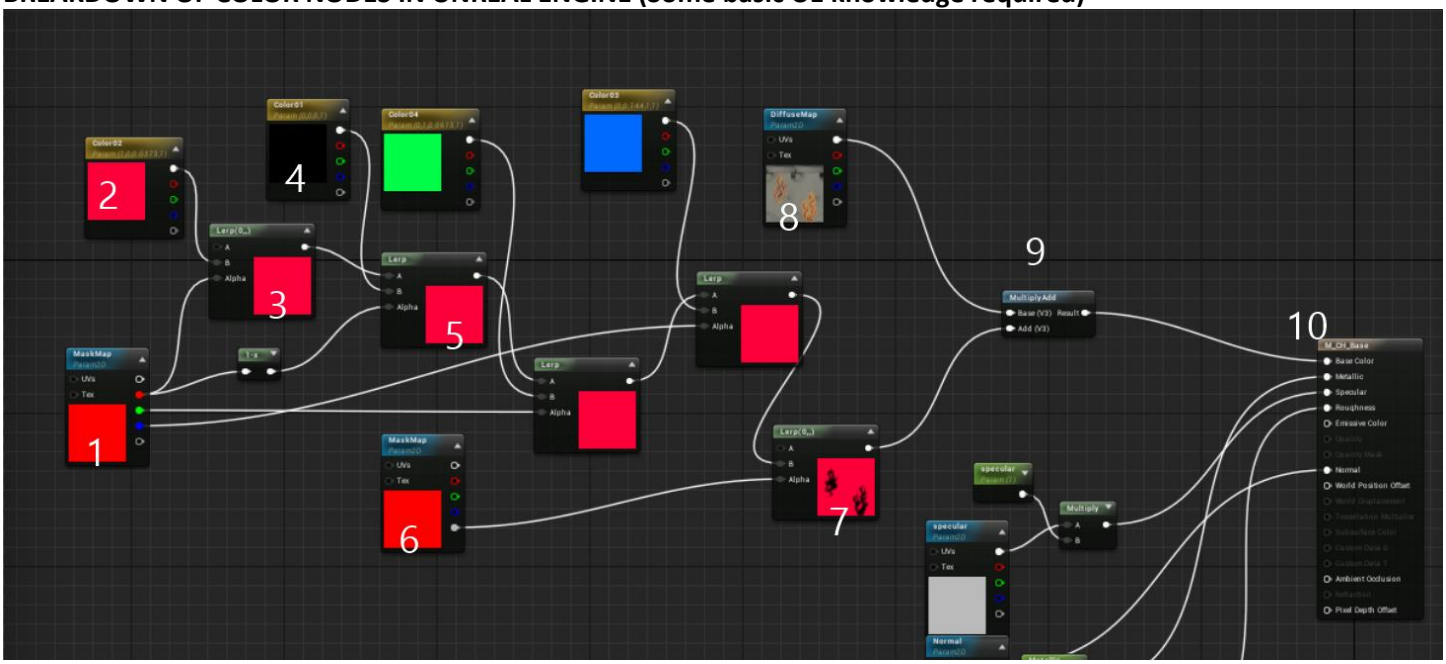
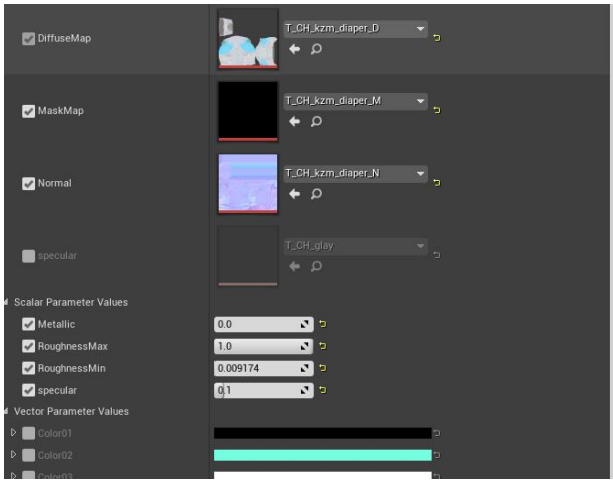
BREAKDOWN OF COLOR NODES IN UNREAL ENGINE (Some basic UE knowledge required)

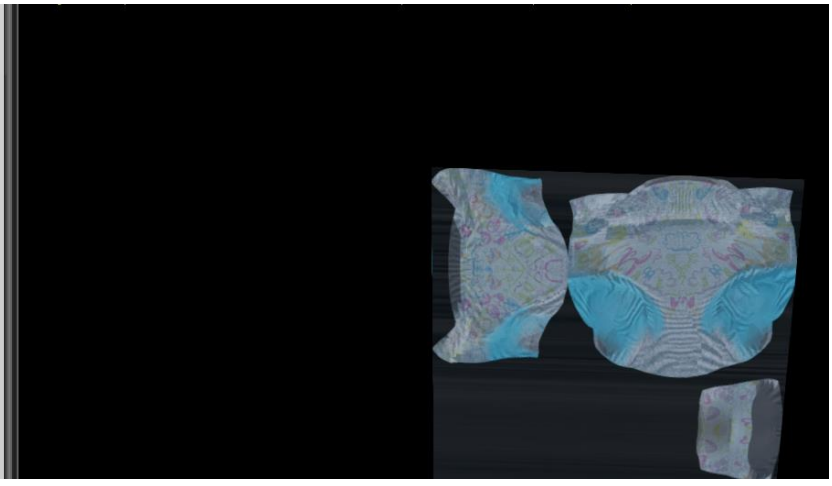


1. Original MaskMap in a TextureParamater2D node.
2. First Color. In this case I'm using red as the replacement color
3. Linear Interpolate Node (Lerp). I plug the replacement color into B (result) and I link the red color channel to the Alpha. This makes it so that all of the red color in the mask is replaced with my replacement color.
4. Second Color. In this case I'm using black.
5. I create a new Lerp and plug the previous one into the A (input) of the new one. I make a new color node and connect it to B (result). I then pull the red channel from the MaskMap and plug into a OneMinus (1-) node which reverses it. My goal is to capture the alpha black of the MaskMap.

NOTE: I repeat the chain process for all 4 color slots.

6. I use a duplicate of the MaskMap to keep things a bit organized.
7. I run the actual alpha channel of the Original MaskMap to the final Lerp in the chain. This makes all of the changes to the MaskMap only apply to the non-transparent portions of the mask. In this case, the flames are transparent in the Original MaskMap. Essentially, I know have an adusted MaskMap that incorporates all of the Materials base colors.
8. Original DiffuseMap in a TextureParameter2D node.
9. I use Multiply/Add and input my DiffuseMap into the A (input) and my Adjusted MaskMap into B (results)
10. I then run the final results into the base Material section.





I created a MI from the Base Material and I can see that I have 4 texture parameters available. I checkbox the textures I'll be using and select the appropriate texture files. Boom we have a working Material Instance that we can cook and input directly into the game.

In this instance my MaskMap looks black but it's actually a transparent dxt5 which allows the no coloring and a full reveal of the diffuse map. If I were to use a different mask, I could use the Vector parameter Values at the bottom of the screenshot to adjust them directly in unreal engine.

I'm still learning but this was the hardest part of the process to understand. Hopefully, this helps anybody else who wants to take a stab at the process.