**2018180020 박재우(클라이언트) 2주차 기록**

**프레임 워크 제작중의 오류 해결**

서연이가 프레임 워크 제작을 진행하던 중 Direct X 12 오류 발생



D3D12 ERROR: ID3D12CommandList::SetGraphicsRoot32BitConstants: The currently set root signature declares parameter [1] with type CBV, so it is invalid to set root constants here. [ EXECUTION ERROR #709: SET\_ROOT\_CONSTANT\_INVALID]

이 오류의 내용은 RootParameter 1번 인덱스의 타입이 CBV로 설정 되어있는 반면에 값은 32BitConstants로 넘겨주려고 하기 때문에 나타나는 오류였다. 이 오류를 해결하기 위한 두가지 방법을 생각했는데 첫번째로 RootParameter의 타입을 CBV에서 32BitConstants로 변경하는 방법이 이었다.



D3D12 ERROR: ID3D12Device::CreateRootSignature: Root Signature size exceeds maximum of 64 32-bit units. Costs in units: Descriptor Tables cost 1 each, Root CBVs cost 2 each, Root SRVs cost 2 each, Root UAVs cost 2 each, and Root Constants cost 1 per 32-bit value. [ STATE\_CREATION ERROR #699: CREATE\_ROOT\_SIGNATURE\_NOT\_SUPPORTED\_ON\_DEVICE]

가장 간단한 방법이라 생각 했으나 RootSignature의 최대 크기를 벗어난다는 오류가 발생했고 이로 인해 RootSignature가 제대로 생성되지 않아 그 뒤로 RootSignature가 없다는 오류가 발생 했다.

두번째 방법은 값을 넘겨주는 방식을 32BitConstants에서 CBV로 넘겨주는 방법이다.

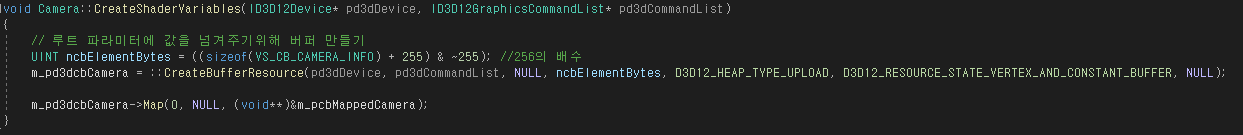


이 방법을 위해 ID3D12Resource 변수와 카메라의 View와 Projection 정보를 가지고 있는 구조체 변수를 선언했다. UpdateShaderVariables 함수에서 View의 값을 직접 넘겨주던 방식에서 memcpy를 이요해 m\_pcbMappedCamera에 해당되는 값을 넣어준다.



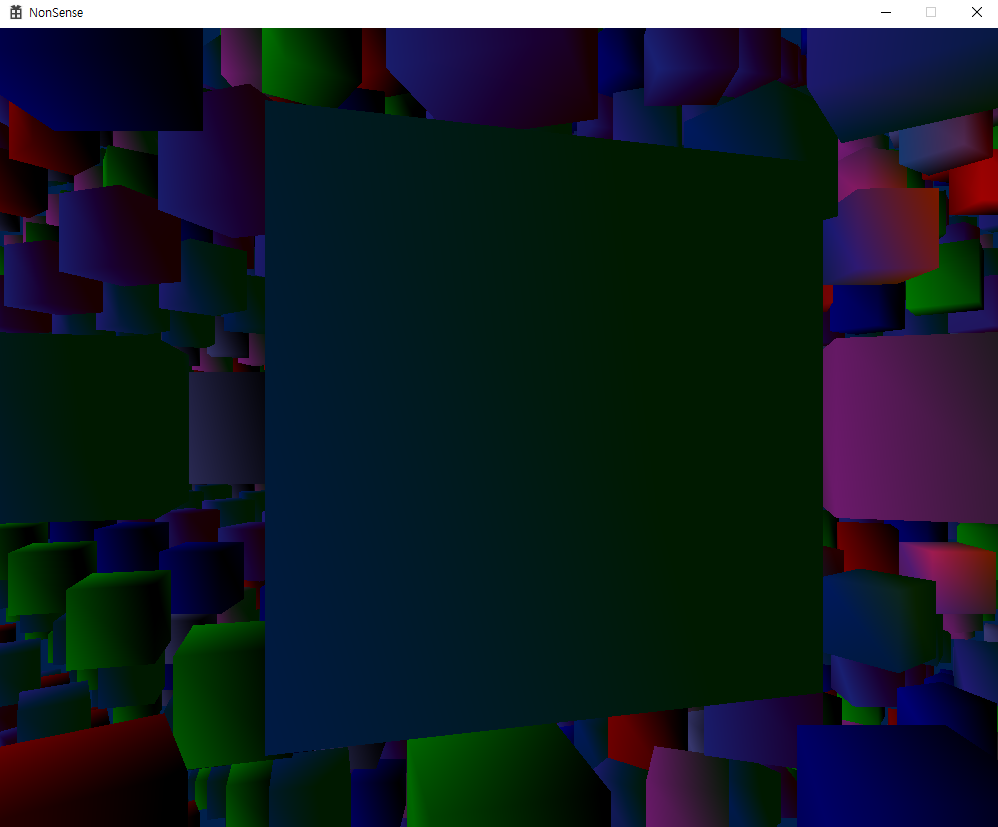
그 후 리소스 변수를 이용해 GPU가상 주소를 얻어 RootParameter에 넘겨준다.

m\_pcbMappedCamera에 저장된 값을 m\_pd3dcbCamera를 이용해 값을 넘겨주기 위해서 m\_pcbMappedCamera의 값을 m\_pd3dcbCamera에 매핑 해줘야 한다.



추가로 타입이 CBV이기 때문에 버퍼를 생성해 줘야한다.

이 과정으로 오류를 해결이 가능했고 이후 실행해 보니 BuildObjects() 함수를 실행할 때 마지막에 CreateShaderVariables() 함수를 부르지 않아 Material과 Light의 값의 버퍼가 생성 되지 않아 실행이 되지 않았기 때문에 BuildObjects()함수 마지막 부분에 CreateShaderVariables()를 호출해 주었다.



정상적으로 실행이 된 모습

**모델을 유니티로 추출할 때 발생하는 문제**

유니티로 모델을 추출할 때 머리카락 부분의 본이 제대로 추출되지 않아 찌그러지는 현상이 있었다. 이용희 교수님께 질문 한 결과 본이 너무 많아 문제가 되는 것으로 추측된다는 의견을 받았고 해결 방법으로는 모델의 FBX자체를 수정해서 머리카락의 본을 제거하는 방법을 제안하셨다.

아직 이 방법으로 수정을 해보지 않아 확실하지 않음.

두번째 방법으로 FBX SDK로 로드하는 방법이다. FBX SDK 예제로 우리의 모델을 띄웠더니 문제없이 그려지는 모습을 확인했다. 새로운 프로젝트를 생성해서 FBX를 추출하는 Exporter를 제작하는 방향으로 해야 할 듯하다. 따라서 계획을 변경해서 다음 주차부터 Exporter 제작을 할 예정.

**VIVOX**

Vivox에 대한 정보를 얻기 위해 유니티의 개발자 포럼에 접속을 해 봤지만 승인이 따로 필요했다.

승인을 요청하고 1월13일에 승인이 되어 검색이 가능하게 되었다. 찾아보니 Vivox측에서 접근할 때 사용할 인증키를 주고 그 인증키를 이용해 Vivox 서버에 접속하는 방식이고 우리가 추가로 서버를 만들 필요는 없었다. 앞으로 Vivox를 어떻게 사용해야하는 지 더 공부 해야한다.