2020182031 이서연 졸업작품 11주차 진행

1. 버그 수정

저번 주에 SV\_TARGET0인 Scene을 추가했더니

ID3D12CommandList::ClearRenderTargetView: The clear values do not match those passed to resource creation. The clear operation is typically slower as a result; but will still clear to the desired value.

라는 오류가 발생했다고 생각했었다.

그러나 과거Commit으로 돌아간 시점에서 MRT수를 4로 늘리고 Scene을 추가해주었는데도 오류가 발생하지 않았다.

그것은 SV\_TARGET0을 추가해주어서 생긴 문제가 아니라는 것이다.

그렇게 과거의 코드와 비교하며 계속 코드를 수정해보다가 어이없는 이유를 발견했다.

바로 그것은 초기화 시켜주는 색에 따라서 오류가 발생하는 것 이였다.

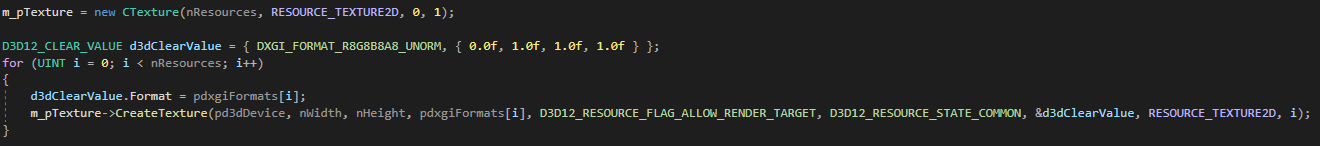
꼭 Float4(0.0f, 0.0f, 1.0f, 1.0f) 즉, 파란색이여야 오류가 발생하지 않고 있다.

Render Target View와 초기화 값이 호환이 되어야 한다는 점은 알고 있는데 현재 Render Target View의 Format은 DXGI\_FORMAT\_R8G8B8A8\_UNORM 을 사용하고 있다.

R, G, B, A를 각각8비트씩 사용하는 평범한 Texture Format인데 왜 파란색만 호환이 되는 건지 모르겠다.

일단 파란색으로 해서 이 버그는 해결해 두었고 이 문제에 대해선 천천히 알아보아야겠다.

Clear Value가 다른 이유를 알아냈다.



Screen Shader의 Texture를 생성하는 코드인데 여기서 Clear Value를 지정해준다.

지금은 확인을 위하여 수정이 되어있지만 원래는 파란색으로 되어있고, Texture를 만들 때 이 값을 넣어주기 때문에 이와 같은 값으로 초기화 해주어야 하는 것 이였다.

앞으로도 공통적인 Clear Value를 사용해주기 위하여 stdafx.cpp에 Clear Value를 추가하여 사용해 준다.

1. Debug Shader

Debug Shader는 Screen Shader와 같은 Texture정보를 쓰되, 리소스의 크기를 줄여줄 것이다.

그렇기에 Screen Shader와 같은 함수가 많으므로 Screen Shader의 자식 Class로 만들어 주었다.

하지만 사용하는 Pixel Shader가 다르므로 Create Pixel Shader함수, 그리고 Constant Buffer도 다른 것을 사용하므로 관련된 함수들은 따로 만들어 주었다.

Debug Shader의 Texture Resource는 Screen Shader와 같은 것을 사용하지만 Screen Shader는 화면 전체 크기의 리소스를 사용하는 반면에 Debug Shader는 Screen위에 여러 Debug화면을 띄워야 하므로 크기가 작아야 한다.

하지만 이미 만들어진 Resource의 크기를 바꾸어 주는 것은 불가능하므로 작은 사이즈의 Texture Resource를 다시 만들어 주려고 한다.



그러기 위해서 먼저 똑 같은 양의 Resource를 더 만들어줘야 하므로 Descriptor Heap의 Descriptor개수에 MRT개수만큼 더 늘려 주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그 후에 Screen Shader의 Resource를 다 만들어 준 후 Descriptor Handle을 MRT개수만큼 사이즈를 옮겨주었다.

그럼 Screen Shader의 Resource바로 뒤에 Debug Shader Resource가 오는 것이다.

나는 Debug가 약 4개가 있다면 화면의 상단에 화면 비율에 맞게 가로 세로가 MRT개수로 나눈 사이즈가 되도록 만들어주려고 한다.



그래서 이렇게 MRT (Depth포함)으로 나누어 주었다.

키보드, 벌집, 혼잡한이(가) 표시된 사진

자동 생성된 설명좌측 하단의 검정색 사각형이 Debug Shader부분이다.

확인을 해보니 Screen Shader에서 쓰는 Texture와 같은 Texture배열을 사용하는데 출력이 되지 않는 것을 볼 수 있다.

계속 구현하다 보니 내가 하고 있는 방법이 맞나? 라는 의문점이 들기 시작했다.

아무리 생각해도 화면 크기를 줄였다고 Pixel Shader값이 어두워지는 부분이 이해가 가지 않았고, 그래서 float4인 빨간색을 출력하도록 코드를 바꾸어 보았다.

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명색이 달라 진 것은 코드를 확인하다가 수정한 것

어두워지지 않고 쨍한 빨간색이 나오는 것을 확인 할 수 있는데, 이것은 결과적으로 Pixel Shader의 값이 어두워진 것이 아니라 그것을 채우는 Multi Render Target Texture의 색상이 어두워졌다는 것을 알 수 있었다.

상식적으로 Texture가 어두워지는 이유를 생각 할 수가 없었기에 Sampler를 확인해 보았다.

텍스트이(가) 표시된 사진

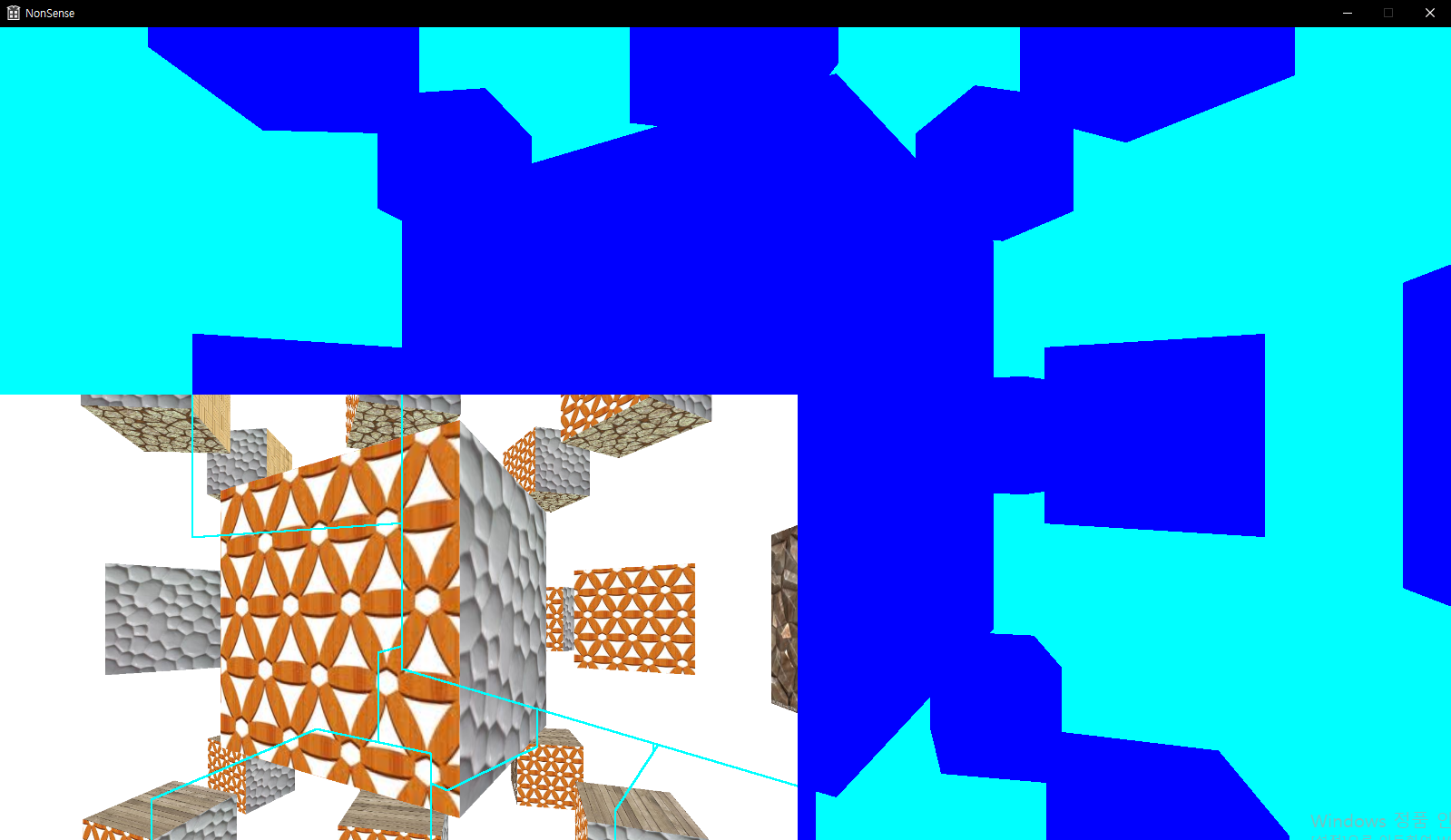
자동 생성된 설명

이 부분이 원래 Sampler부분이다.

아무리 보아도 Texture의 색을 어둡게 만드는 부분이 없다고 판단하여 넘어갈 뻔 했지만 Max LOD인 D3D12\_FLAOT32\_MAX를 0으로 바꾸자 정상적으로 작아지는 것을 확인 할 수 있었다.

저 수치는 LOD, 즉 최대 MipMap의 레벨을 정해주는 것인데 현재 매우 큰 값이 들어가 있다.

다음 사진이 0으로 고치고 나서의 상태이다.



다음주에 바로 Debug Shader를 만들 생각이다.

1. Animation Controller와 병합

Model과 Animation Controller 띄우기를 성공하였다고 하여 그 코드와 내 코드를 합치려고 한다.

현재 Model은 Standard Shader라고 하는 Shader로 출력이 되는데 이 방식은 Model의 여러가지 Map들을 Texture로 불러와, 최종 색상을 계산하여 Pixel Shader가 출력해주는 방식이다.

이것을 내가 구현한 Deferred Rendering방식으로 수정하려고 한다.텍스트, 장난감이(가) 표시된 사진

자동 생성된 설명

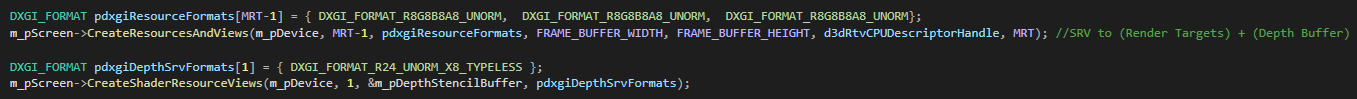
현재 출력되는 캐릭터의 모습인데 얼굴이 이상하게 보이는 것은 Texture의 문제이고 고칠 예정이다.

1. Depth값 버그

Depth값이 예전엔 나왔는데 어느 순간부터 나오지 않기 시작했다.

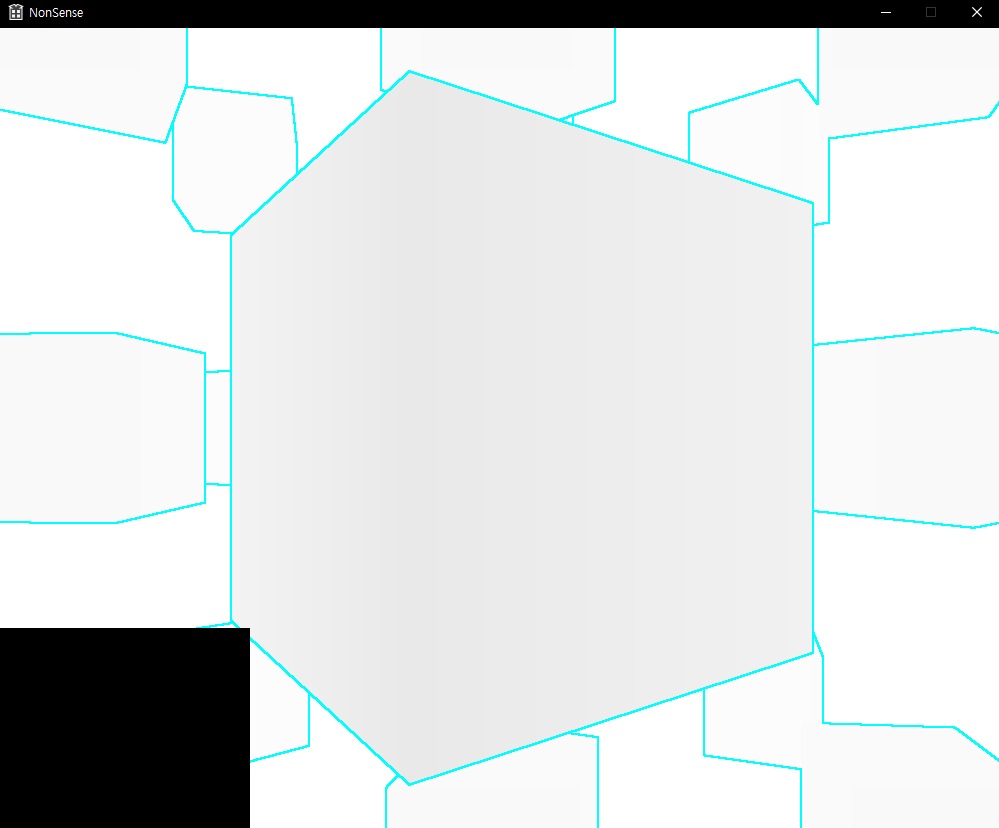
Depth값을 출력해주면 그냥 하얀 화면이 출력 될 뿐 이였는데 이 버그를 고치려고 한다.

내 추측 상 Texture의 마지막 배열에 Depth값이 온전히 들어가지 않는 것 같다는 생각이 들어서 코드를 계속 둘러보았다.



원인을 발견했다! 이 상태는 수정본이고 원래 Resource를 MRT-1이 아닌 MRT개 만큼 생성해 주었다.

그러나 나의 MRT는 Depth가 포함된 수 이므로 MRT-1만큼 리소스를 만들고 MRT만큼 Descriptor Heap을 만들어줘야 했는데 착각을 하고 하나씩 더 많이 만들어 준 것 이였다.

이것이 수정을 하고 Depth값을 출력해 준 것이다.

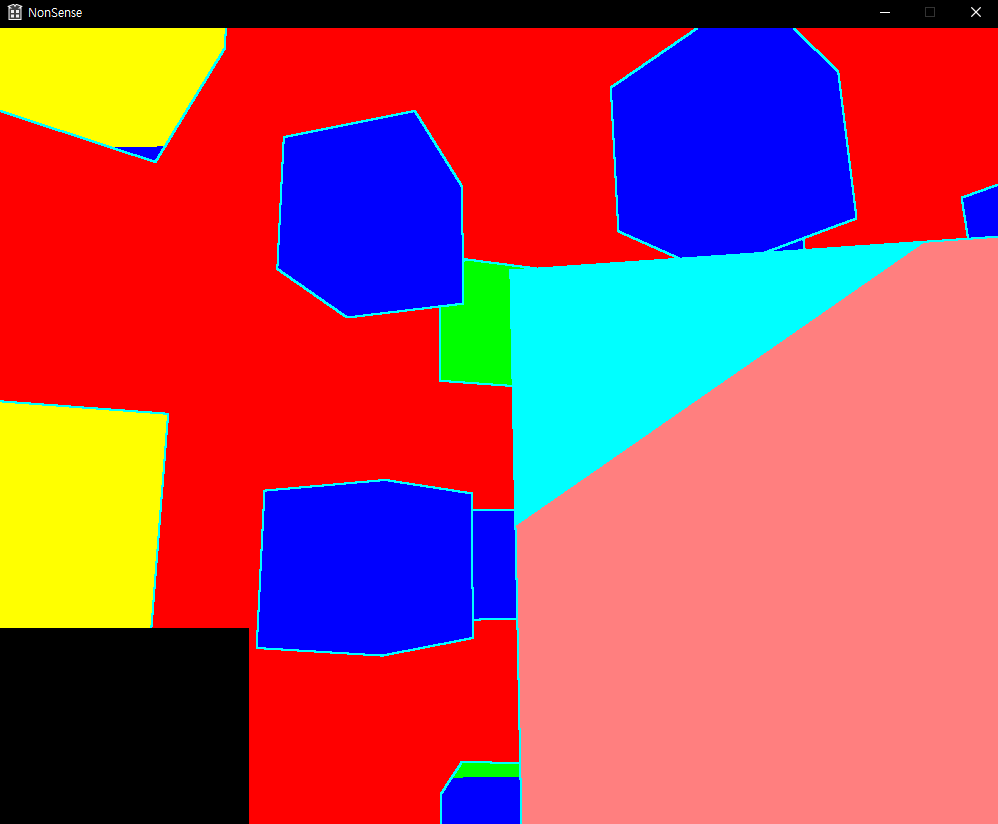
Depth값을 출력하면 확실하게 구분이 안되고 흐리게 단색처럼 나오는 것이 너무 마음에 들지 않는다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그래서 위와 같은 함수를 만들어 주었다.

그냥 흑백으로 Depth를 구분하려 하니까 사람의 눈으로 차이를 구분하기 너무 힘들어서 범위별로 아예 다른 색상을 적용하여 눈에 띄게 하였다.

이것이 그 결과이다.

아무래도 작은 육면체 몇 개만 띄우다 보니까 나타나는 현상 같아서 나중에 모델을 띄우게 되면 다시 흑백으로 뜨게 바꿔 줄 생각이다.