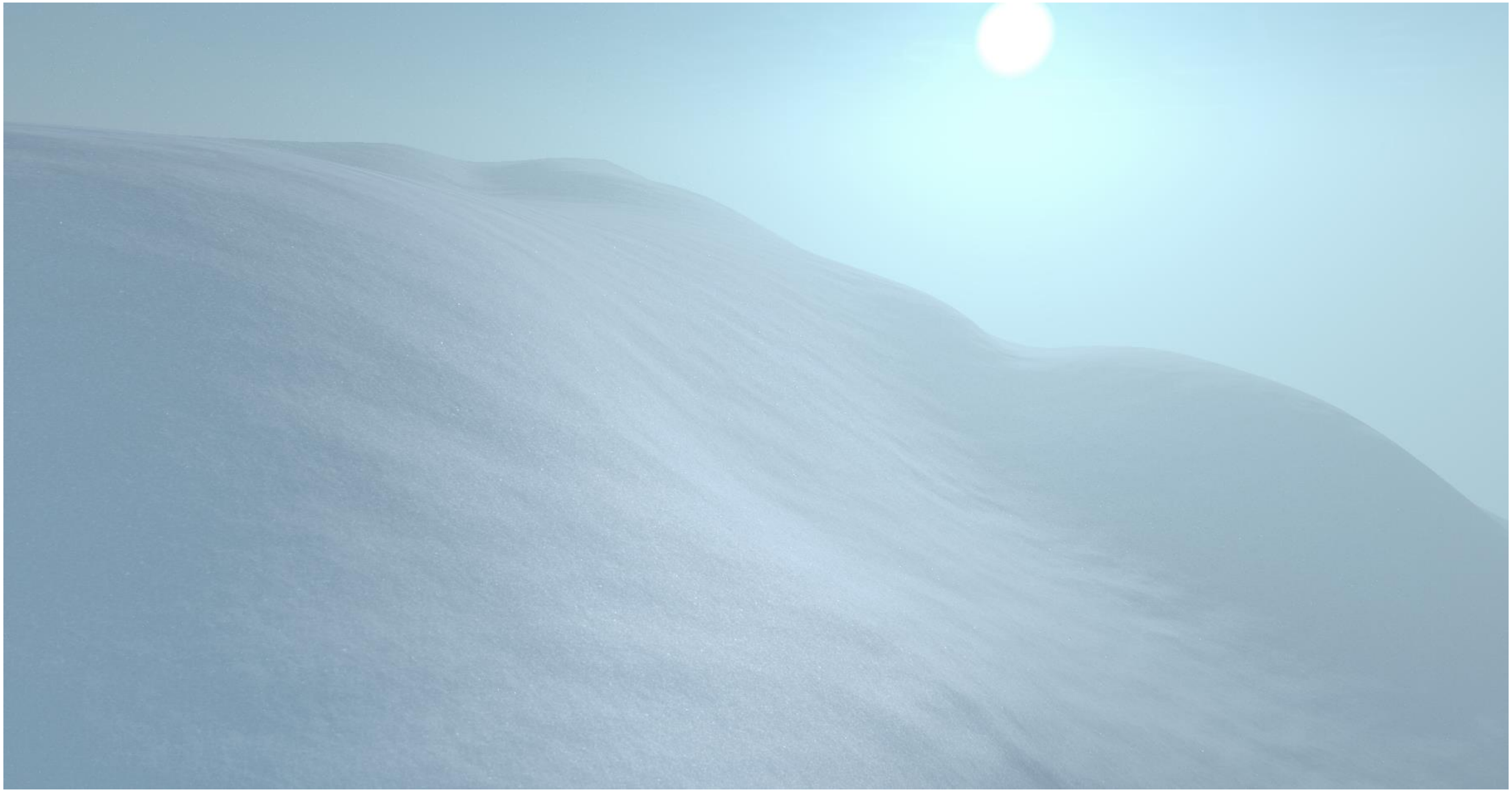




No Noise

With Noise

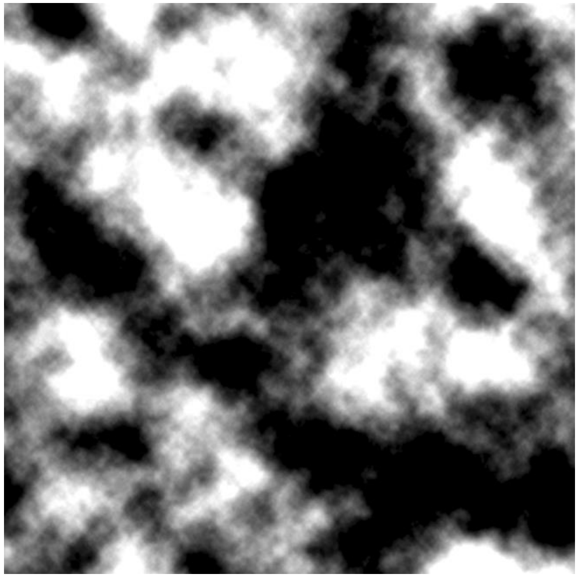
Large Normal Noise(with)

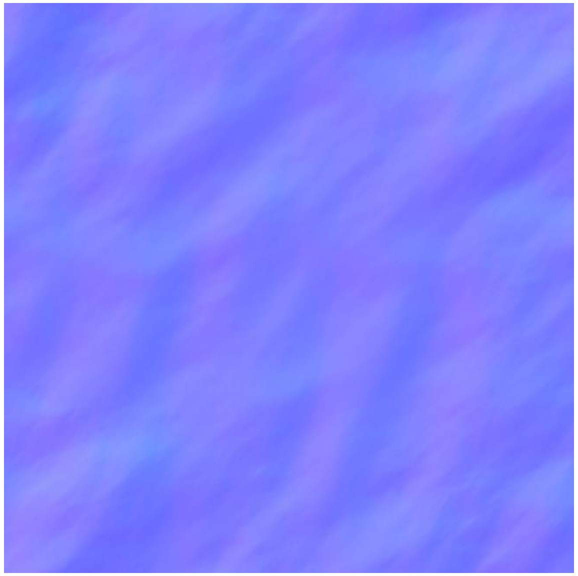


Large Normal Noise(Without)

Large Normal Noise

What it is:
A low tiled normal map to add appropriate variation to the entire scene. Also helps to sell “snow”





They were required to:

- Break things up, not over repeat

What I did:
-Created a good normal map for the snow
-Adjusted the intensity off of render clouds

SP Solution:
-Same as I did

Other Notes:
-Normal map intensity was make or break. Softer was usually better.
-I used the renderclouds to go between no normals, and some. The detail normals were added later to make sure everything had some visual information

Final thoughts and implementation

The way I see this best being implemented in our current system is as a new type of shader. Here are some suggestions on that shader:

- Use the red channel to blend in a crumpled snow texture
- Use the green channel to blend in a more hard packed, ice like type of snow
- Leave the blue channel open for artists to put in their own texture. This lets them blend the snow out to other parts of the environment.
- Remove all artist control over the red and green channel with the exception being whether or not it is world space UV’s or not. Leaving artists in control over the 4th material of their choosing is necessary to let it match with the rest of the surroundings.
- The above suggestion includes all aspects (from glints to the spec noise, normal map, etc.). Having the extra channel lets them pic variation of their own choosing, but keeping all the rest of the snow under lock and key keeps things contiguous. This matters because as I found on my own, tweaking minor things can make or break the final visual result in a big way.

What is in the UPK:

- Snowshader 004_INST is the best tweaked version of the snow, however most of the core values in the base material are still correct.
- Extra textures or models to be exported for testing