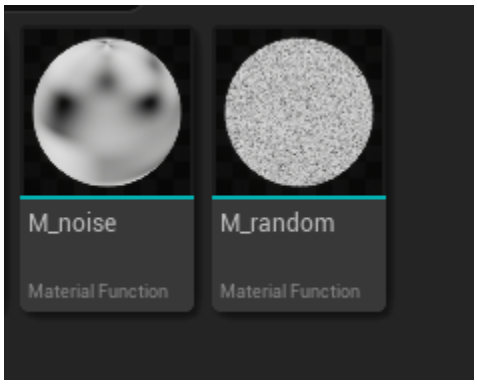
Zachary Adams
Imgd-4099-2023
/FINAL_PROJECT

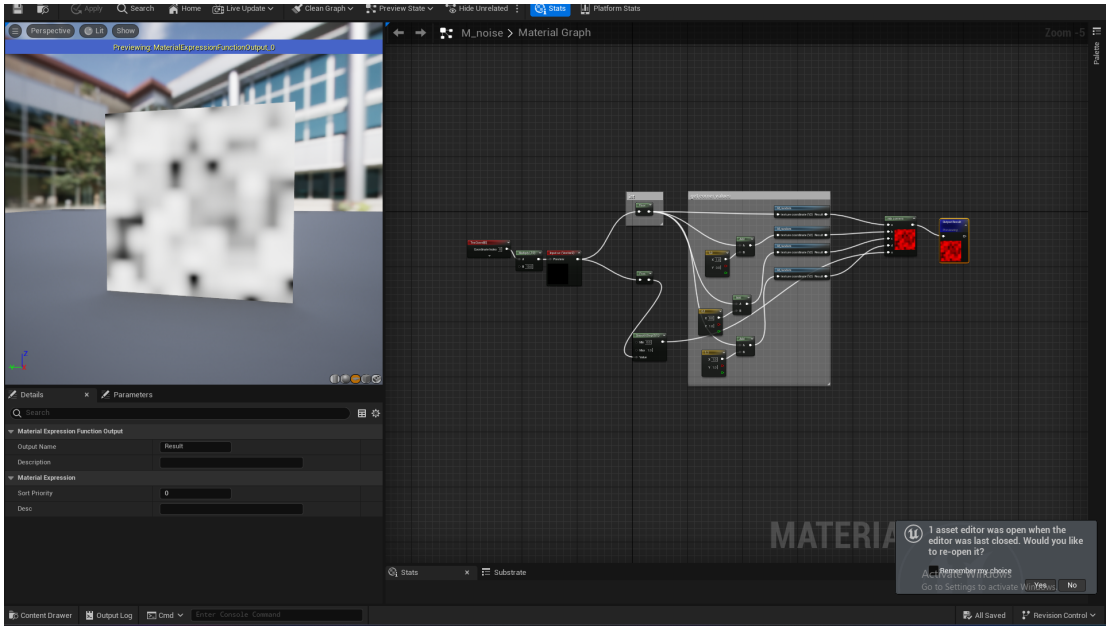
I wanted to learn how to implement the material from this class on fragment shaders in Unreal Engine by applying the logic from our WebGPU assignment to HLSL and Material Graphs. I also wanted the shader to visualize live audio from a music track or instrument. I was aiming to use organic shapes and patterns. A limited palette of colors focused on cool hues on a black background.

I read some articles on Frequency spectrum and EQ for sound production to understand what instruments typically take up what music ranges to decide how I would represent these concepts visually. low: primary shapes and general speed of flow. Mids: Character Color and Luminosity. Highs: Crispness and Definition

I implemented FFT analysis with Unreal's Blueprint nodes: I used a **Start Analyzing Output** node to start the FFT and **Get Magnitude of Frequencies,** which accepts an array of frequencies and returns an array of magnitudes associated with the given frequency. I used those magnitude values to set specific parameters for my custom material in real time.

Implementing the shaders was straightforward. I used HLSL code and Material Functions to make random, noise, and fractional Brownian motion materials. I used the book of Shaders and my previous assignments for this part.

- 



Because the shader drives the geometry, it was necessary to create geometry with proper scale, enough subdivisions, and proper UV coordinates, which is simple for a plane.

I created the functionality in my Audio Visualizer Blueprint object. Which was composed of an **Audio Capture Component** and **A Static Mesh Component** (the subdivided plane geometry.) With the material applied to the geometry, the material could change based on the frequency values of the audio.

The magnitudes were very sensitive to the audio changes and updated too fast, which I couldn't figure out. It was difficult for me to get smooth reactions. I struggled to get the behavior I wanted from the frequencies in the blueprint and to abandon implementing mids or highs as I had planned.