2020182031 이서연 졸업작품 12주차 진행

1. OMSetRenderTargets함수의 구조 수정
2. Debug Shader

저번 주에 Mipmap관련하여 수정을 한 부분의 설명이 부족한 것 같아 추가설명을 적어보려 한다.

Texture의 크기가 1:1에서 1:2인 절반이 될 때, LOD가 1 오르게 된다.

즉, 1:1 ~ 1:2 사이의 크기에 있으면 LOD 0과 LOD 1결과를 비율적으로 계산하여 결과가 나오게 되는데, 나 같은 경우 그래서 Vertex 범위를 -1~+1에서 -1~0으로 줄일수록 점점 화면이 어두워지고, 0부터는 아예 검정색이 나오는 것을 확인할 수 있었다.

그 이유는 Mipmap이 어떻게 설정이 되는지는 모르겠지만, LOD 1의 결과값이 들어가지 않고 있기 때문에 검정색 값이 섞여 들여가 비율에 따라 점점 어두워지는 것 이였다.

이런 Mipmap을 왜 굳이 사용하나 생각이 들었지만 멀리 있는 물체의 해상도가 선명하면 노이즈가 발생하는 등의 Mipmap관련 문제들이 존재한다는 것을 기억해냈다.

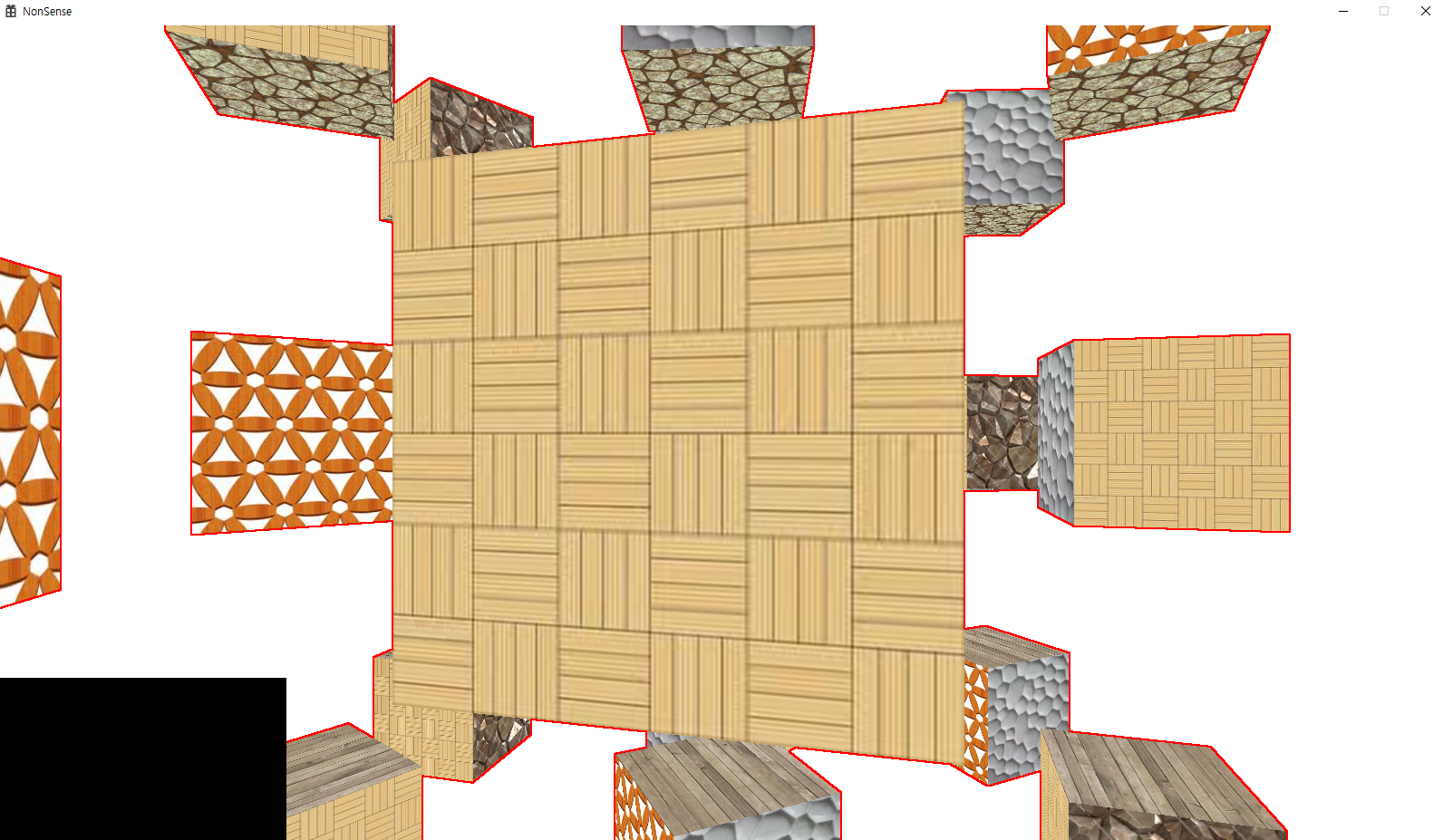
아무튼 이렇게 문제를 수정했으니, 이제 Debug Shader를 진짜로 만들어보려 한다.

1. Object ID개편 시도

현재 Object ID는 Object Shader에서 Constant Buffer로 Object하나 당 1씩 값을 올려서 넘겨주고 있다.

문들 이런 방법이 아니라 Shader코드에서 이를 처리할 수 있지 않을까? 하는 마음에 사용하던 Object ID를 지워 두고 SV\_InstanceID를 사용하여 Vertex Shader에서 Instance값을 넘겨주어 보았다.

내 머릿속에서 이 값은 각 객체마다 1씩 값이 올라가면서 전달되기에 내 생각대로 구현이 될 것이라고 생각했으나 결과는 달랐다.

가장 바깥 테두리에만 외곽선이 생김

왜 이렇게 다른 결과가 나올까? 생각해봤는데 아무래도 이 육면체들이 모두 같은 모델을 사용하는 Instancing 객체들이기 때문인 것 같다.

육면체 Vertex를 생성할 때 한번 Instance ID가 늘어나고, 그 이후 같은 값을 계속 사용하므로 하나의 Instance ID를 가지고 있다고 판단이 되어, 이 방법을 사용하지 않기로 하였다.