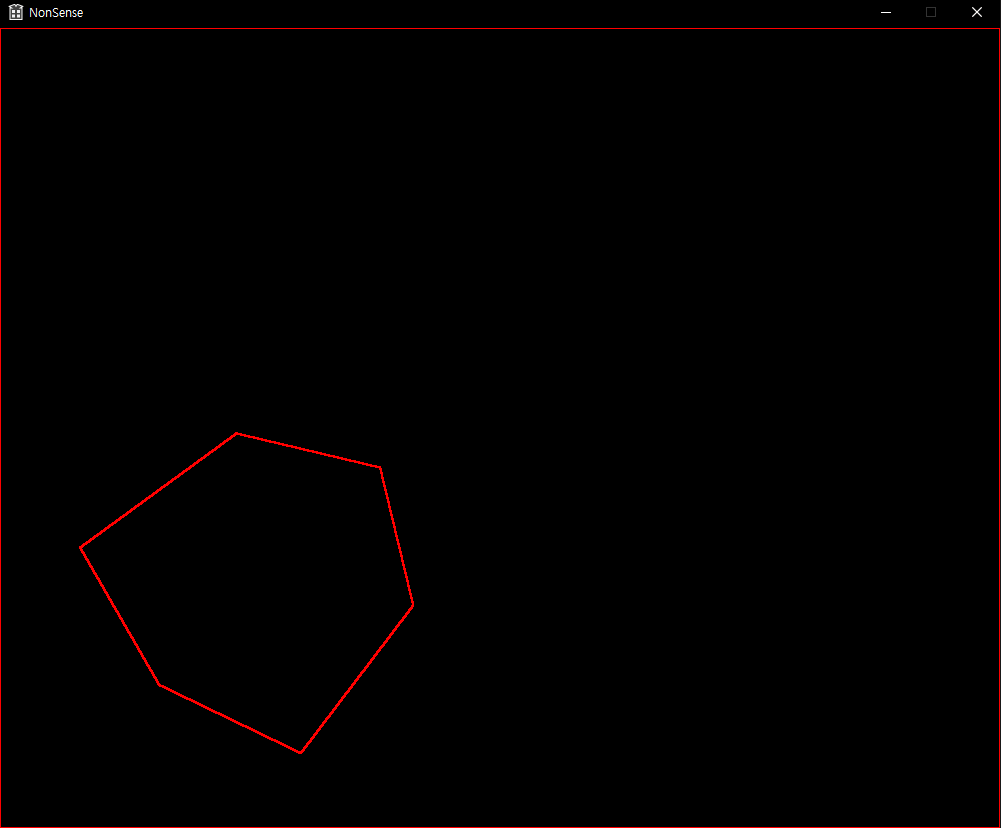
2020182031 이서연 졸업작품 9주차 진행

1. 외곽선

MRT부분을 뒤로하고 외곽선을 먼저 하려고 하는데 외곽선에 이상한 버그가 생김을 알 수 있었다.

일단 저번주에 내가 찾으려고 했던 Normal데이터에서 Alpha값을 읽어 Object ID를 불러오는 방법은 생각보다 쉬웠다.

그냥 Texture를 Position으로 배열로 만들어서 그 좌표의 Alpha값을 입력해주면 되는 것 이였다.

이것이 현재 검정 배경에 외곽선만 그렸을 때의 상황이다.

확인이 잘되게 하기 위하여 배경은 검정색에 외곽선 두께9, 빨간색으로 그려주었다.

육면체의 외곽선이 잘 따지는 것을 눈으로 확인할 수 있으나, 분명 육면체가 많은 코드인데 하나만 그려지고, 화면 전체에 빨간 테두리가 생기는 것을 알 수 있다.

이건 Object ID가 잘 들어가지 않고 있다고 생각이 들었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

여기가 바로 Object ID와 재질 등 Object정보를 넘겨주는 곳이다.

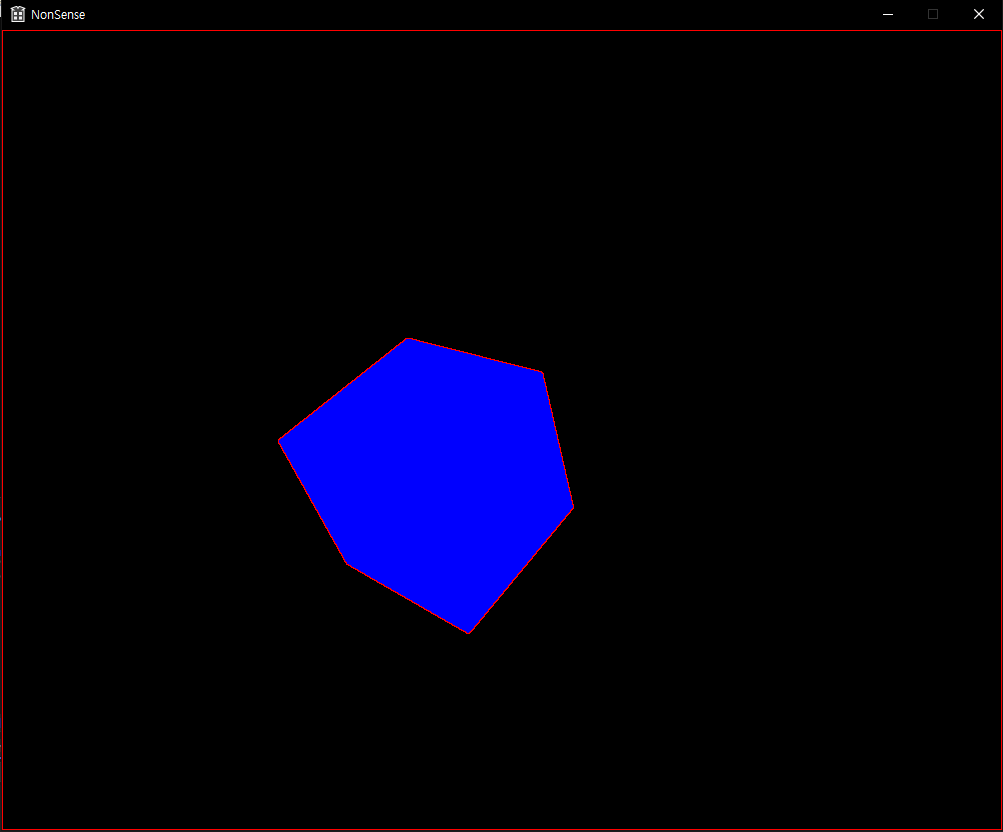
여태껏 이 코드를 사용해서 그냥 두고 있긴 했는데 예전부터 이 부분에 대하여 의문점을 가지긴 하였다.

과연 이 코드가 Constant Buffer로 데이터를 잘 넘겨주고 있는 걸까? 값을 넣어주고 있는 곳은 지역변수인 pbMappedcbGameObject이다. 이 변수는 함수내에서 바뀌기만 할 뿐 어디로 값을 보내주고 있지 않을 뿐 더러 Constant Buffer View와도 연결되어 있지 않다.

하지만 이상한 점은 이 코드의 Object ID를 넣어주는 부분을 주석 처리하면 정상적인 결과는 나오지 않지만 결과가 달라지는 것을 확인할 수 있었다.

내 생각대로라면 이 코드는 아무 영향도 끼치지 않고 있어야 하는데 영향을 끼치고 있다는 것이다.

또한 작동이 잘 되는 교수님의 코드에서도 이와 같은 코드를 사용하고 있다.

Object ID가 0인 부분을 파란색으로 Return

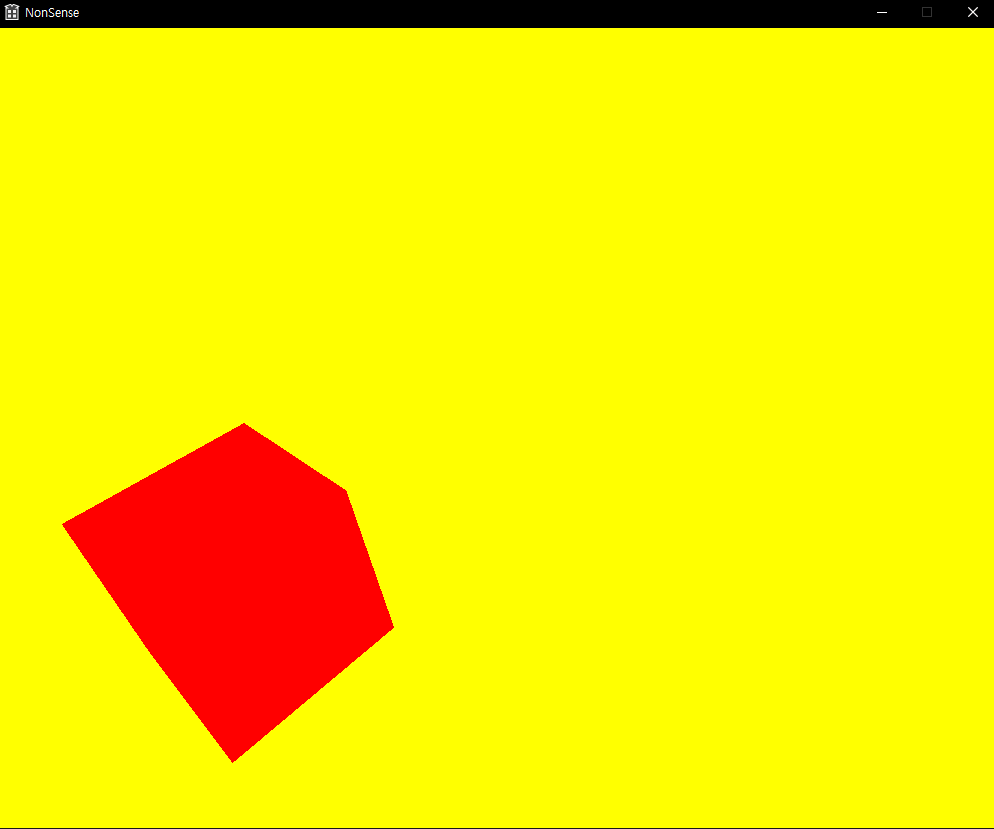
Object ID가 정말 들어가지 않고 있나 확인해보기 위하여 Pixel Shader에서 Object ID가 1보다 작은 부분을 파란색으로 Return해보니 저 부분만 파랗게 변했다.

또한 1보다 작은 부분을 파란색으로 Return해도 같은 결과가 나온다.

즉, 다른 부분들의 Object ID는 NULL값이 들어가서 알 수 없는 큰 수가 들어가고 있고, 첫번째 육면체인 저 육면체만 0으로 잘 들어가고 있다는 뜻 이였다.

하지만 교수님의 코드에서는 저 코드만으로 잘 돌아가고 있는 것이 참 이상했다.

이 부분만 해결하면 외곽선은 완성이 될 것 같다.

Object ID가 0일 땐 빨간색, 1일 땐 노란색을 출력하게 해보았다.

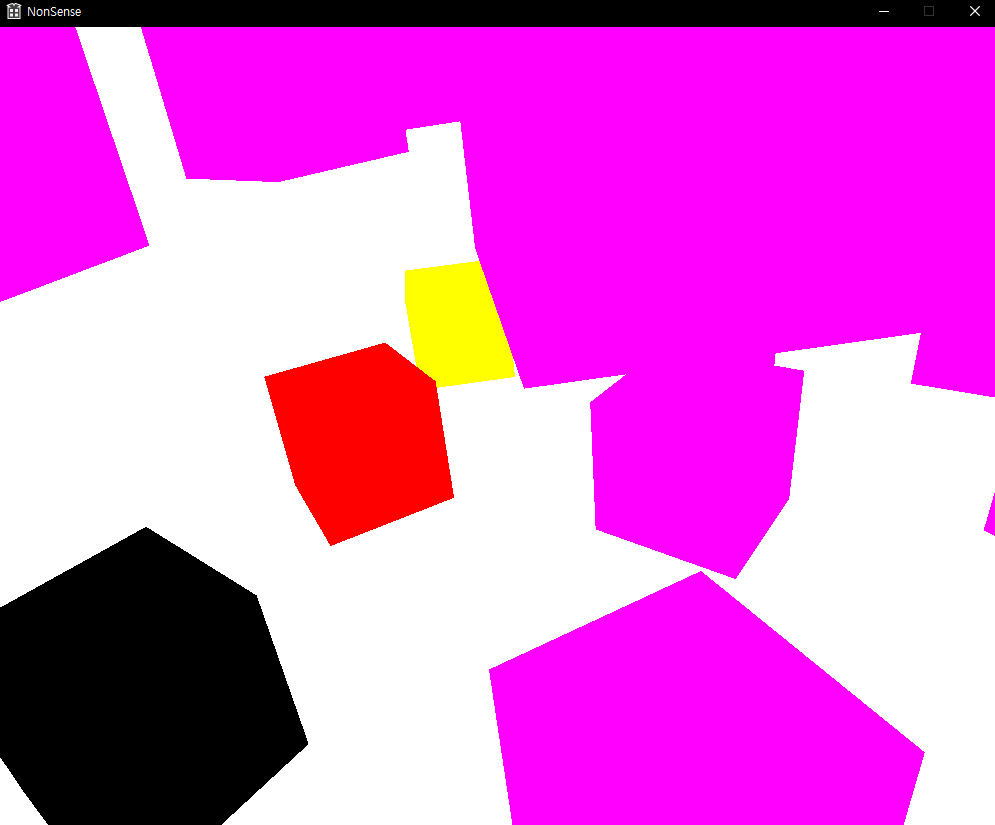
NULL값이 들어가고 있다고 생각했는데 그 외의 값들이 1로 들어가고 있었다.

NULL값이 아니라는 것은 값이 들어가긴 들어가고 있다는 뜻인데, 어째서 1이라는 숫자가 가득 들어있는지 모르겠다. 그래서 화면에 테두리가 생기는 것 같기도 하다.

Constant Buffer에 잘 들어가고 있는지를 확인하기 위하여 MRT로 바꾸기 전에

텍스트이(가) 표시된 사진

자동 생성된 설명이와 같이 위와 같은 방법으로 TARGET0인 Scene을 object ID에 따라서 색이 바뀌게 해보았다.

이게 그 결과이다.

Object ID가 잘 들어가고 있음을 알 수 있는데 그렇다면 왜 Texture로 변경 시 0과 1만 들어가는 걸까?

UINT인 Object ID와 Texture에 옮겨준 Object ID의 차이를 생각해 보았는데 float4인 RGBA로 값을 넣어주고 있었기 때문에 Normal Texture의 Alpha값으로 들어가는 Object ID가 자동으로 0~1로 바뀌는 건가? 하는 생각이 들었다.

그래서 Object ID값이 아닌 1/(Object ID+1) <- (Object ID가 0이 있을 것을 고려)을 넣어주었다.

잠옷이(가) 표시된 사진

자동 생성된 설명그랬더니 다행히도 외곽선들이 나오기 시작했다!!

그러나 왼쪽 아래에 있는 Object의 외곽선이 출력 되지 않는 것을 확인할 수 있다.

왼쪽 아래에 외곽선이 출력되지 않는 이유는 Object들이 없는 곳의 Object ID가 왜인지는 모르겠지만 1로 채워져 있어서, 1/(Object ID+1)이 1이 되는 0번째 Object의 외곽선이 출력되지 않는 것 이였다.

그래서 +1이 아닌 +2를 하여 1/2부터 만들어주면 1과 겹칠 일이 안 생기니 그렇게 수정해주었다.

그리고 화면 테두리에도 외곽선이 그려지는 버그를 고쳐야만 한다.

아무래도 바탕의 Object ID와 게임 창을 넘어서는 어딘가까지 검사가 되는듯 하다.

이건 막코딩으로 해결했는데 Texture Position의 XY를 검사하여 0과 맨 끝부분은 외곽선을 그려주지 않는 방법을 사용하였다.

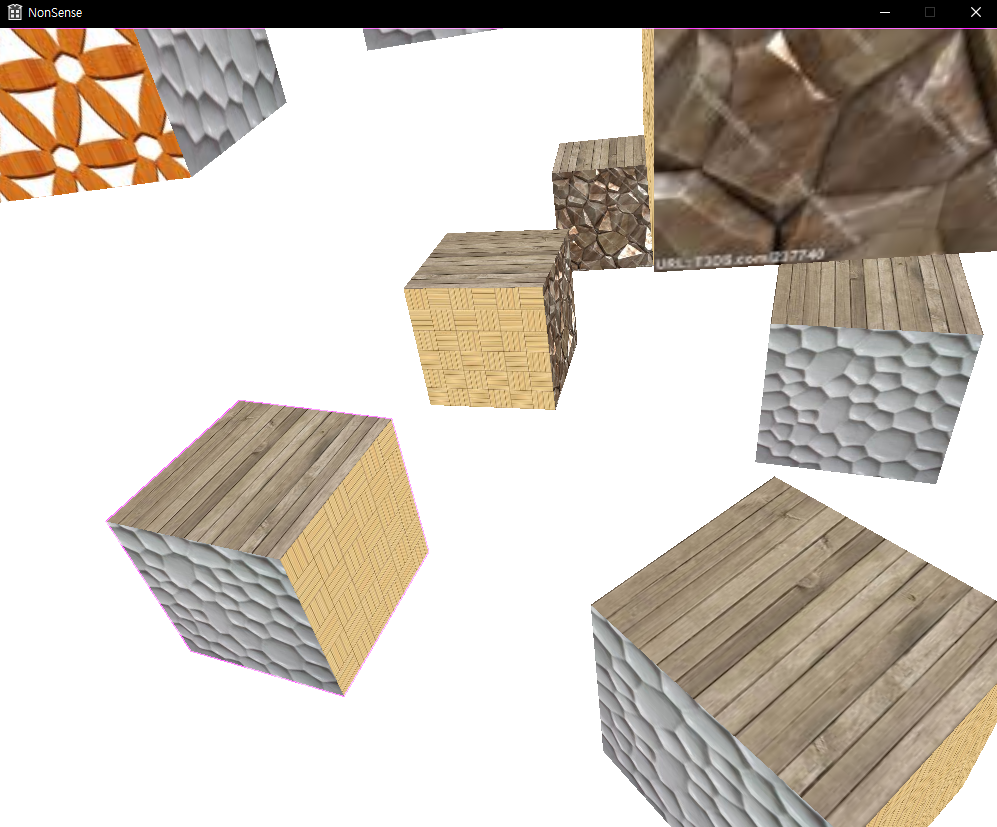
하지만 이 방법은 실제 외곽선이 드러나야 하는 부분도 화면의 끝부분에 있다면 출력되지 않는 문제점이 발생해서 다른 방법을 찾아야 했다.

혹시 화면 밖의 Object ID가 0이진 않을까? 싶어서 비교하는 두 Object ID중에 0이 있다면 외곽선 계산을 하지 않도록 하였다.

그랬더니 딱 원하던 결과가 나왔다! 외곽선 완성~

외곽선이 현재 출력 색상 + 외곽선 색상으로 적용되어, 무조건 더 밝아지고 색상이 합쳐지게 되어 배경이 밝을 때 외곽선이 보이지 않는 버그가 발생하였다.

그래서 Edge함수를 따로 분리하지 않고 Pixel Shader에 합쳐서 외곽선인지 체크를 한 후 외곽선이면 외곽선 색상을 출력하고 그렇지 않으면 원래 색상을 출력하도록 하였다.

수정 전 텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명수정 후

사진으로 비교해보니 더욱 큰 차이를 느낄 수 있었다. 수정 전에는 배경이 하얗기 때문에 사실상 Line Size가 9로 되어있었는데도 거의 1~3정도의 사이즈를 내고 있었다.

1. Deferred Lighting

전에 Position부분을 화면에 출력만 하고 정보를 넘겨주지 못하는 상태로 마무리가 지어졌다.

이 부분을 어떻게 해결해야 할까? 고민을 했는데

텍스트이(가) 표시된 사진

자동 생성된 설명 이와 같이 TARGET0가 아닌 TARGET1부터 설정하니 모든 정보가 넘어가게 되었는데 생각보다 쉽게 해결되었다

* 해결된 것이 아니었다.

CreateGraphicsPipelineState: The Pixel Shader expects a Render Target View bound to slot 3, but the PSO indicates that none will be bound.

The render target format in slot 3 does not match that specified by the current pipeline state.

이 두 오류가 발생하고 있었다.

물론 원하는 결과가 나오긴 하지만 이런 오류들은 잡아주어야 하기 때문에 알아보아야 한다.

TARGET0부분을 그려 주기만 할 뿐 넘겨주지 않는 문제이기 때문에 그냥 TARGET0를 Scene으로 설정하여 FLOAT4(0,0,0,0)이라는 의미 없는 값을 넣어주고, MRT수를 하나 늘려 그대로 전달해 주었더니 원하던 결과가 나오게 되었다.

1. Screen constant Buffer 추가

Screen을 출력할 때 Option(그래픽 옵션)들을 설정가능한 Constant Buffer를 추가해주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명이것이 Constant Buffer의 Data들이다.

Line Color는 외곽선의 색, Draw Options는 나중에 MRT를 따로 출력하기 위하여 어떤 종류의 Texture를 출력할 지 정해주는 것이다.

Line Size는 외곽선의 두께(0~9) 그리고 Toon Shading은 Toon Shading계수를 전달해 주기로 하였다.

Register(b5)로 Constant Buffer를 연결하여 외곽선을 출력하니 내가 넣어주는 정보대로 출력되는 것을 확인 가능했고, 이는 나중에 게임의 환경 설정에서 그래픽 설정이 가능하도록 할 것이다.