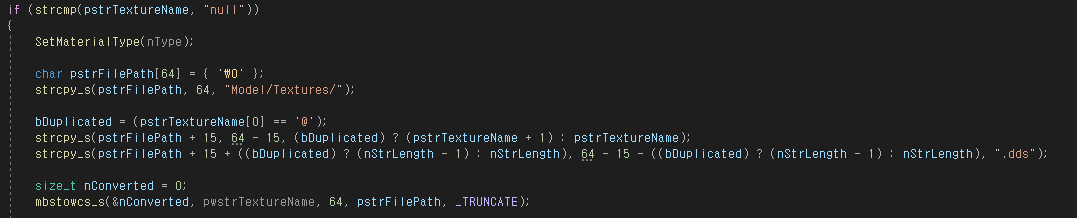
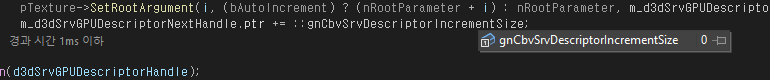
**2018180020 박재우(클라이언트) 12주차 기록**

텍스쳐가 모두 Skin으로 입혀지는 문제를 해결해야 한다.



위 코드는 텍스쳐의 파일을 읽을 때 파일의 경로를 생성하는 코드이다. pwstrTextureName에 경로를 넣으면 그 경로의 텍스쳐를 사용하는 구조이다. 현재 사용하는 모델은 텍스쳐가 총 7번 불리는데 디버그를 해봐도 7번 모두 잘 읽는 것은 확인했다.

다음으로 의심가는 부분은 UpdateShaderVariable에서 값이 Skin만 넘어갈 수도 있다.



CreateShaderVariable할 때 DescriptorHandle의 포인터를 이용해 텍스쳐들을 저장하는데 Size가 0이라서 한 곳에만 텍스쳐가 저장되어 마지막에 불린 Skin텍스쳐만 저장되어 모두 Skin으로 입혀지는 문제인 것 같다.



코드를 보니 이미DescriptorSize를 설정해주고 있었다. 이것을 사용해서 그려본다.

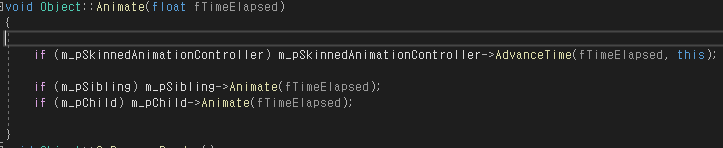
정상적으로 잘 그려졌다!

얼굴은 Alpha값이 문제인 것 같으므로 서연이한테 수정해 달라고 한다.

**애니메이션 재생**

모델을 성공적으로 그렸으니 이제 애니메이션을 재생해 보도록 한다.

Scene의 Animate함수에서 불러온 모델의 Animate를 호출한다.



Animation Controller의 AdvanceTime 함수를 호출한다. AdvanceTime은 경과 시간을 받아 시간에 따른 뼈들의 위치를 Set 해준다.

Animate함수 이후 Render를 해주면



사진들이라 표현하기 어렵지만 애니메이션이 재생되는 모습은 확인할 수 있었다.

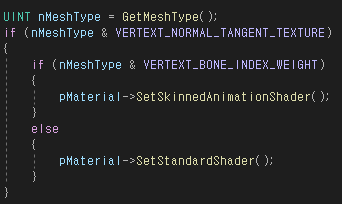
이제 유니티로 여러 모델과 모델의 여러 애니메이션을 추출하고 띄워보도록 한다.

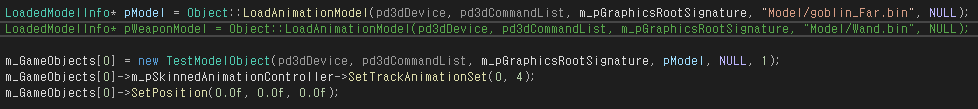


무기를 손에 들고 있는 모델을 띄워보니 무기가 망가져 있다. 추출과정에서 뼈의 영향을 받아 버린 듯하다. 그래서 무기의 모델만 따로 추출하여 캐릭터의 손에 붙이는 방법으로 시도해 보기로 했다.

원인은 무기 자체는 스키닝 모델이 아닌데 스키닝 쉐이더로 랜더링 하려 해서 문제가 된다. 그래서 StandardShader로 로드한 모델을 그린다.

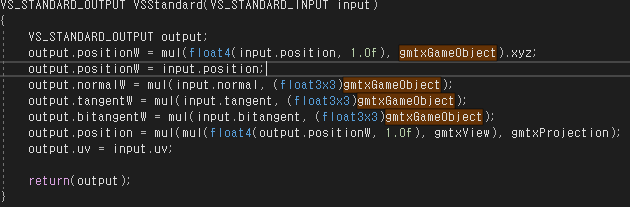
모델을 로드하는 과정에서 Mesh의 타입들을 기록해서 Material을 만들 때 기록된 타입으로 쉐이더를 설정해준다.





이제 모델을 불러 띄워봤지만

무기가 나오지 않는다.

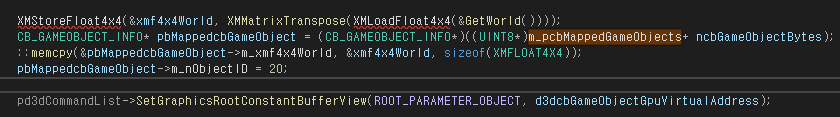


VS의 코드를 살짝 수정해서 테스트를 했다. positionW를 일반 position값으로 바꿨더니



그려지기는 했다.

gmtxGameObject의 값이 잘 안들어왔다고 추측할 수 있다.



UpdateShaderVariable 함수의 일부분이다.

pbMappedcbGameObject가 m\_pcbMappedGameObjects의 포인터를 받아와서 그 포인터를 이용해 값을 변경해주는 구조였는데 CBV가 익숙하지 않아 서연이가 Shader에 만들어 놓은 것을 참고하서 만들었다. 뒤에 ncbGameObjectBytes를 더하는데 서연이가 만든 Shader에서의 ncbGameObjectBytes는 하나의 Shader가 그리는 오브젝트가 많아 그 오브젝트의 크기만큼 포인터를 늘려주는 역할이었던 것이었다. 하지만 나는 오브젝트 하나를 그리기 때문에 저기서 추가로 더할 필요가 없었다.



문제가 되는 부분을 지우고 실행해 보니 잘 되는 것을 확인할 수 있었다.

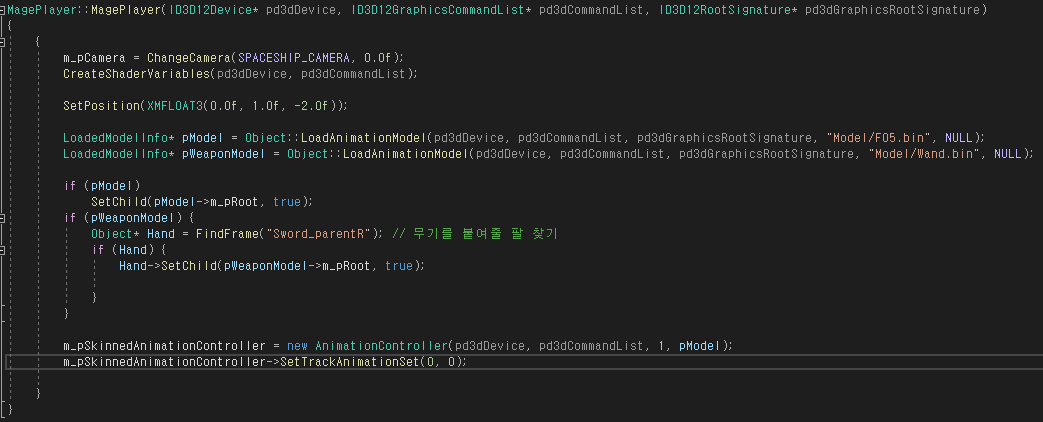
손에 들려있는 모양새가 어색하지만 이 또한 해결해야 할 문제이다.

**키 입력에 따른 애니메이션 재생**

하나의 애니메이션이 무한 반복하고 있으면 의미가 없으므로 키 입력에 따라서 애니메이션을 바꿔줘야 한다.

우선 플레이어 객체에 모델을 추가해 3인칭 화면을 만들어 본다.

MagePlayer 클래스를 추가하고 모델을 추가하는 작업을 했다.



카메라 모드를 바꿔주는 코드도 추가하고 GameFrameWork.cpp의 BuildObject 함수에서 플레이어를 생성할 때 클래스를 MagePlayer로 바꾼다.



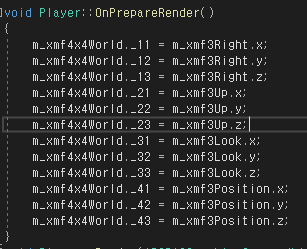
그리고 GameScene에서 Animate가 호출될 때 플레이어의 Animate함수도 호출해 애니메이션이 재생되게 한다.

****

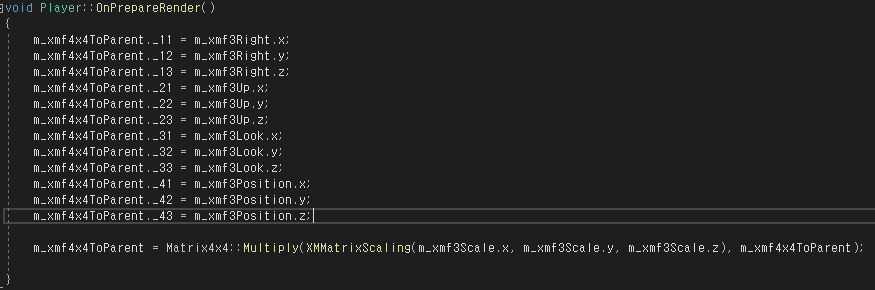
**(이 때부터 디퍼드 렌더링이다.)**

원점 위치에 플레이어가 그려진 모습을 볼 수 있다.

지금은 움직여도 카메라만 움직인다. 플레이어 모델도 같이 움직이게 해야 한다.



OnPrepareRender에서 World행렬을 바꿔주고 있었다. 사용하는 모델은 단일 모델이 아닌 계층 구조의 모델이므로 World가 아닌 ToParent행렬에 값을 넣어준다.



위와 같이 변경해주고 Scale을 할 때를 대비해 Scale값도 반영해준다.

이제 모델이 잘 움직이니 입력에 맞게 애니메이션을 재생하는 작업을 해준다.

먼저 걷는 모션이다. 걷는 다는 것은 플레이어가 이동한다는 뜻이고 이동하면 속도 관련 변수에 변동이 있을 것이다.

일단은 방향과 상관없이 움직임이 있다면 걷는 모션이 나오게 해보겠다.

x방향과 z방향의 속도의 벡터 길이가 0이 아니면 움직이고 있다는 뜻이니까 조건에 맞느다면 AnimateSet을 변경해준다.



움직이면 애니메이션이 걷는 모션으로 바뀐다.