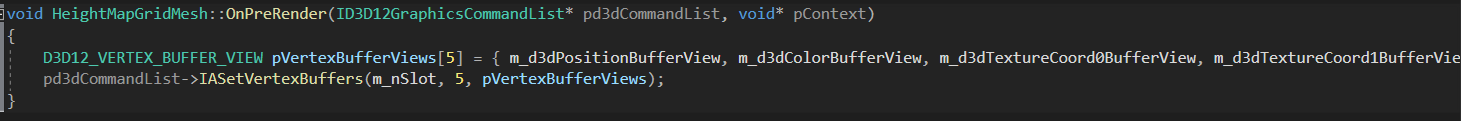
2020182031 이서연 졸업작품 16주차 진행

1. Terrain Map 과 Deferred Rendering과의 병합

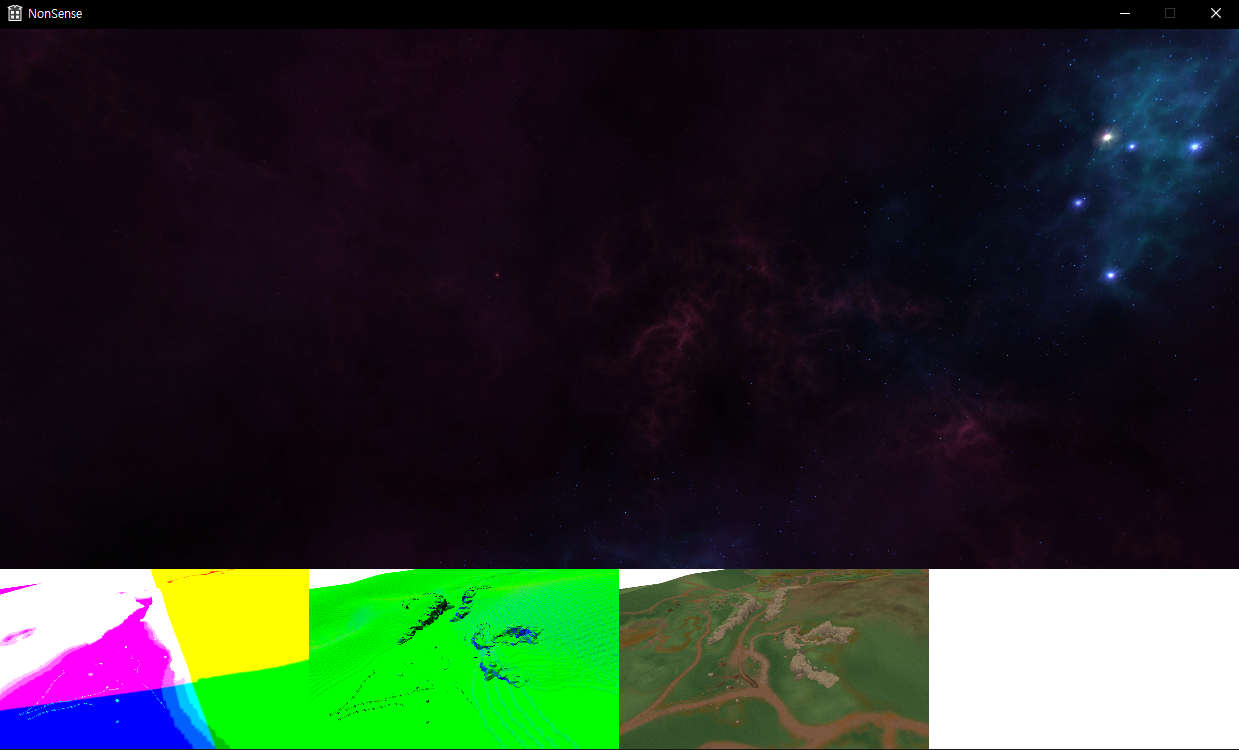
저번 주의 상황에서 Normal값 계산은 올바르게 되고 있다는 것을 디버깅으로 확인하였다.

Normal값이 검정색, 즉 0,0,0으로 계속 들어가는 것은 Input에 전달하면서 생기는 일임을 알아냈고, 이 현상을 고쳐야 Normal이 잘 계산되는 지 확인할 수 있으므로 일단 Normal에 빨간색 (1,0,0)을 넣어두고 검정색이 아니라 빨간색이 잘 출력되는지 확인해보기로 하였다.

결과는 여전히 검정색으로 나오는 것을 확인할 수 있었고, Shader코드로 Input할 때 어떤 부분에서 문제가 생기는 것인지 찾아보았다.



문제는 바로 이곳에 있었는데 내가 Normal Buffer View를 Set해주지 않고 있었기 때문에 추가해 주었다.

결과적으로 이렇게 Terrain의 normal이 나오기 시작하였다.

이제는 Sky Box에 Terrain이 가려지는 문제를 해결해야 하는데 그 전에 나에게 의문점이 하나 생겼다.

그것은 바로 다른 Object (캐릭터, 플레이어 등…)은 Sky Box에 가려지지 않는데 왜 Terrain만 Sky Box에 가려지는 것인지에 대해 의문이 들었는데 이게 왜 의문점인가? 하고 한다면

내가 현재 그리고 있는 화면은 실제 모든 Object들이 바로 그려지는 것이 아니라 MRT를 사용하여 정보를 추출해 낸 뒤 그것들을 조합하여 하나의 Texture로 나타내어 화면에 Texture를 그려주는 것이다.

그렇다면 내가 보고 있는 화면은 개별적인 Object들이 아니라 한 장의 Texture라는 것인데 어째서 한 장의 Texture에 그려져 잇는 Player와 Terrain이 Depth값이 개별적으로 적용되어 어떤 것은 보이고 어떤 것은 안보일 수 있는 것일까?

납작하게 종잇장 같은 Texture라면 전부 보이거나 아예 안보이거나 해야 할 텐데 부분적으로 그려지는 모습이 이상하게 느껴졌다.

그 해답은 바로 Depth Buffer에 있었다.

실제 얇은 Texture를 그리고 있는 것은 맞지만 Depth Buffer에는 모든 Depth값이 저장되어 있고, Depth Test를 할 때 Depth Buffer를 사용하기 때문에 Texture에 볼륨감이 있는 것처럼 부분부분 보이는 것이었다.

그렇다면 내 Sky Box Shader는 Object가 그려지지 않는 빈곳에만 Sky Box를 그리는 작업을 수행하지 못하고 있는 것인가? 아니면 내 Terrain이 이상하게 작동하고 있는 것인가를 알아보기 위하여 Terrain의 Depth를 출력해 확인해보았는데 정상이었다.

그리고 Player나 다른 Object들도 충분히 멀어지면 지워지는 것으로 보아, 여태까지 그려졌던 것이 Sky Box의 범위 내에 있어서 카메라를 따라다니는 육면체인 Sky Box보다 Object의 Depth값이 더 작아서 그냥 그려지고 있었던 것이다.

Sky Box Shader는 Depth Test는 하지만 Depth Write는 하지 않고 있으며, Depth값이 Less or Equal일 때 Depth Test를 수행한다.

하지만 이것은 Sky Box의 Depth가 1일때만 내가 원하는 결과가 나올 수 있다.

왜냐하면 Object들은 아무리 멀리 있어도 Depth가 1에 가까워 질 뿐 1이 되진 않기 때문에 Sky Box의 Depth가 1일 때 어떤 것이 그려져, Depth가 적용되어 있는 공간은 무조건 Sky Box보다 앞에 그려지게 된다. (Sky Box의 Depth = 1, Object의 Depth = 1이하)

하지만 나는 Sky Box의 Depth를 1로 바꾸어 주는 작업을 하지 않고, 실제 Sky Box의 Depth를 사용하고 있었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그래서 Vertex Shader를 다음과 같이 바꾸어 주었는데 position을 보면 모든 계산이 끝난 뒤 XYZW가 아닌 XYWW를 넣어주는 것을 볼 수 있다.

언뜻 보면 W엔 1이 들어가니까, XYWW로 Z를 W로바꾸어, Z에 1이라는 값을 넣는 것으로 보인다.

그럼 왜 Z에 1을 넣지 않고 굳이 XYWW라고 적지? 라는 의문이 들 수 있고 실제로 저렇게 해주지 않고 Z값에 1을 손수 넣어주면 내가 원하는 대로 작동하지 않는 것을 볼 수 있다.

정확한 이유는 모르겠지만 원근 나눗셈의 과정에서 Z와 W가 같아야 나누었을 때 1이되어서 그런 것이 아닐까? 하고 생각하고 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

결과적으로 이렇게 잘 작동하는 Sky Box를 얻어낼 수 있었다!

1. Terrain Player

교수님의 메일을 읽고 생각을 해보았는데 정말 교수님의 말씀대로 중간발표까지 한달 남음 이 시점에서 그림자같이 부가적인 부분보다 이제 지금까지 만든 툴들을 기반으로 한 스테이지의 게임을 중간발표 전까지 빠르게 만들어야 한다는 생각이 들었다.

그러기 위해선 로그인, 로비, 사용자 설정 방 로비 등… 서버가 포함된 공간 제작과 전체적인 Map의 완성, 충돌처리, Monster들의 행동패턴, 공격에 대한 상호작용, 공격 등… 구현해야 하는 부분이 너무 많이 남아있었고, 그림자를 먼저 할 스케줄이 힘들어 보여, 계획을 변경하기로 하였다.

먼저 현재 작동하는 Object들이 따로따로 움직이고 있기 때문에 Player먼저 Terrain Player화해주기로 결정했다.

Terrain Player란 중력을 받으면서 Terrain에 따라 Player가 언덕을 넘듯이 움직이는 것을 말한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

플레이어의 위치에 해당하는 Terrain의 Y값을 받아와서 Player가 더 아래에 있다면 Player의 속도와 위치를 보정해주는 작업을 한다.

Player는 중력을 받아야 바닥에 닿는지 확인이 가능하므로 중력도 넣어주었다.

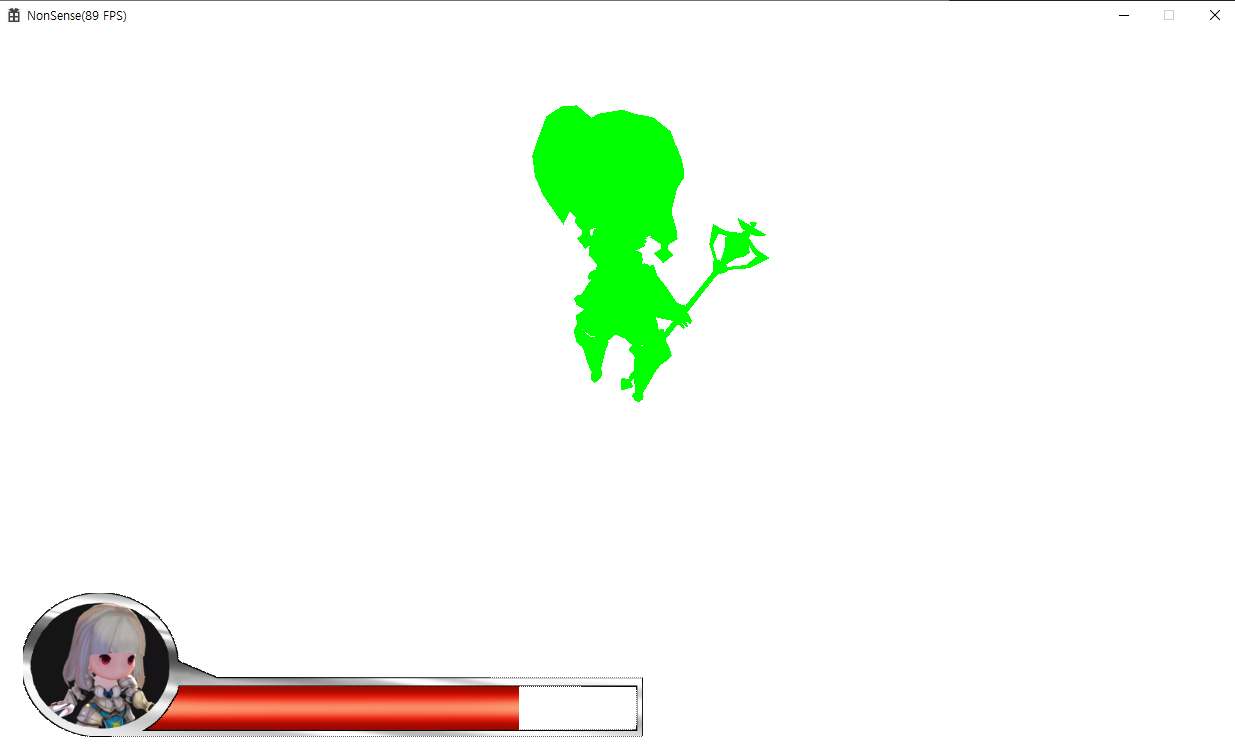
그리고 Mage Player가 생성될 때 Player Updated Context에 Terrain을 넣어주어, 비교가 가능하게 해 주었다.

그리고 Player의 Update에서 위 함수를 넣어주었더니 작동을 하긴 하는데 무언가 그려지는 Terrain과 다르게 작용하는 것을 확인하였다.

1. 노트북을 위한 Branch 생성 및 UI작업

아직 코드 최적화 관련 작업을 하지 않아서 그런지, 노트북에서 Terrain을 포함한 코드를 돌리면 아무것도 출력이 되지 않는다.

그렇기 때문에 원활한 환경인 집에서 Terrain과 공격과 관련된 여러가지 복잡한 그래픽의 작업을 하도록 하고 UI같이 단순한 2D그래픽 작업을 노트북에서 하기 위하여 노트북을 위한 Branch를 하나 생성해주고 화면을 간단하게 바꾸어 주었다.

최소한의 Render만 하는 화면

그리고 UI를 Object List에 넣어줘서 한 번에 관리해주기로 하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명다음과 같이 List 세트를 만들어 주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그러면 이렇게 생성만 해주고 변수에 넣어주지 않아도 저절로 UI List에 UI Object가 담기게 된다.

UI는 가장 마지막에 그려줘야 하므로 Render(불투명 객체), Blend Render(투명 객체)외의 다른 Render함수를 만들어 주기로 하였다. (UI Render)

텍스트이(가) 표시된 사진

자동 생성된 설명

그리고 다음과 같은 순서로 Frame Work에 넣어주었다.



이렇게 출력되는 비상!!!이 걸렸지만… Depth Stencil Buffer가 Sky Box의 것으로 설정되어 있는 것을 확인하고 고쳐주었더니 정상적으로 작동하였다.

1. Release Mode실행 가능하게 바꾸기

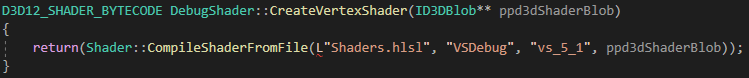
우리의 코드는 처음부터 치명적인 버그가 있었는데 그것은 바로 Release Mode로 실행이 안된다는 것이다.

텍스트, 컴퓨터, 스크린샷이(가) 표시된 사진

자동 생성된 설명

우선 보이는 것처럼 엄청나게 많은 오류가 생기는데 모두 고쳐보려고 한다.

가장 많이 보이는 오류는 WCHAR과 L문자열의 변환 차이인데 Debug모드에선 되는데 Release에선 안 되나 보다. (혹시나 해서 유니코드 집합이 맞는지도 확인해 보았다.)



보이는 것처럼 L에 빨간 줄이 쳐지게 된다.

그래서 L”문자열”의 형태를 모두 (WCHAR\*)”문자열”의 형태로 바꿔주어 형식에 맞춰주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그리고 l-value관련 오류들이 많이 나타났는데 이것도 함수로 불러오는 값을 변수에 지정해 준 후 변수를 넣어주는 방식으로 고쳐주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

또 다른 예시로 함수 내에서 l-value를 사용하기 위하여 변수를 만들어 넣어주었다.



다음과 같이 고쳐주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명