2020182031 이서연 졸업작품 13주차 진행

1. DDS 리소스 오류 수정

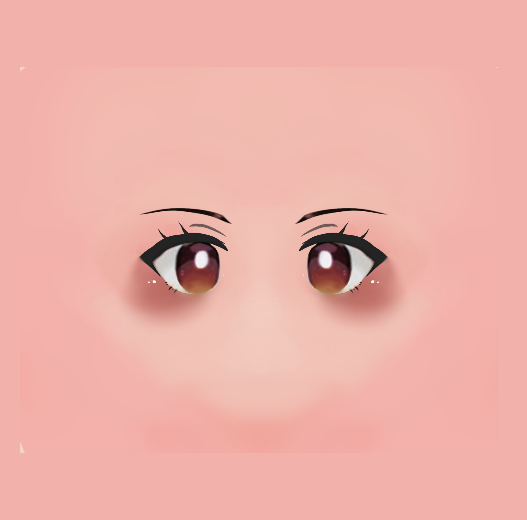
현재 우리 캐릭터의 얼굴 상태

왜 이런 가루상 같은 얼굴을 하고 있냐… 생각해보면 무조건 Alpha 채널의 문제라고 할 수 있다.

그래서 내가 제작한 Alpha값이 있는 PNG를 그대로 DDS로 변환하여 적용해보았다.

 <- PNG  <- 결과

이상하다 분명 Alpha값이 잘 적용되어있는데.. 그래서 급한대로 일단 피부를 칠해주기로 결정

 <- PNG 인형, 장난감이(가) 표시된 사진

자동 생성된 설명 <- 결과

이제야 좀 예뻐졌지만 사실 이 Alpha값 문제는 빌보드 때문에 해결해야 한다.

1. Lighting Pass 생성

Geometry Pass를 만들었으니 이제 Lighting Pass를 만들어야 할 때이다.

원래는 Lighting Volume을 사용하여 조명 + Pixel의 계산을 하려했으나 일단 임시적으로 조명 \* Pixel의 계산을 하는 Phong Lighting을 해보기로 했다.

텍스트이(가) 표시된 사진

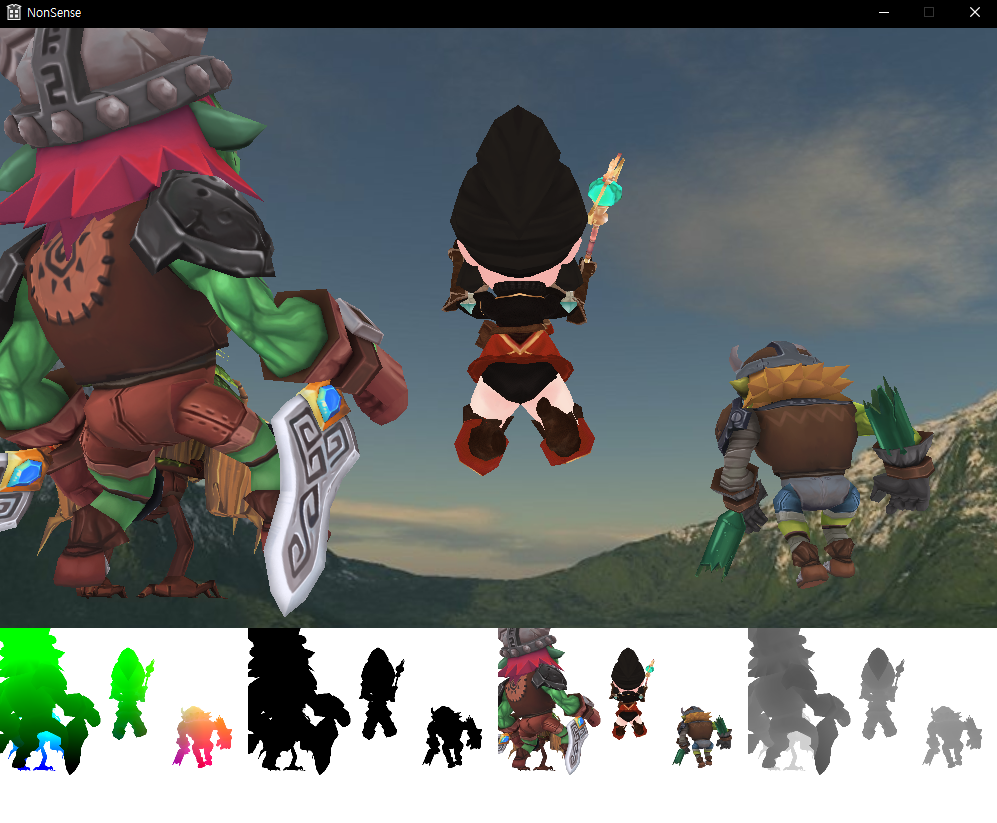
자동 생성된 설명Normal값 대신 Lighting함수 적용

텍스트이(가) 표시된 사진

자동 생성된 설명Lighting함수는 교수님의 코드를 사용했는데 대충 계산 방식이 이러하다.

최대 조명 개수만큼 조명을 검색 한 후 POINT\_LIGHT, DIRECTIONAL\_LIGHT, SPOT\_LIGHT를 가려 낸 후 각각 계산을 한 뒤 Color에 더해주는 방식으로, 조명 수 \* 모든 Object의 Pixel수라는 엄청난 계산양을 가진 함수이다.

그리곤 Global Ambient을 더해주고 이 모든 계산에는 Material의 값이 들어가는데 이 gMaterial은 재우님이 Object들을 그릴 때 넣어 주시는 것 같다.

Lighting을 적용 한 모습

아예 검정색으로 나오는 것을 확인 할 수 있는데 조명이 없나 싶지만 Player가 손전등을 들고 있는 것처럼 Player의 Position과 Look Vector를 가지는 Spot Light가 존재하는데 출력되지 않고 있다.

이상하다 싶어서 Global Ambient를 0.5씩으로 확 올려봤는데 이렇게 하면 저 검정색이 회색으로 변해야 한다고 생각했으나 같은 결과가 나왔다.

그래서 Material들이 잘 들어가지 않고 있는 것 같아서 Global Ambient만 출력되도록 해보았으나 Ambient를 0.9씩으로 해도 검정색으로 출력되고 있었다.

어떤 값이 이상한지 알아보기 위하여 return값 등 가장 먼 변수부터 시작해서 여러 변수들을 바꿔본 결과

텍스트이(가) 표시된 사진

자동 생성된 설명이럴 땐 빨간색으로 나오지만 여기에 gMaterial을 곱하면 검정색이 되는 것을 보니 빛은 잘 들어가지만 (이거도 사실 확실하진 않지만 다른 예제에서 나오는 것을 보면 잘 들어가는 듯 하다) gMaterial이 이상하게 들어가고 있다는 것을 알 수 있다.

1. Sky Box 생성

이제 텅 빈 Map에 Sky Box를 생성해 주려고 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

위와 같이 안쪽을 바라보는 Sky Cube Mesh를 만들어준다. (Texture가 안쪽에 그려져야 하기 때문)

텍스트이(가) 표시된 사진

자동 생성된 설명

스카이 박스의 생성자인데 20사이즈의 육면체 Mesh를 생성해주고 Mesh를 Set해준다.

그리고 Texture를 Set해줄건데 먼저 Sky Box Texture를 Load해준다. RESOURCE\_TEXTURE\_CUBE가 Cube형 Texture 유형이다.



DDS파일을 열어보면 다음과 같이 6개의 Texture가 하나의 Texture로 묶여서 저장되어 있다.

그 다음은 미리 만들어준 Sky Box Shader를 만들어서 연결해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

다음과 같이 Position값만을 가져와서 그려주는데 그 이유는 Sky Box는 실제로 커다랗게 육면체를 그리는 것이 아니라 카메라 주변에 작게 육면체를 그려서 안쪽에 Texture를 입혀주는 것 이므로 카메라의 위치로 Sky Box의 위치를 바꿔줘야 하기 때문이다.

또한 Depth검사를 하지 않아, 가장 가까이 그려도 모든 Object들이 다 그려지게 해준다.

그리고 Sky Box의 Texture가 담긴 Root Parameter 번호는 17번이기에 Shader Resource View를 17번에 Texture가 Set되도록 해준다.

그리고 Material을 만들어서 Texture와 Shader를 Set해주어 하나의 재질로 묶어주면 Sky Box준비 완료!

그려주는 순서를 Sky Box -> Alpha가 없는 Object -> Light -> 앞의 정보들을 합친 Screen -> Alpha가 있는 Object (Forward) -> UI 식으로 그려줘야 할 것 같아서 Scene의 Object Render앞에 넣어주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그랬더니 Sky Box가 출력되지 않았다!

생각 해 보니 Deferred Rendering내에서 그려도 결국 최종적으로 그려지는 Screen에 막혀 그려지지 않고 있던 것이다.

텍스트이(가) 표시된 사진

자동 생성된 설명 Screen Shader를 출력하지 않는 모습

이 상태는 MRT의 Target0인 Scene부분이라고 볼 수 있다.

원래는 float4(0,1,0,1)로 Object가 있는 부분만 초록색으로 나와야 하는데 Sky Box가 이곳에 그려지고 있는 것이 확인되었다.

그래서 Sky Box를 Deferred Rendering에서 빼낸 뒤 Screen Shader가 출력된 이후 그려 주기로 하였다.

예전에 3D 게임 프로그래밍 시간 때 교수님에게 배웠던 Sky Box내용 중 Sky Box를 가장 나중에 Render하면 효율적이라는 말을 들은 적이 있다.

그 이유는 화면에 그릴 때 Sky Box를 그리고 Object를 그리면 Sky Box를 화면 전체에 한번 꼭 그려야 하지만 Object를 먼저 그리고 Sky Box를 그리면 계산이 가벼운 Depth Test에서 Object부분은 거르고 Rasterization단계로 넘어가기 때문에 Object부분은 빼고 그릴 수 있어 더 계산양이 적어지기 때문이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그래서 다음과 같이 코드를 짜주었다.

Screen의 On Prepare Render Target함수 이후가 MRT에 적용하는 내용이고 OM Set Render Targets함수 이후가 Swap Chain Buffer에 그려주는 것이다.

텍스트, 모니터, 실내, 화면이(가) 표시된 사진

자동 생성된 설명Screen을 먼저 출력하고 Sky Box를 그린 결과이다.

Object들이 출력이 안 되고 있는 것을 확인할 수 있는데, 그도 그럴 것이 Depth Test를 끄긴 했지만 가장 마지막에 그려서 화면을 덮기 때문이다.

텍스트, 명판, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그래서 Depth Stencil State를 다음과 같이 바꾸어 주었다. Depth Test를 하는 대신, Depth Function을 D3D12\_COMPARISON\_FUNC\_LESS\_EQUAL로 바꾸어 주는 것이다.

이게 무슨 뜻이냐 면 해당 Shader의 Depth값보다 작거나 같은 값을 Pass시켜 준다는 뜻이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이렇게 Sky Box 구현을 성공했다!