Week 2 Task – Material Systems and Techniques

Callum Wade 2404781 07/10/25

1. Stylized Water Material with World Position Offset

What is it?

A Stylized Water Material with World Position Offset is a shader technique most commonly used in Unreal Engine to create water surfaces that feature dynamic wave movement and effects with a heavy performance cost.
Animated Meshes can be used to make water, however it can be more costly to performance due to the fact that the CPU has to update the vertex positions of the water for every instance each frame, reducing frame rate. Whereas, the world position offset (WPO) calculations happen entirely in the GPU for each instance of the mesh using that material. Also, all objects using the same material can be batched into a single draw call, reducing CPU overhead.
The direction and intensity of the waves can be changed in the material by adjusting different parameters or functions. One way to adjust the wave direction is by adjusting the directional vector's X and Y components to the flow's direction. To change the wave intensity, you could use a scalar parameter to multiply the output of the wave displacement calculation that determines the wave intensity
For a smooth wave deformation, the object that the material is placed on requires a high vertex density so that there are enough vertices to move for the wave shapes to appear natural. The recommended vertex grid for using this material is a 32x32 or 64x64.
The water material uses and reacts to light in different ways to create realistic looking water. The way the water reacts to light is by changing colour and brightness based on light direction and intensity. The way it uses light is by using effects like transparency to show and light up object below the water. Another effect used is refraction to distort the objects in a realistic way.

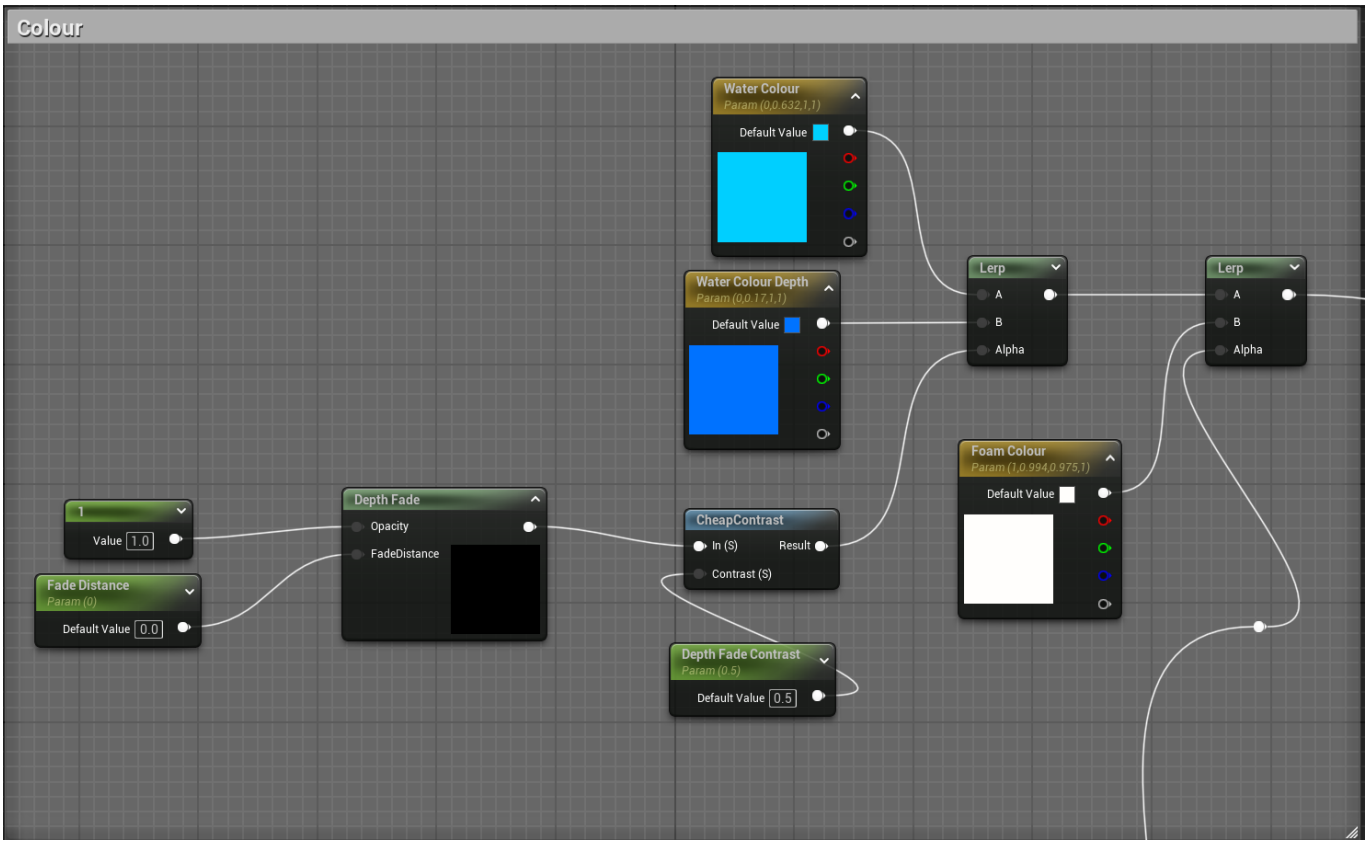
Final result


Water Material

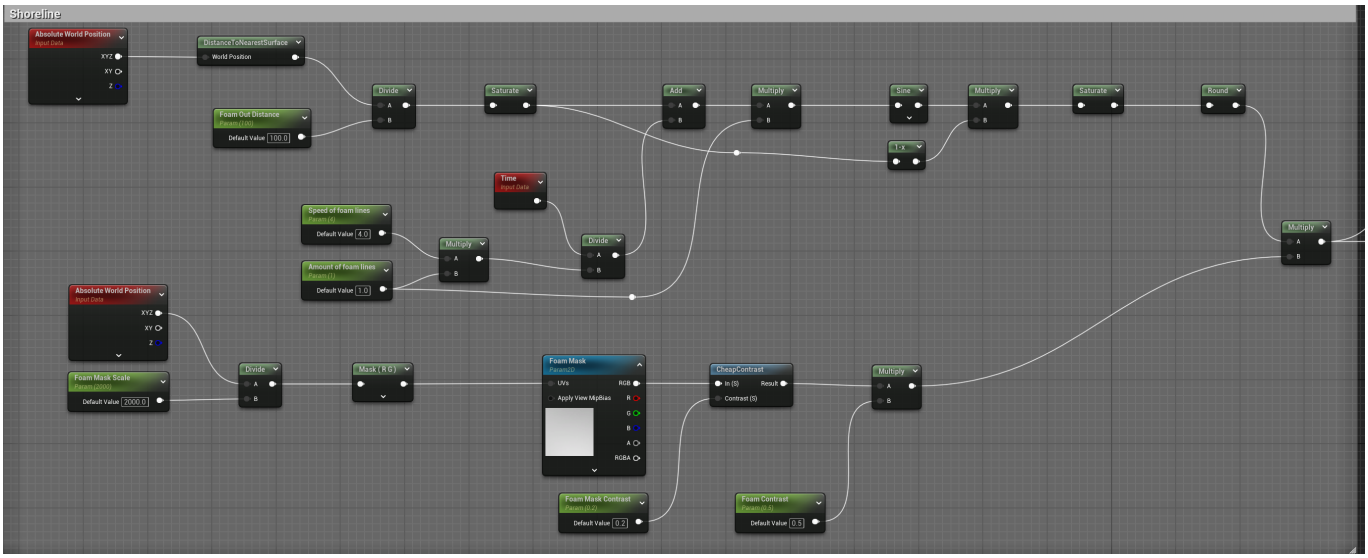
Water Material Test

Blueprints

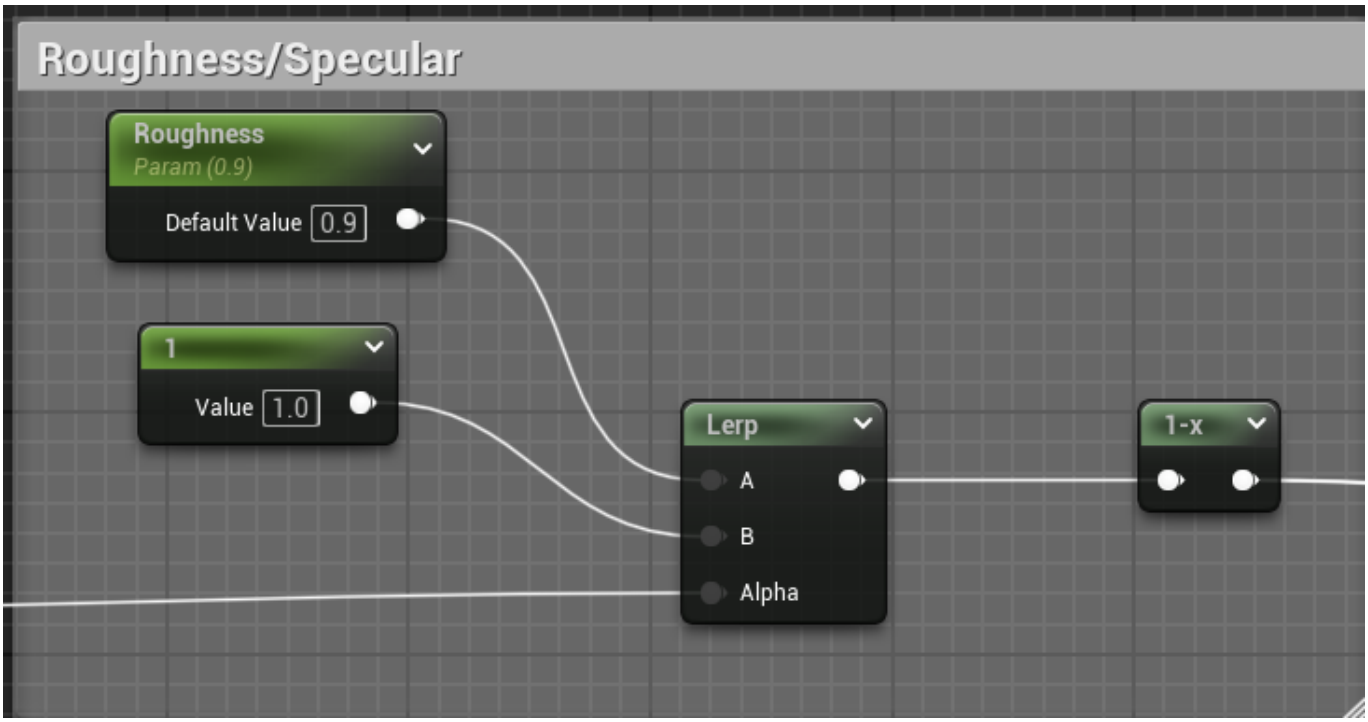
Colour Blueprint



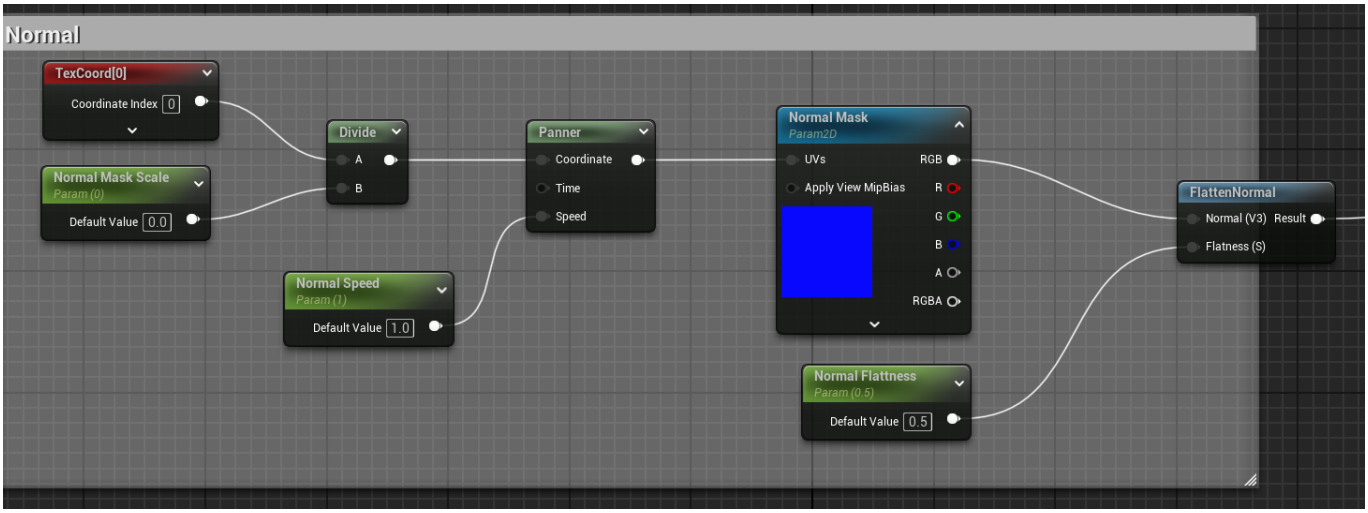
Shoreline Blueprint



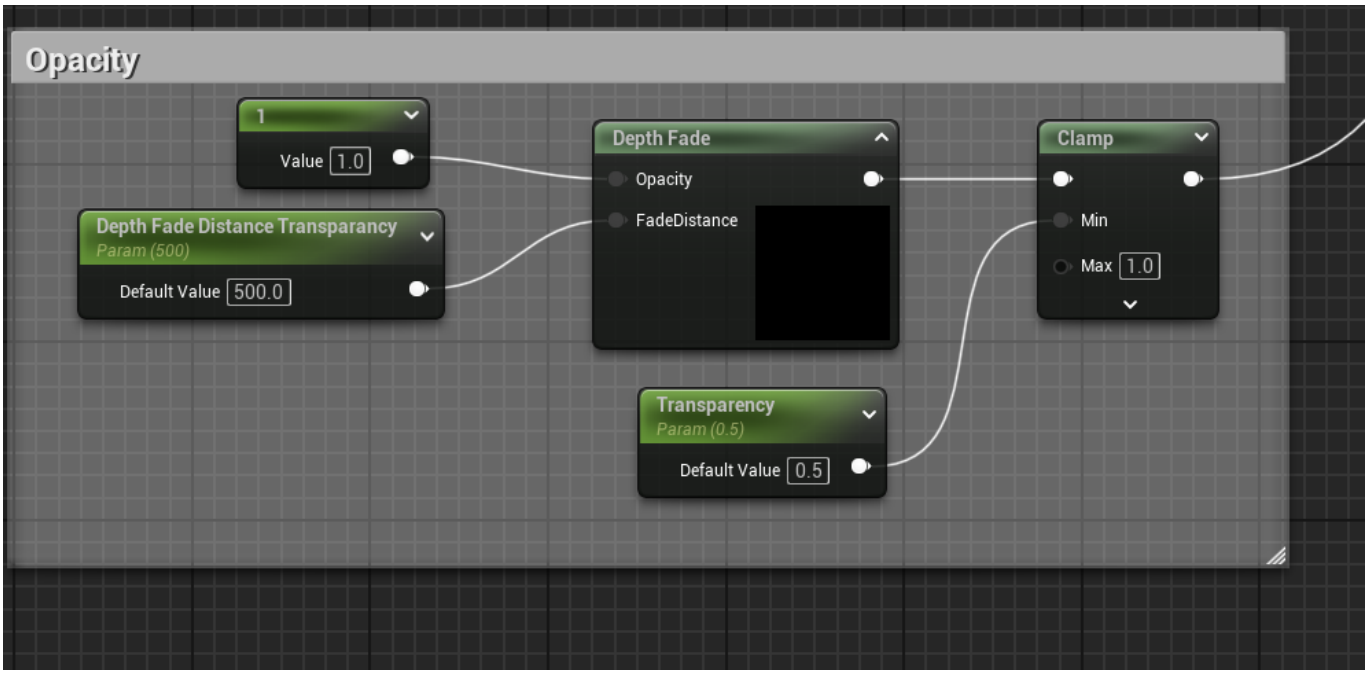
Roughness Blueprint



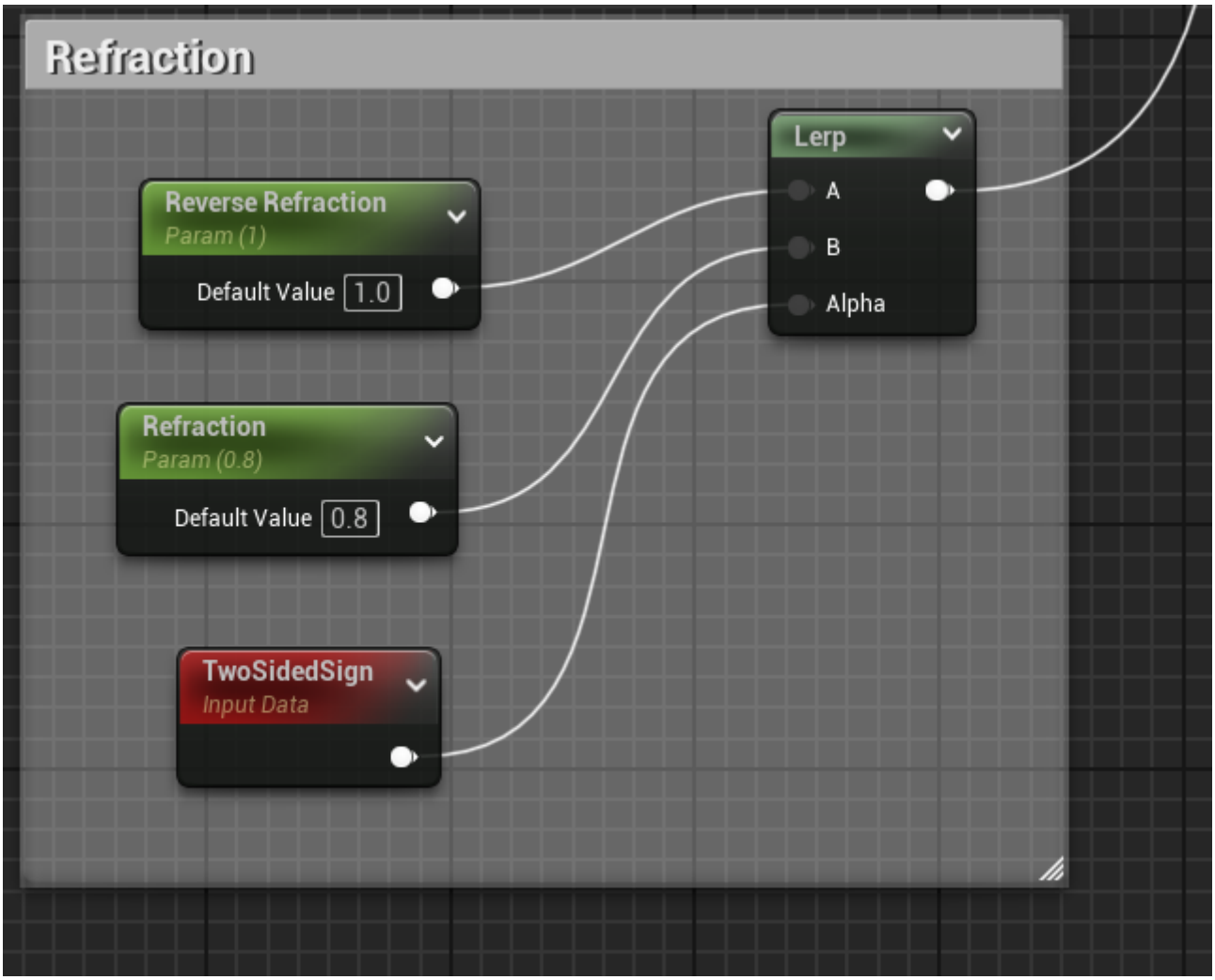
Normals Blueprint



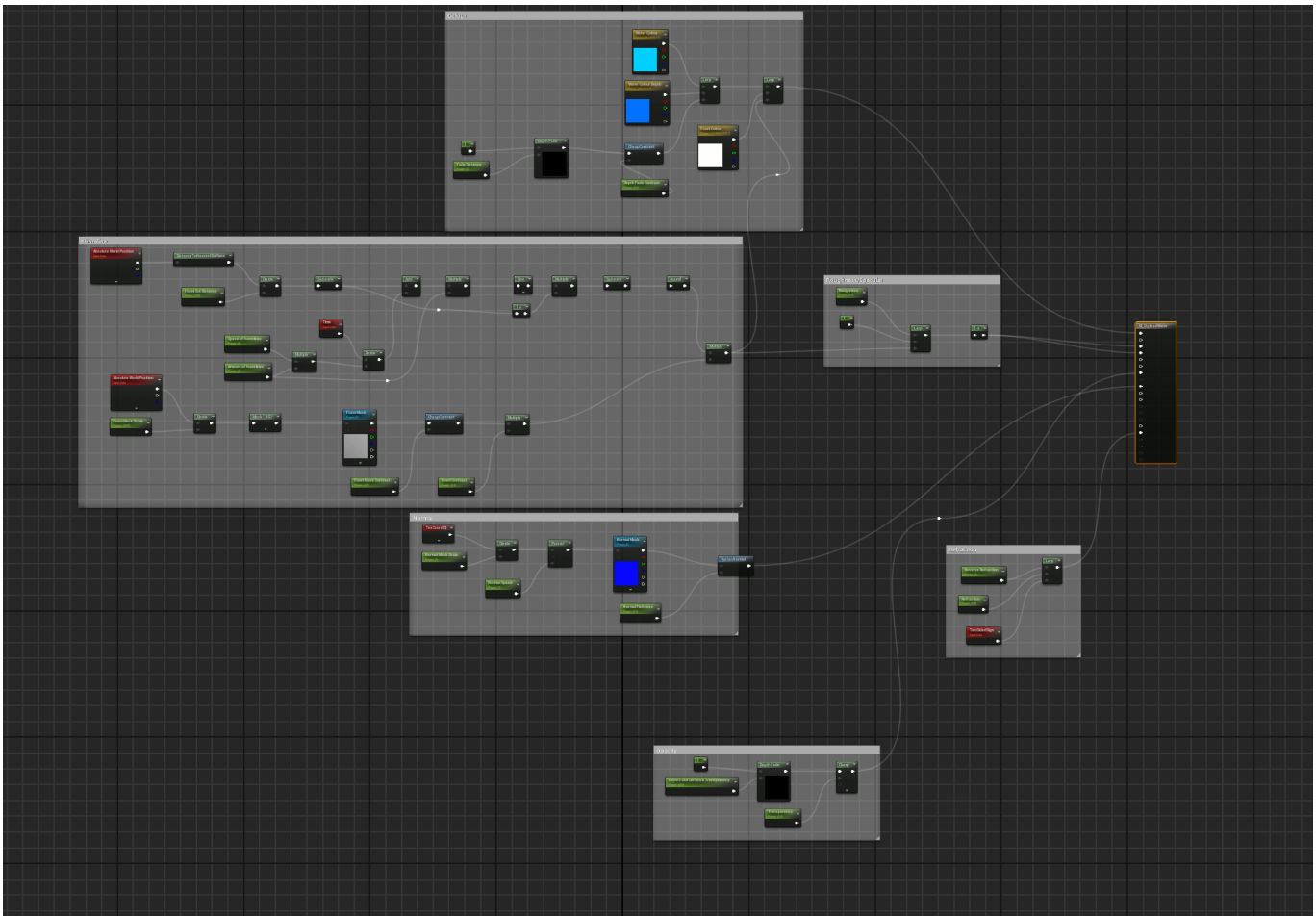
Opacity Blueprint



Refraction Blueprint



Entire Blueprint



2. Material Parameter Collection (MPC) System

What is it?

A material parameter collection (MPC) system is a feature used in Unreal Engine that allows multiple materials to share and use the same set of parameters. An MPC stores scalar and vector parameters in one collection that can be changed in real-time, affecting all materials referencing that collection simultaneously.
MPC should be used over using individual material parameters whenever possible due to how easy and efficient it makes modifying materials. When changing parameters for multiple objects in real-time, using a MPC allows you to change multiple parameters for multiple different materials all at once whereas using individual material parameters means that you have to constantly change between different materials to get to different parameters.
Using an MPC upholds visual consistency across materials by centralising key parameters that control common visual traits. By referencing the same MPC, multiple materials share identical values for parameters like colour, lighting and transparency, ensuring they all respond uniformly to changes.
As well as being a more efficient method, using a MPC provides several performance benefits including, reduced draw calls and material instances, efficient real-time updates, lower CPU overhead and improved memory usage.

Problems during creation

1. When lurping between the wet and dry textures, they would change too quick and too much.
2. When going to multiply the base colour of the landscape with the base colour, it gave an error as a float 3 and float 4 couldn't be multiplied together.

3. The global controller blueprint wasn't properly changing the light intensity of the sky light in the scene.
4. I tried using a button to control changing the time of day but it wasn't doing anything.

Solutions

1. I added a divide node to the RainAmount parameter before it went into the lurp node so that the parameter can be changed easier while the material would change slower.
2. To solve this problem, I used a component mask to turn the float 4 into a float 3.
3. Instead of adjusting the intensity of the sky light, I changed the directional light.
4. Instead of using a button to change the time of day, I used a delay node to change it after 5 seconds.

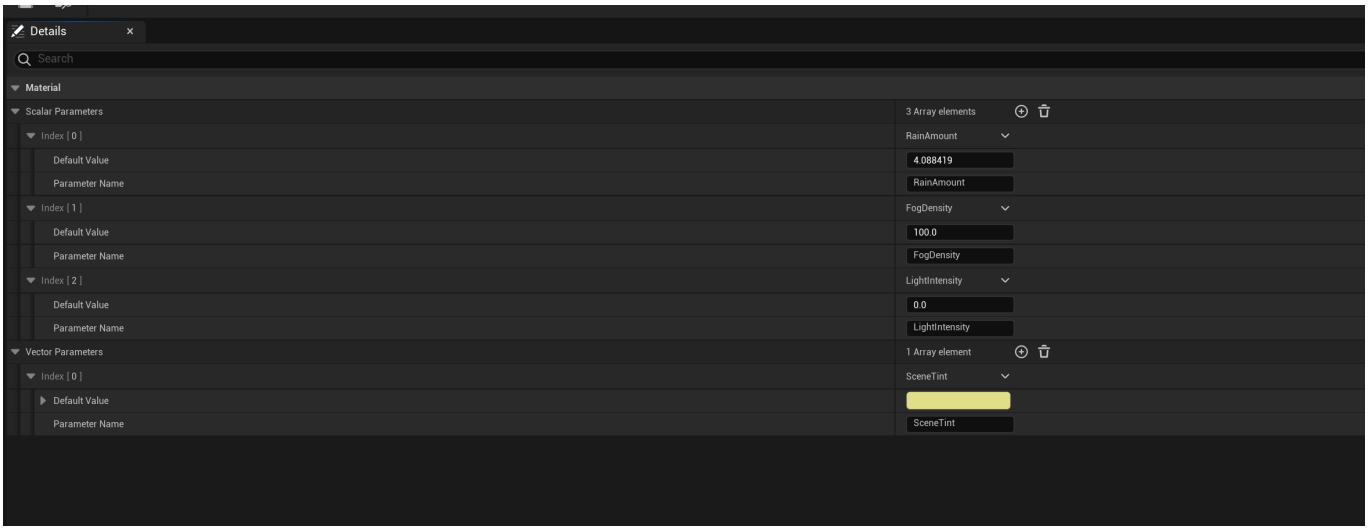
Technical choices

Final result

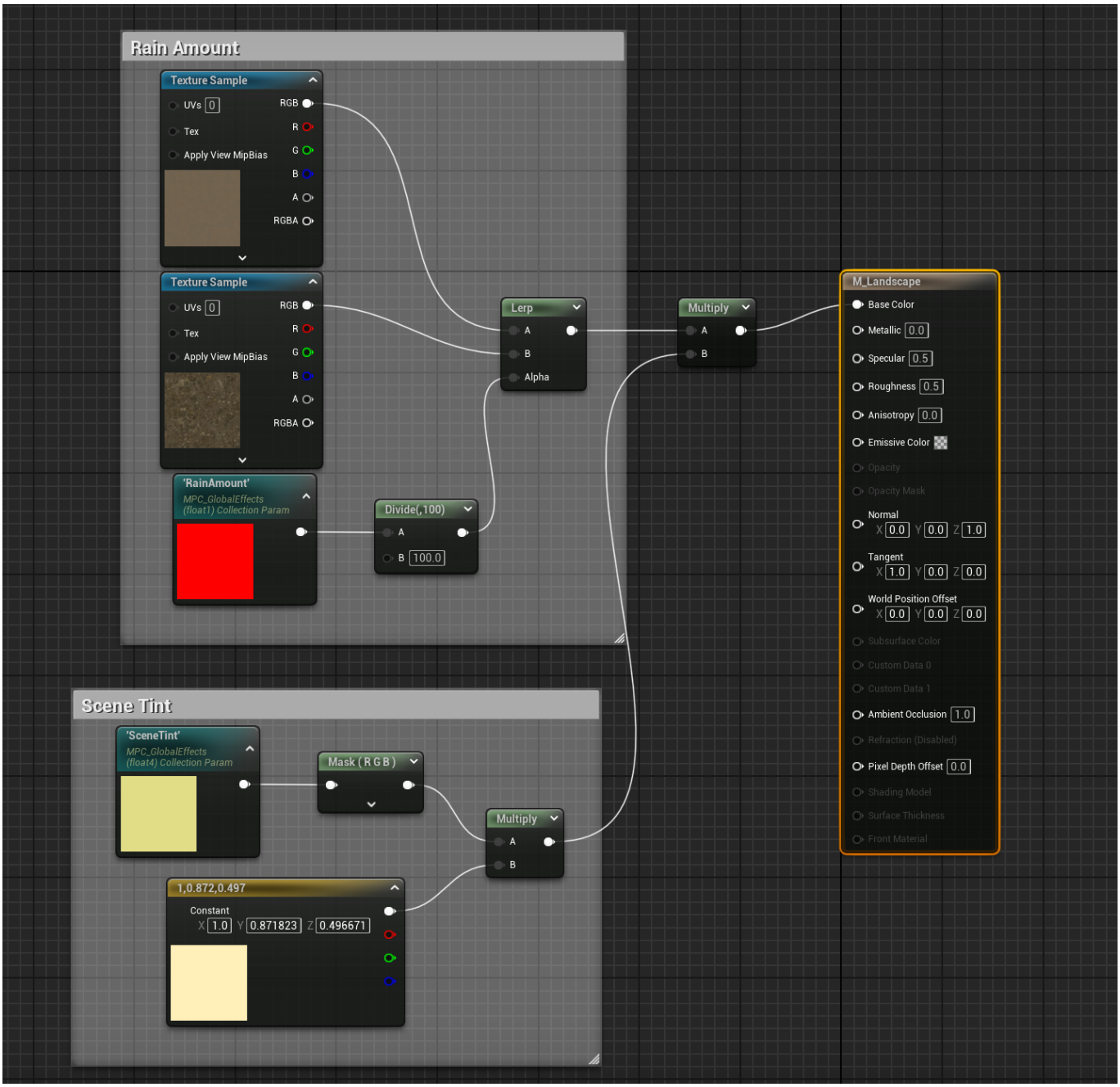
Material Parameter Collection (MPC) System

Blueprints

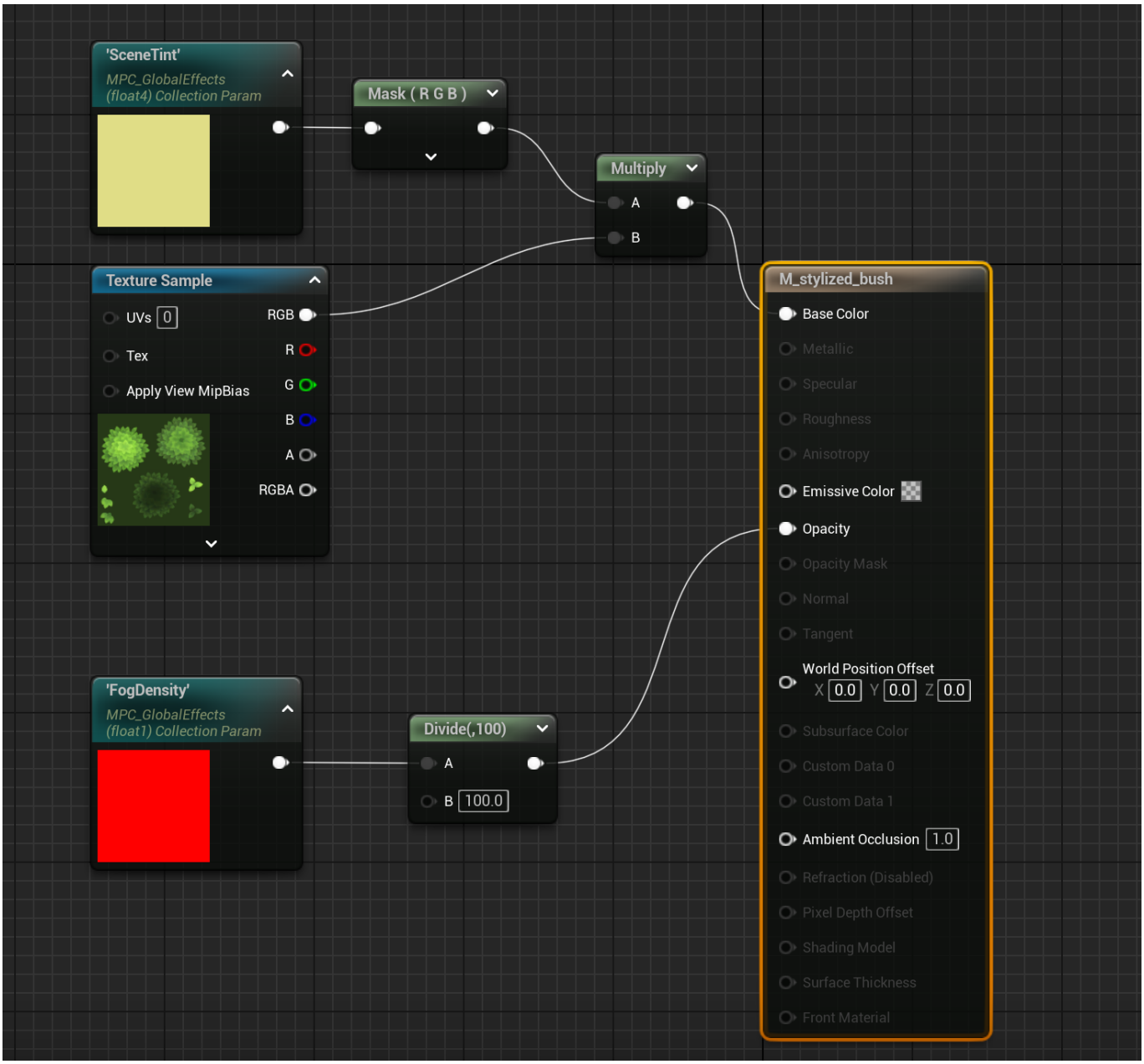
MPC



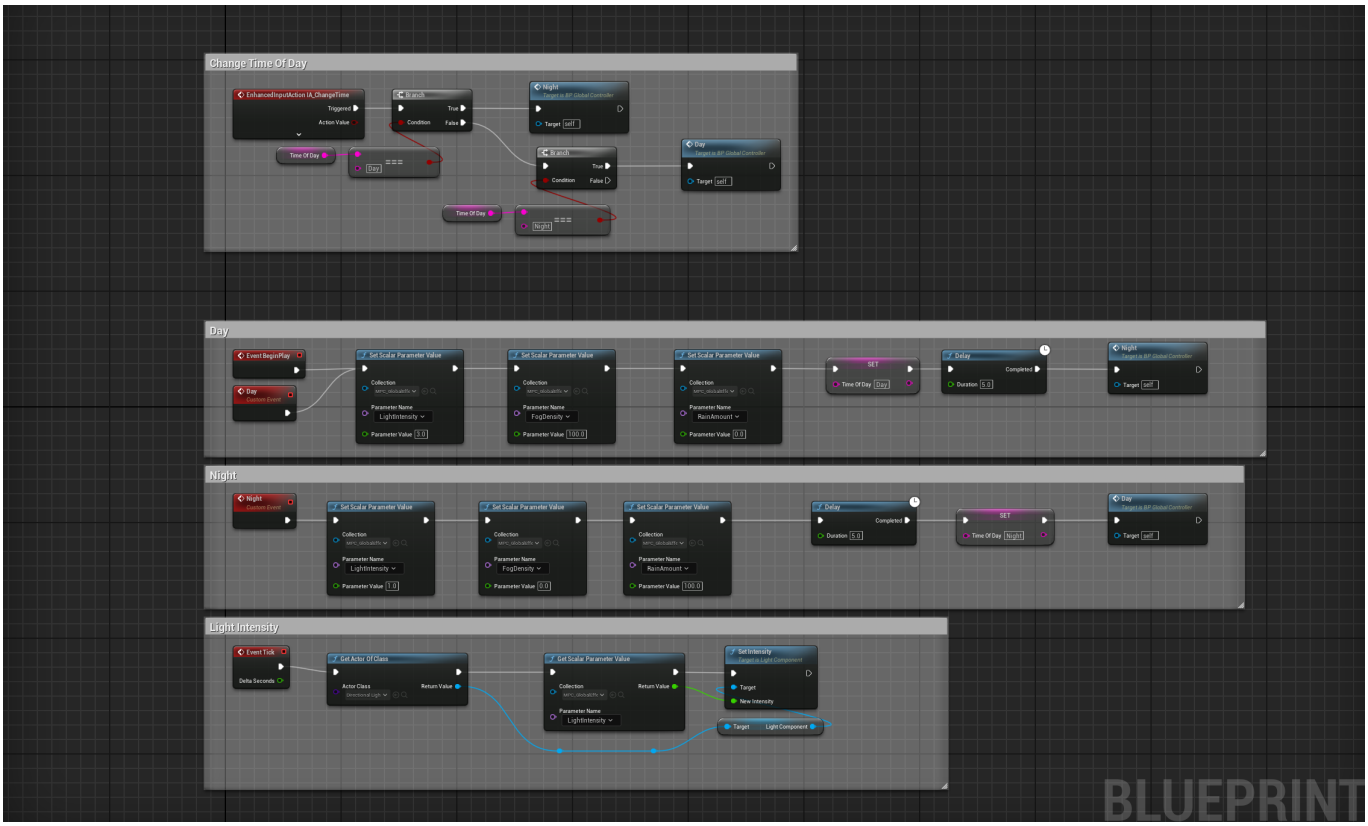
Landscape Material



Bush Material



Global Controller

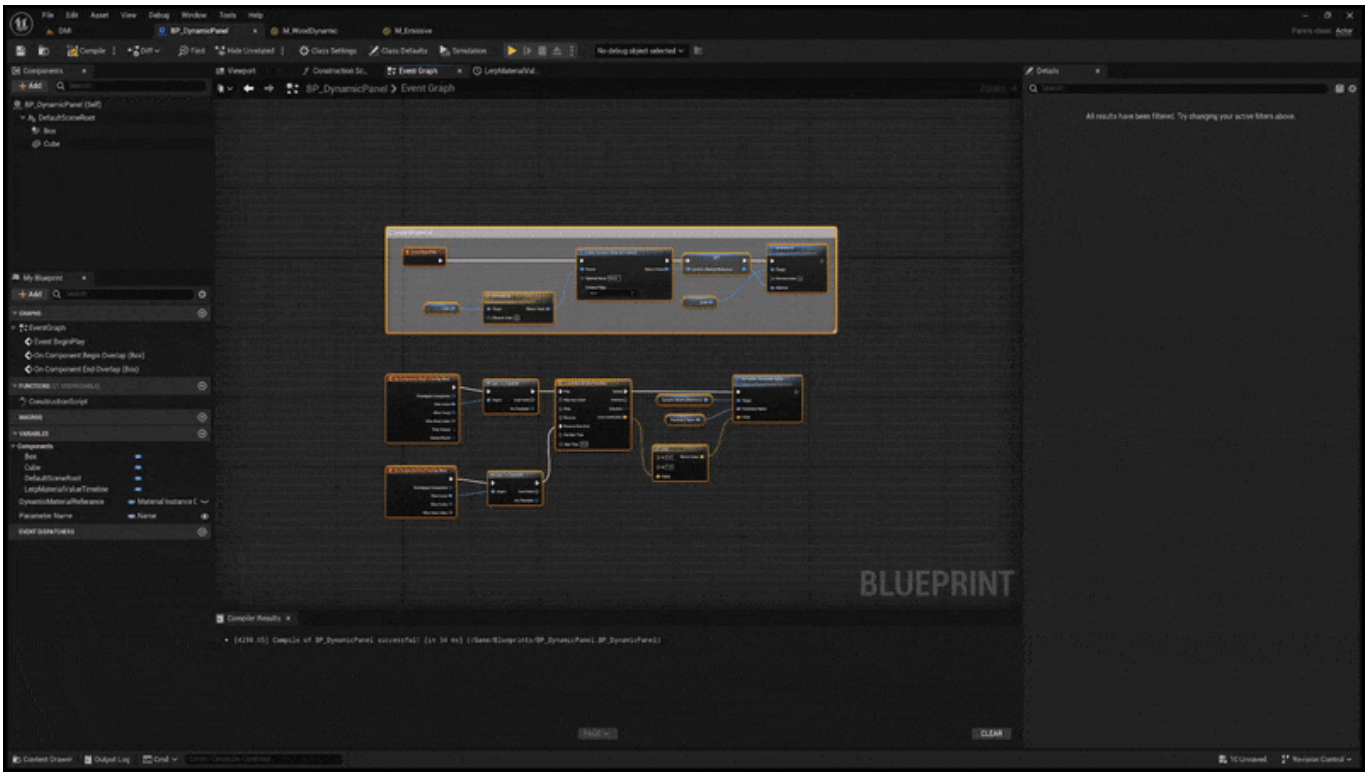


3. Dynamic Material Instance with Runtime Control

What is it?

A dynamic material instance (DMI) with runtime control refers to a material instance that allows developers to modify material parameters during the game's runtime. Unlike regular material instances, dynamic material instances enable real-time changes to properties such as colour, texture, scalar values and vector values through blueprints or code.
Both dynamic material instances and material instance constants can both be great for different scenarios instead of just using one or the other. Material instance constants are best used when you need a fixed material that doesn't change parameters during gameplay. The benefits of using this material type is that it is efficient for performance as these materials are precompiled and don't affect runtime overhead. Dynamic material instances are best used for materials that change during gameplay, such as damage effects. The benefit of using a DMI is that the material will allow runtime modification, however this does slightly add more performance overhead compared to material instance constants.
One ways to optimise DMI creation and updates is by minimising parameter updates by only updating the material parameters when they actually need to change, rather then every frame or unnecessarily. Doing this will lower processing load and rendering performance impact. Another way is by reusing shared DMI whenever possible on objects that share the same dynamic effects. Doing this decreases memory usage and runtime cost. A third way is by using performance profiling tools to locate bottlenecks in material update calls and optimising based on game constraints.
Gameplay scenarios that benefit from runtime material modification includes: damage feedback, customisation, animated effects, user interaction, environmental effects, user interaction and game event feedback.

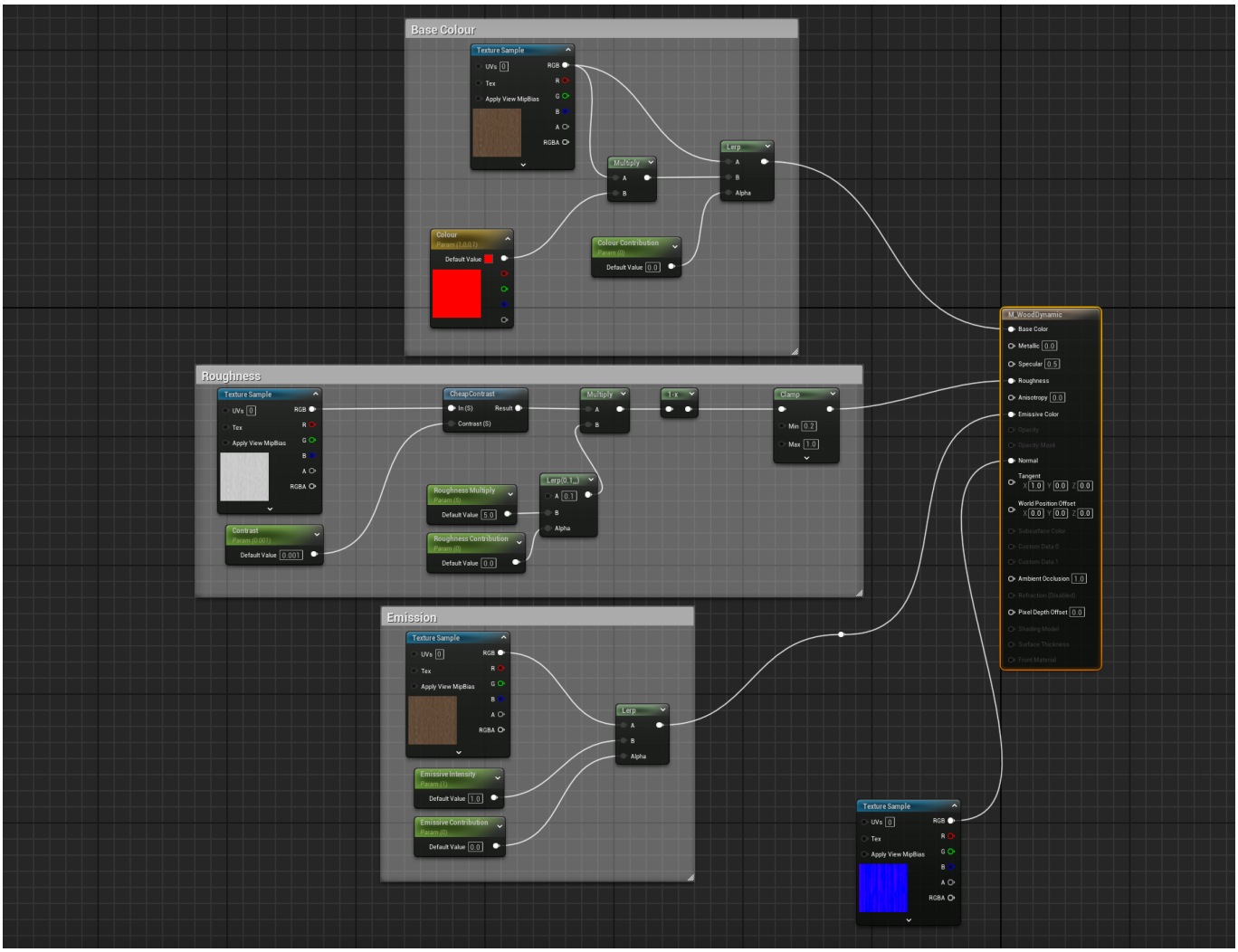
Final result



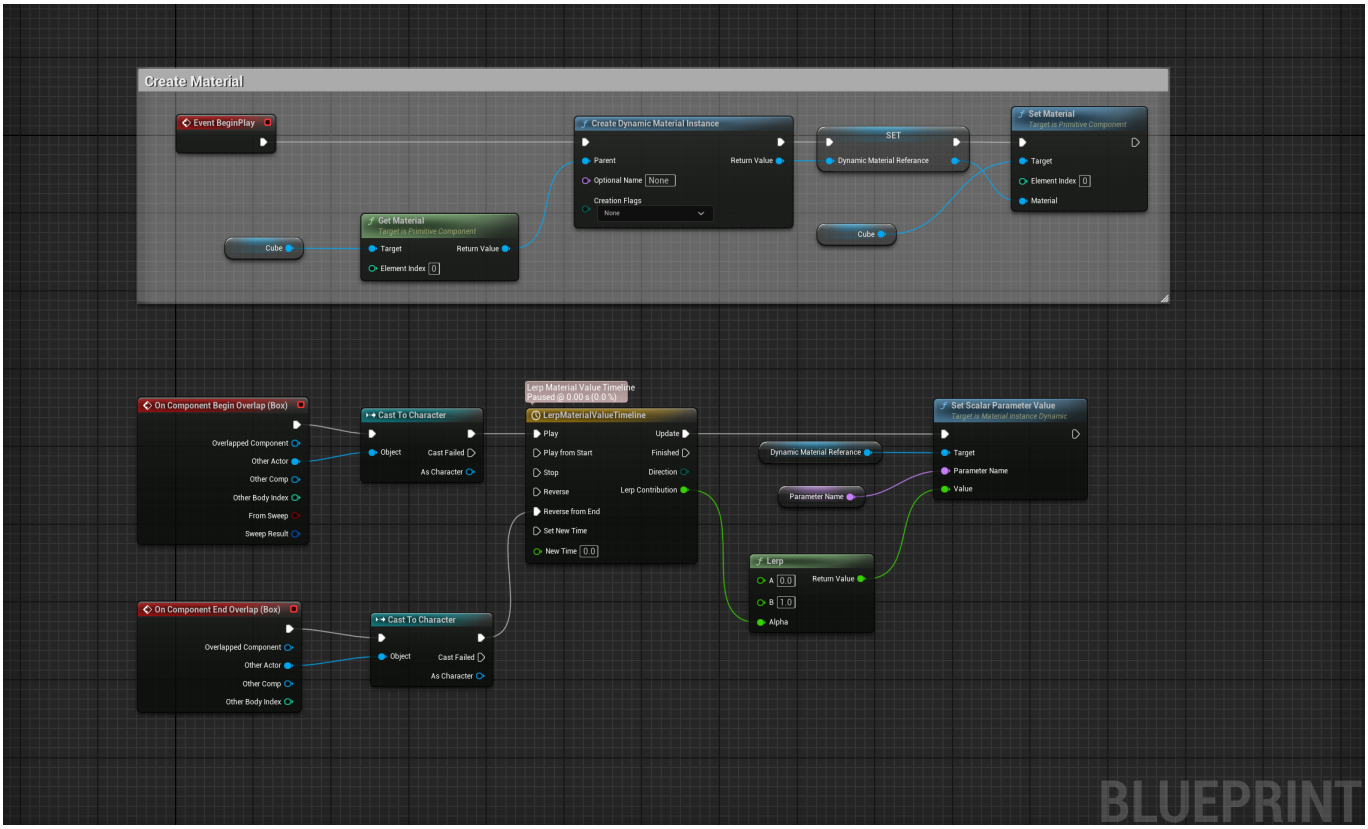
Dynamic Material Instance Showcase

Blueprints

Wood Material



Dynamic Panel



4. Parallax Occlusion Mapping (POM) Material

What is it?

A parallax occlusion mapping (POM) material is a type of shader material technique used in 3D rendering to create the illusion of complex 3D surface detail on flat textures without added geometric complexity. Height maps are used to simulate depth and 3D surface variation relative to the viewer's perspective.
The step count in POM significantly affects both visual and quality performance. Increasing the step count improves the visual quality of the parallax effect by providing finer sampling of the height map along the view ray. Doing this reduces the amount of stair stepping or slicing effects seen at lower step counts, creating more realistic and continuous depth illusion. A higher step count however increases GPU workload as the shader performs more samples per pixel, which directly impacts rendering performance, making the material more expensive to compute.
The surfaces that benefit most from parallax occlusion mapping are surfaces with distinct height variations and depth. For example, surfaces with tiles, rough surfaces and patterned surfaces all are surfaces with distinct height variations that will look much more realistic if made using parallax occlusion mapping.
An alternative to POM is bump offset, which does the same job but with worse visuals and better performance. Bump offset adjusts texture coordinates based on the view angle and height map but only uses a single sample, giving a basic illusion of depth which tends to bread down or look flat at sharp angles. Whereas POM performs multiple height map samples to more accurately create an illusion of depth. The positive to bump offset is that it is much cheaper to render as it uses fewer texture samples. As POM uses a more iterative sampling method, it is more GPU intensive.
When it comes to silhouette edges, POM cannot alter the the actual geometry silhouette of the mesh creating issues. One of these issues is that at extreme glancing angles, the outline of objects can reveal the underlying flat geometry, breaking the illusion of depth that POM creates. Another issue is that as the parallax effect happens only within the surface are of the polygon, viewing the edges or ends of teh mesh will expose that the depth effect is simulated.

Problems during creation

1. A glitch occurred with the material where the shadows took over the whole material so that it could'nt be seen.

Solutions

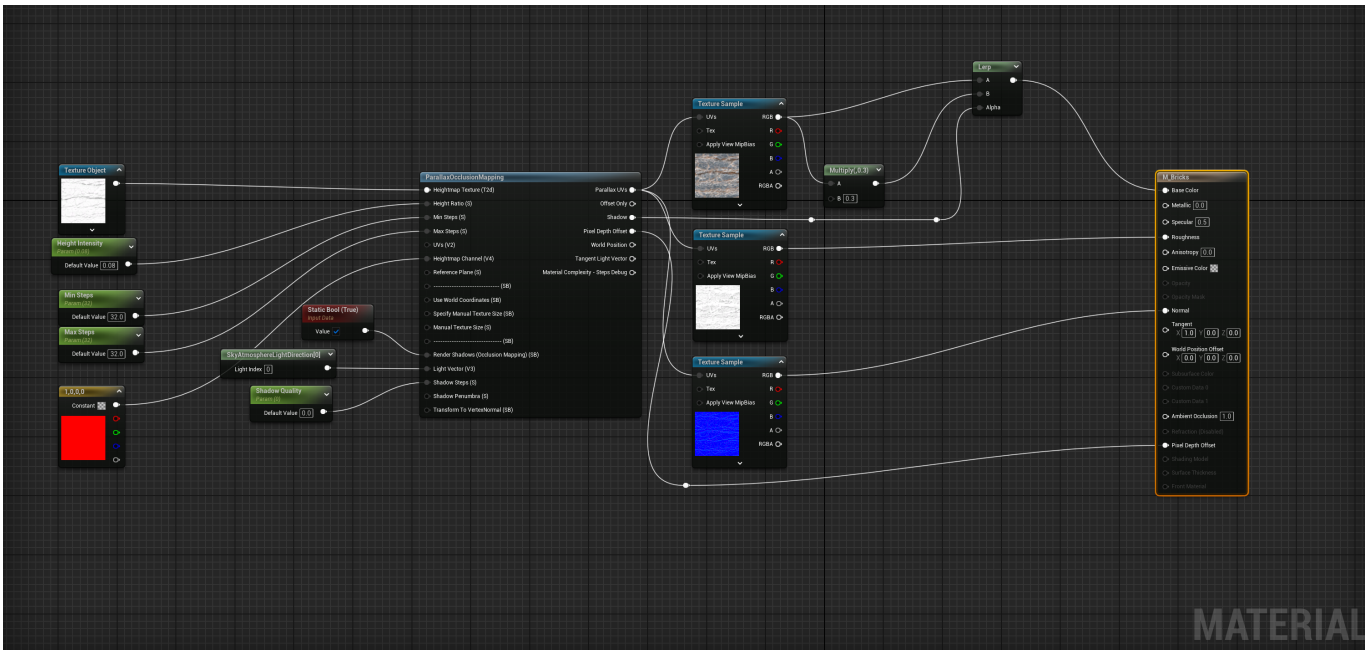
2. To solve this problem, I disabled cast shadows.

Final result

Brick POM Material

Blueprints

Brick Texture



5. Post Process Material

What is it?

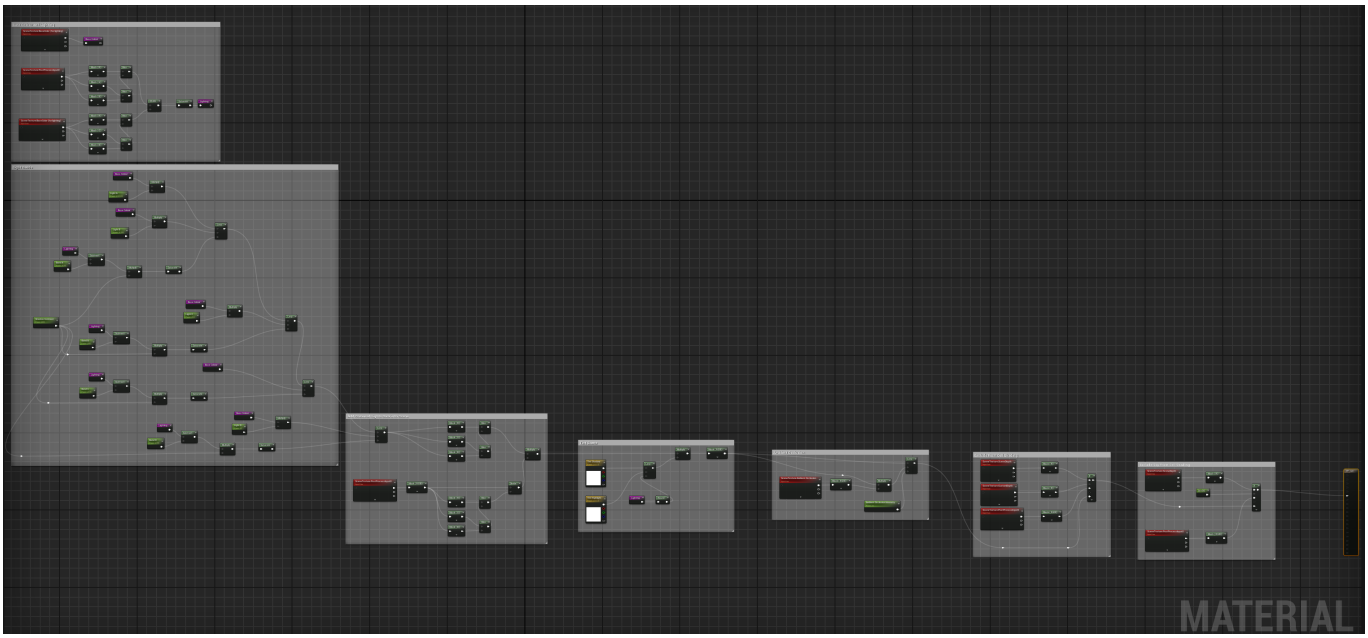
A post process material is a type of material that applies visual effects to the entire rendered scene after the main rendering process has finished. This is used to apply effects like colour grading, bloom, depth of field, motion blur or screen tinting to the entire scene.

Final result

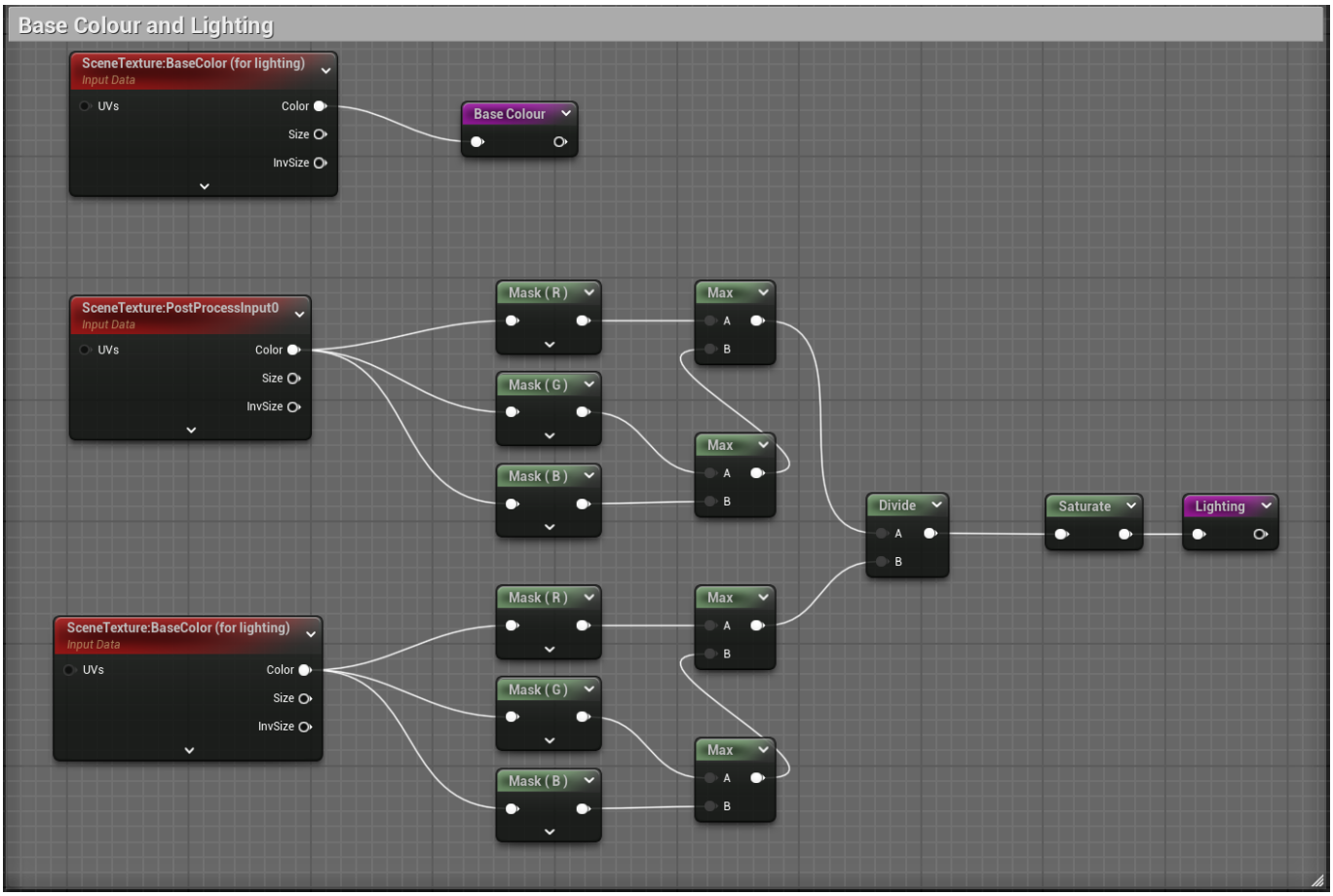
Cell Shading Showcase

Blueprints

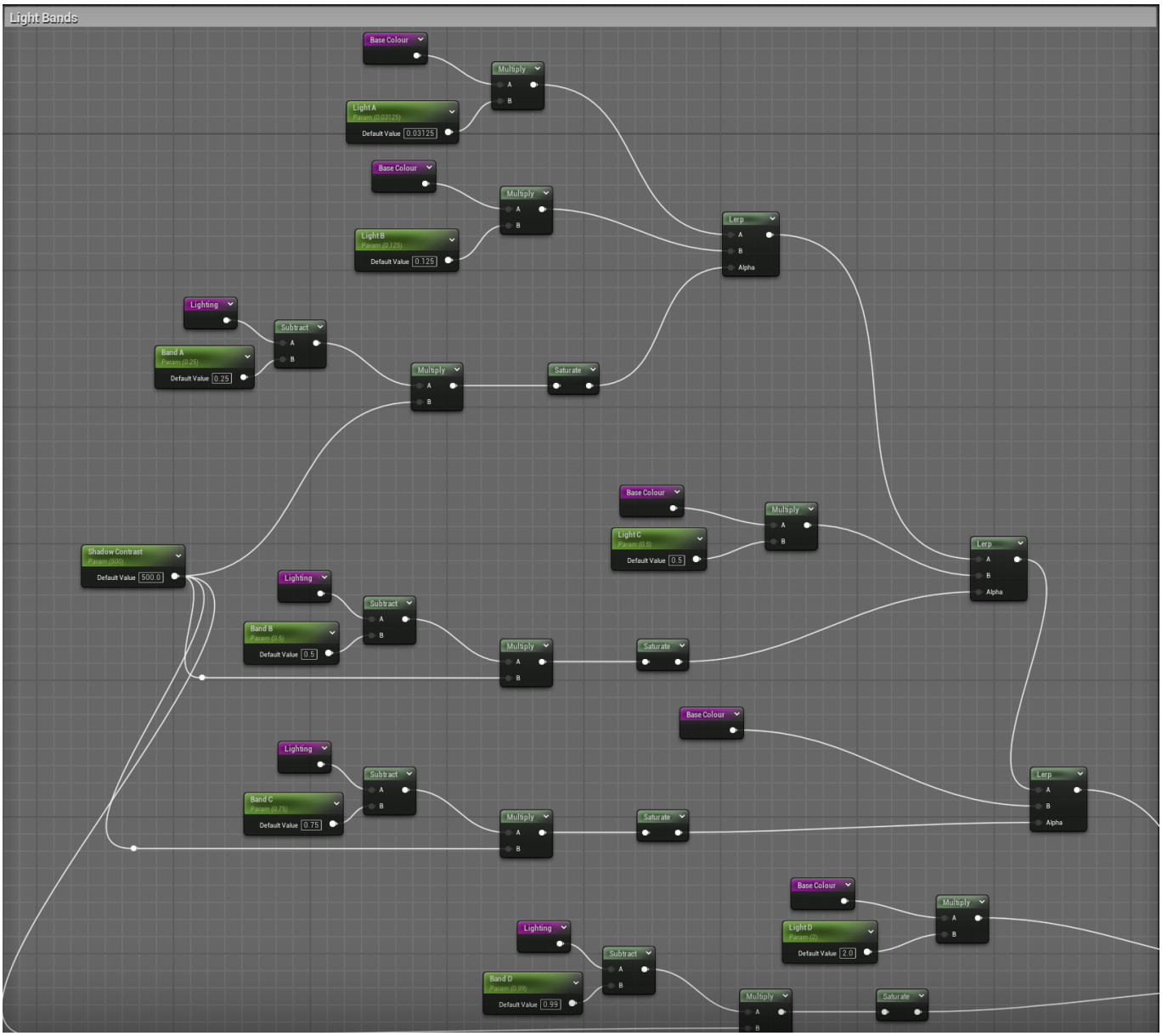
Cell Shader



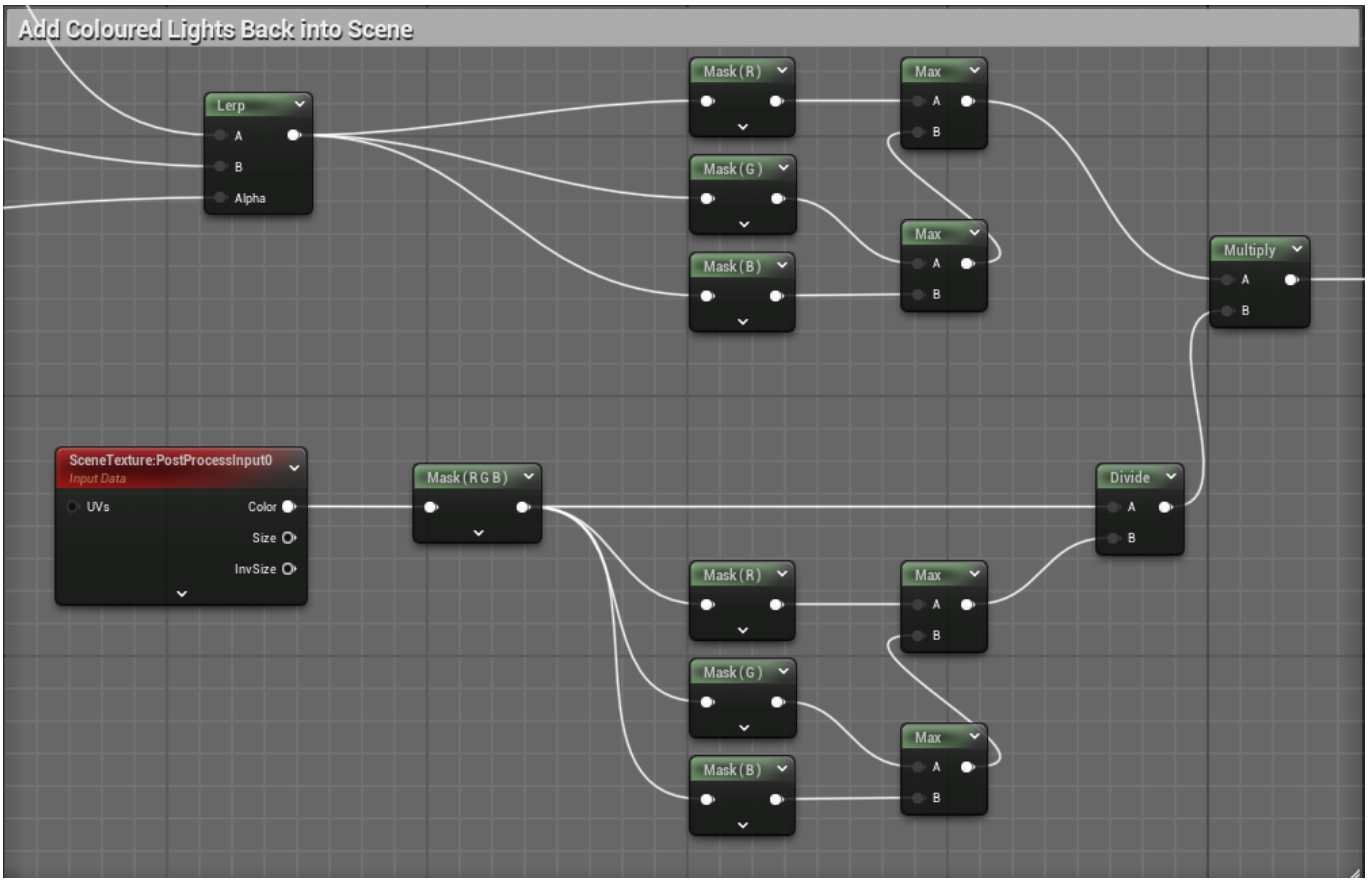
Cell Shading Base Colour and Lighting



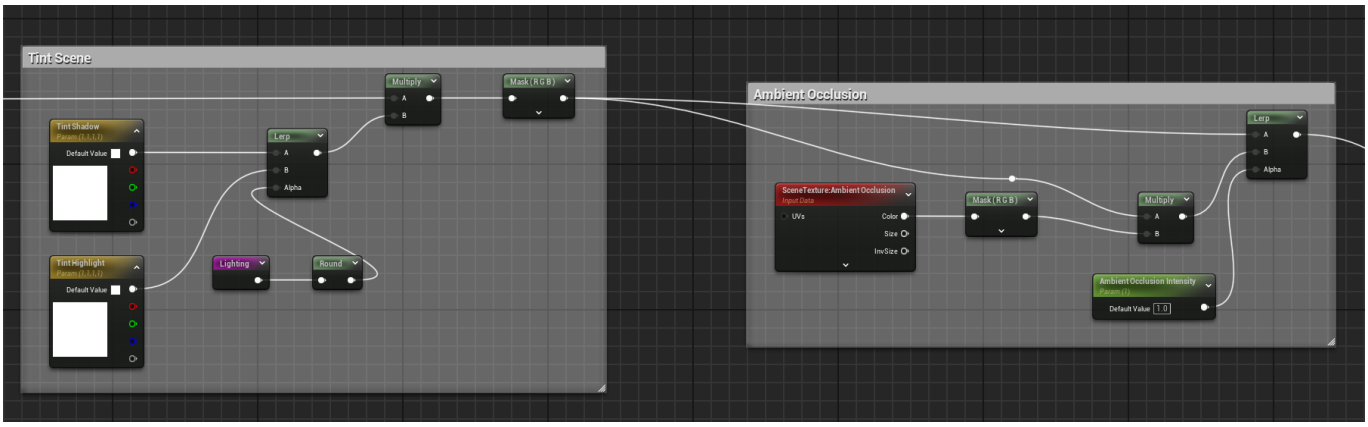
Cell Shading Light Bands



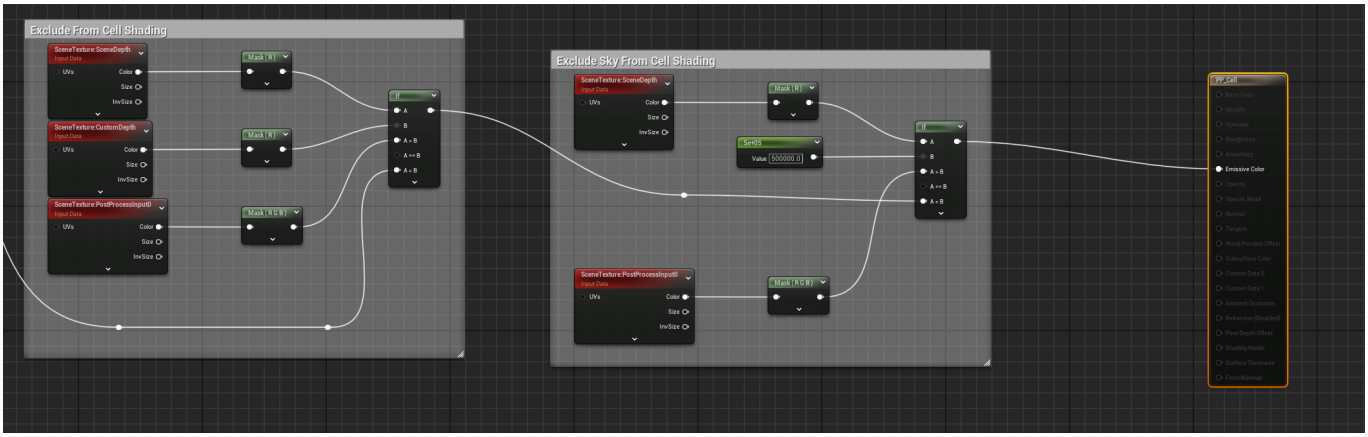
Cell Shading Adding Coloured Lights Back



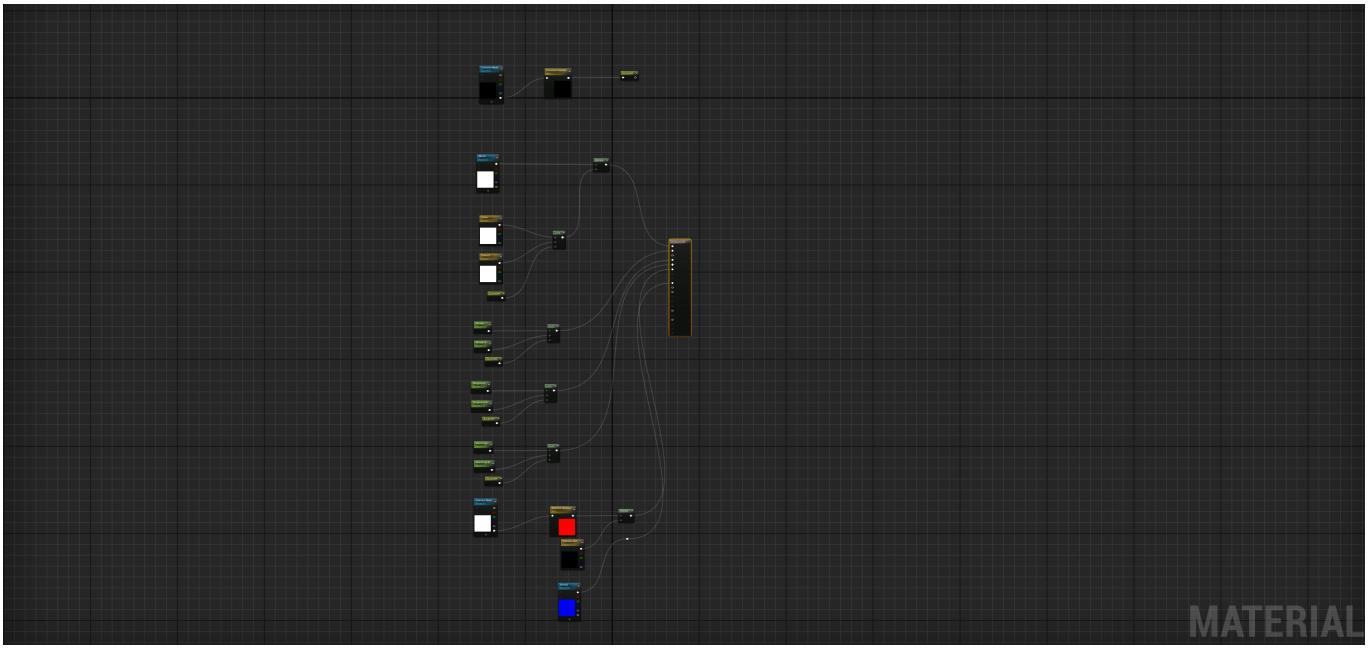
Cell Shading Tint Scene and Ambient Occlusion

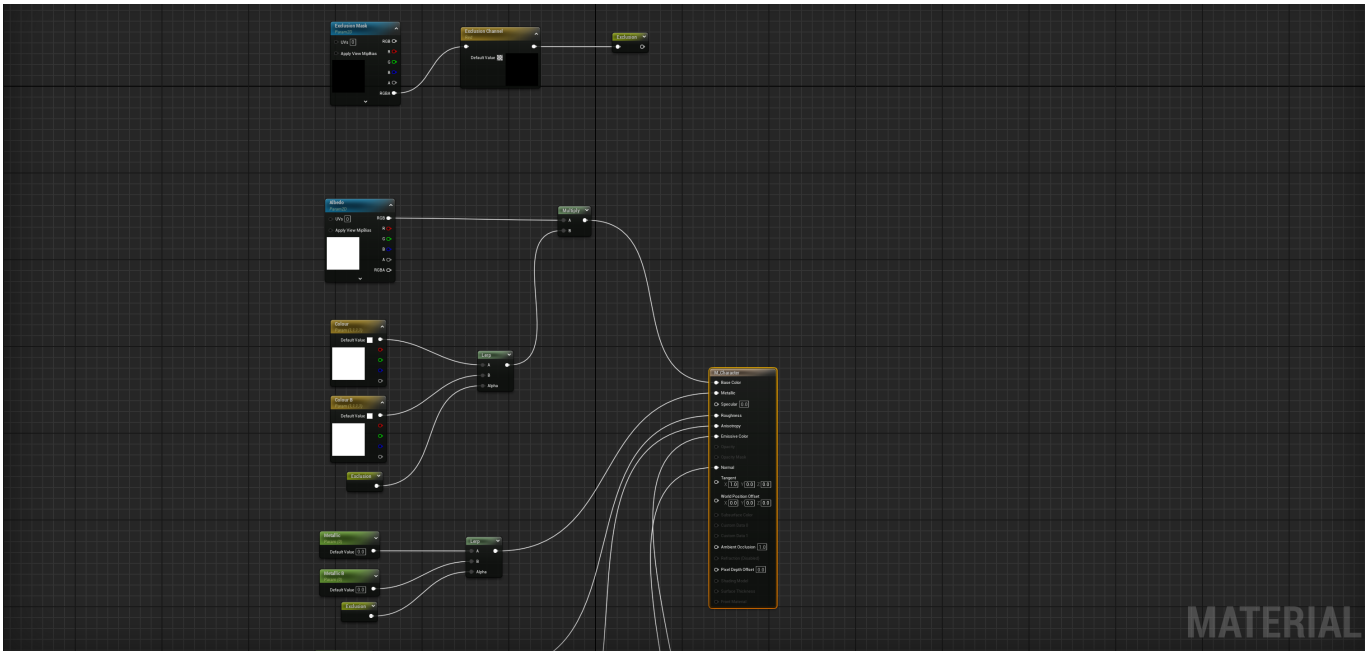


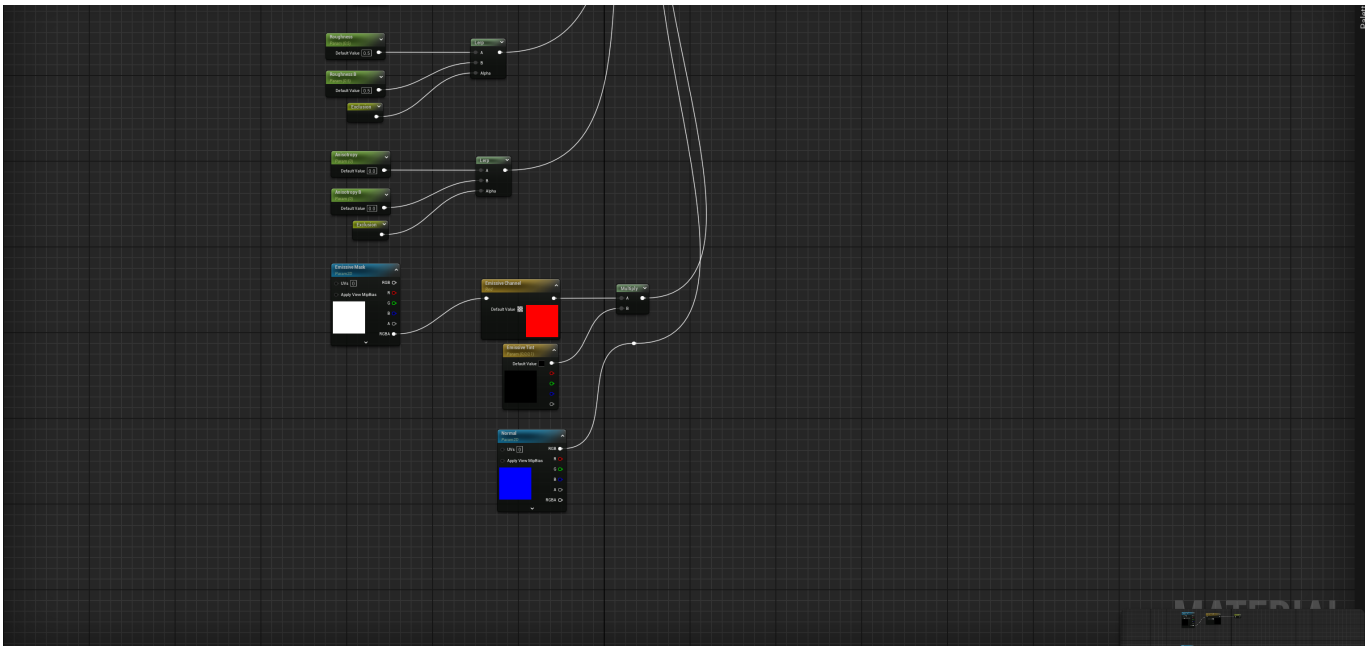
Cell Shading Exclusions



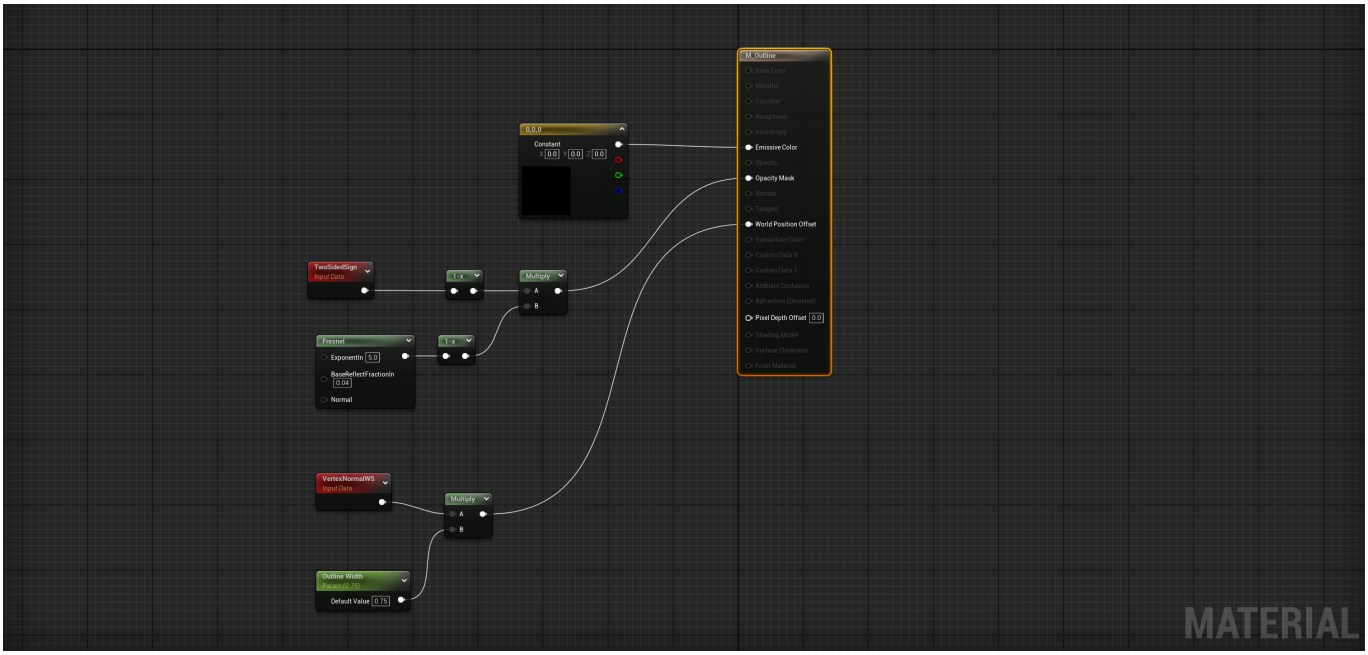
Character Material



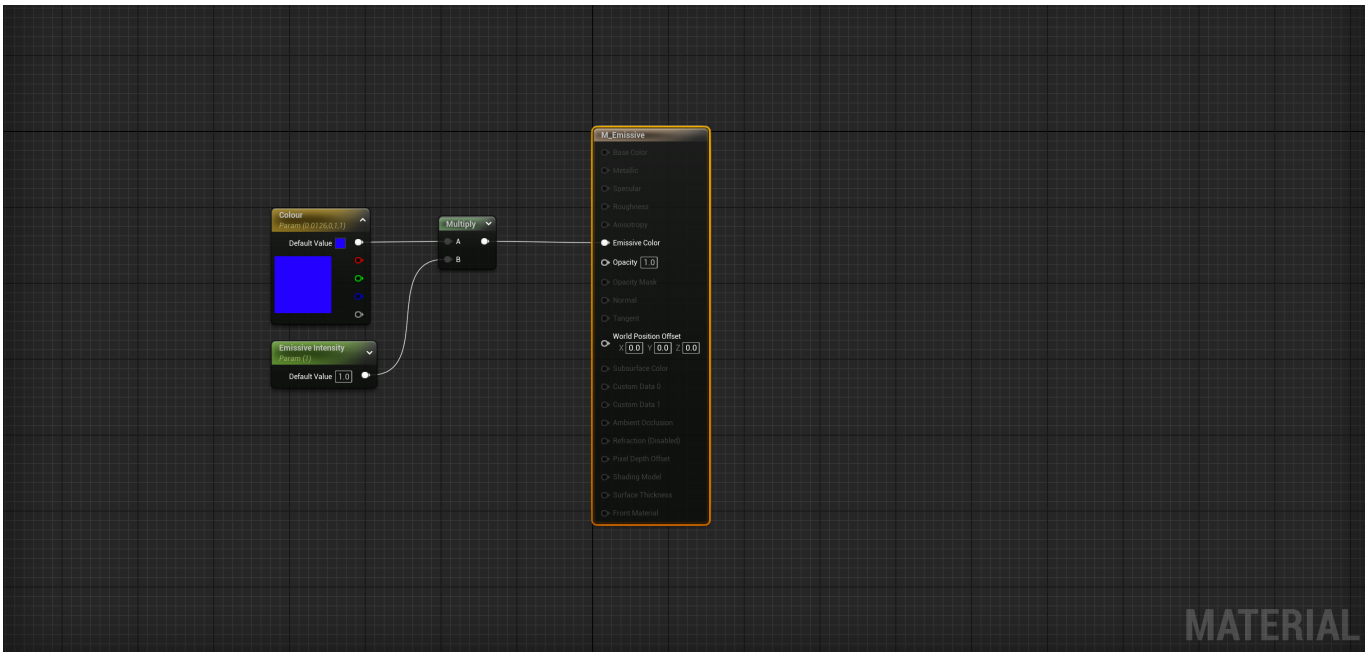




Character Outline



Emissive Material



Bibliography

- Stylized Water Material Tutorial in UE (2024) Directed by Viktoriia Zavhorodnia (akbutea). At: https://www.youtube.com/watch?v=bIEB85ITmEo (Accessed 28/10/2025).
- Dynamic Material Instances - Change Values on Materials Using Blueprint: UE5 Beginner Tutorial - YouTube (s.d.) At: https://www.youtube.com/watch?v=7S8FdUPpxUw (Accessed 28/10/2025).
- Parallax Occlusion Mapping In Unreal Engine 5 | UE5 Beginner Tutorial (2023) Directed by pinkpocketTV. At: https://www.youtube.com/watch?v=WNXPaAnxlQQ (Accessed 28/10/2025).
- How To Create The Ultimate Post Process Cel Shader - Unreal Engine 5 Material Tutorial (2025) Directed by Pitchfork Academy. At: https://www.youtube.com/watch?v=AvWx85Y9LRs (Accessed 28/10/2025).