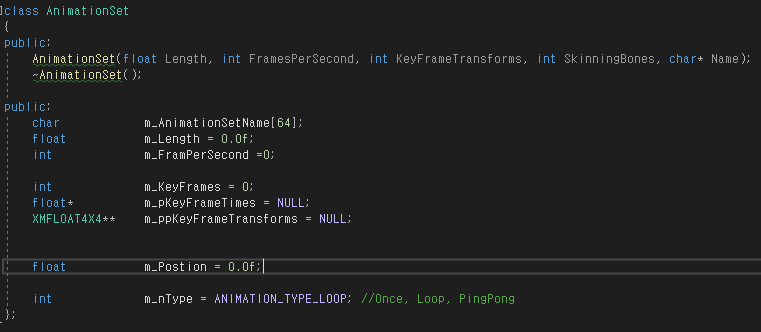
**2018180020 박재우(클라이언트) 8주차 기록**

1. 모델이 포함한 정보를 저장할 객체를 만들어야 한다. (텍스쳐, 머터리얼, 애니메이션 등등..)
2. 모델이 포함한 정보를 읽을 함수를 만들어야 한다.
3. 모델을 렌더링할 때 사용할 쉐이더를 만들어야 한다.
4. 모델이 가지고 있는 리소스를 쉐이더로 넘겨줘야 한다.

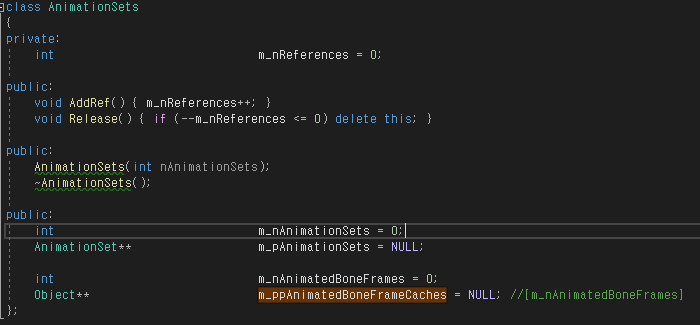
**모델 띄우기 2번**

파일에서 읽어야 할 남은 데이터는 애니메이션 데이터이다. 애니메이션은 <AnimationSets>에 <AnimationSet>이 있고 그 밑에 <Transforms>가 존재한다. 각 프레임마다의 트랜스폼의 데이터가 입력된다. 우선 AnimationSet의 데이터를 담을 클래스를 생성해준다.



AnimationSet의 데이터는 애니메이션의 길이, 이름, 1초당 프레임 수, 키 프레임들의 트랜스폼 데이터 등등 있다. 타입은 Once, Loop, PingPong이 있는데 각각 한번 재생, 무한 재생, 정상 재생했다 역 재생이다.

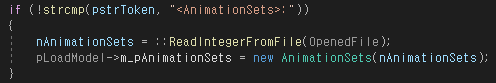
AnimationSet들을 모아놓은 클래스도 필요하다. 이름은 AnimationSets로 한다.



AnimationSets는 위와 같다.

이제 애니메이션을 로드하는 함수를 만든다. 애니메이션을 담을 공간이 필요하므로 LoadedModelInfo 클래스에 AnimationSets를 추가하고 이 AnimationSets에 데이터들을 넣는다.

<AnimationSets>: 후에 정수가 나오는데 이 정수는 AnimationSet의 개수이며 그 개수만큼 <AnimationSet>이 나오게 된다. 따라서 정수 값 하나를 읽어 AnimationSet이 몇 개인지 저장을 해둔다.



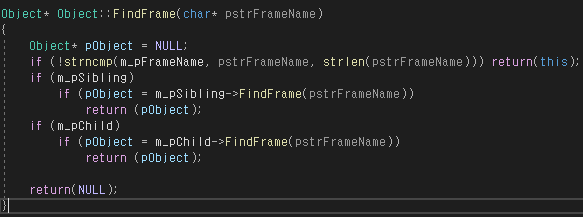
<AnimationSet>: 이 나오기 전에 <FrameNames>: 가 나온다. 이 애니메이션에서 움직이는 뼈들의 이름이 나열 되고 이 뼈들의 이름은 저번에 만든 Mesh를 불러오는 함수에서 해당 데이터들도 이미 읽어온 상태이다.

<FrameNames>:의 구조는 정수 값 과 정수 값과 맞는 수의 프레임 이름이다. 먼저 정수를 읽어 총 몇 개를 읽어야 하는지 저장한 후 각 프레임에 대한 객체들을 생성해준다.



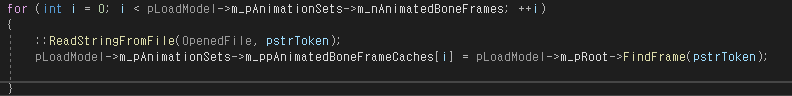
AnimatedBoneFrameCaches는 각 프레임들의 주소를 저장하고 이 값으로 애니메이션이 재생된다.

m\_nAnimatedBoneFrames에 몇 개의 프레임이 있는지 저장했으므로 해당 값만큼 루프를 돌아 m\_ppAnimatedBoneFrameCaches에 저장을 해준다. 그 전에 먼저 읽은 문자열과 일치하는 이름의 프레임을 찾아줄 필요가 있다.

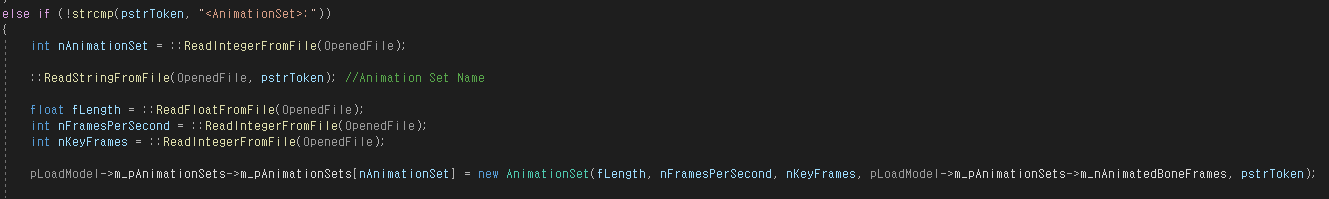


프레임을 찾는 함수는 위와 같다. 자기 자신의 이름을 인자로 받아온 이름과 비교하여 같으면 자기 자신을 반환하고 아니라면 자식 또는 형제노드의 FindFrame 실행한다. 그러면 재귀적으로 함수가 호출되고 해당되는 프레임을 찾으면 최종적으로 프레임의 포인터가 반환될 것이다.

이제 FindFrame을 이용해 읽은 문자열과 같은 이름의 프레임을 찾아 저장한다.



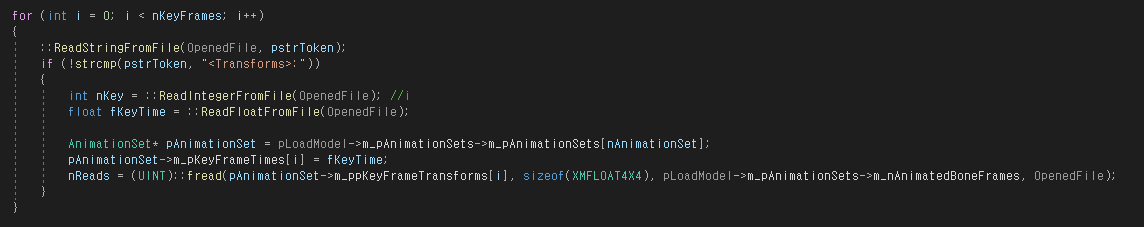
다음은 <AnimationSet>: 이다. <AnimationSet>: 이후 정수, 문자열, 실수, 정수, 정수 순으로 나온다. 첫 번째 정수는 해당 AnimationSet의 번호이다. AnimationSet이 여러 개인 경우 AnimationSet이 배열의 형태로 생성되고 첫 번째 정수가 인덱스 번호가 된다. 문자열은 Animation의 이름이다. 다음으로 나오는 실수는 이 애니메이션이 걸리는 총 시간, 애니메이션의 길이가 된다. 4번째 인자인 정수는 1초당 프레임 수(FPS)이고 마지막 인자는 애니메이션의 재생을 위한 키 프레임의 총 개수이다. 이 개수만큼 뒤의 <Transforms>: 가 나오게 된다.



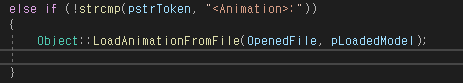
먼저 AnimationSet의 5개의 인자를 모두 읽어와 저장한다. 그리고 읽은 인덱스 번호에 맞는 AnimationSets에 AnimationSet을 생성해준다.

<Transforms>: 는 각 키 프레임들의 행렬 정보이다. 위에서 읽은 키 프레임의 총 개수만큼 나오므로 for문으로 루프를 돌아 개수에 맞게 읽는다.

구조는 정수와 실수가 나오고 그 이후로는 전부 행렬 데이터이다. 앞에 나오는 정수는 각각 몇 번째의 키 프레임인지 이 키 프레임의 시간대이다. 이 시간대가 0.153 이면 0.153초에 이 키 프레임의 차례라는 뜻이다. 이후의 행렬들의 개수는 위에서 읽은 본 프레임의 개수와 같다.

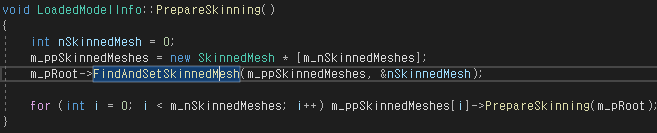


이제 파일을 읽는 작업은 전부 끝났다.

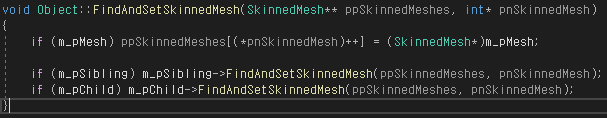


마지막으로 Animation을 읽도록 만든 함수를 호출한다.

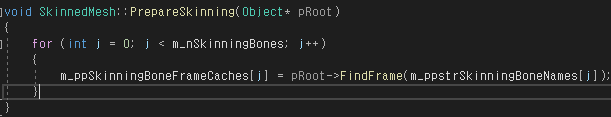
지금까지는 오브젝트를 읽어 LoadedModelInfo의 RootObject에 저장을 하고 Animation을 읽었기 때문에 실제로는 SkinnedMesh를 읽었어도 LoadedModelInfo의 SkinnedMesh에는 데이터가 들어있지 않다. 따라서 읽은 데이터를 기준으로 LoadedModelInfo의 데이터도 바꿔 줘야한다.



LoadedModelInfo에 PrepareSkinning이라는 함수를 만들었다. 비어있던 m\_ppSkinnedMeshes에 새롭게 SkinnedMesh를 생성하고 Root 오브젝트의 FindAndSetSkinnedMesh를 부른다. 이 함수는 오브젝트의 Mesh를 첫 번째 인자인 m\_ppSkinnedMeshes의 Mesh에 저장한다.



ppSkinnedMeshes는 배열의 형태이기 때문에 인덱스 번호가 필요하다. nSkinnedMesh가 0인채로 2번재 인자에 들어가는 것이 이유이다. 해당 인덱스의 ppSkinnedMeshes에 m\_pMesh를 대입하고 다음으로 형제와 자식도 FindAndSetSkinnedMesh를 호출해 읽어온 Mesh를 받는다.



마지막으로 SkinnedMesh의 PrepareSkinning 함수를 부르는데 이 함수는 Mesh의 SkinningBoneFrameCaches도 찾아서 넣어주는 역할이다.

이제 애니메이션을 읽은 후에 PrePareSkinning을 호출하면 파일을 읽어 모델을 로드하는 부분이 끝이 난다.

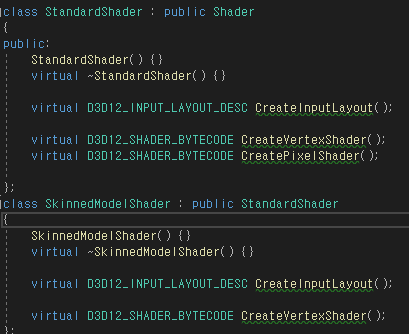


이렇게 모델을 띄우기 위한 2번 과정이 끝이 났다.

**모델 띄우기 3번**

모델을 띄우기 위한 쉐이더는 2개가 필요하다. 하나는 일반 모델을 띄우기 위한 쉐이더와 애니메이션 스키닝 모델을 띄우기 위한 쉐이더이다. 모든 모델은 하나의 함수로 불러오기 때문에 로드하는 함수에서 쉐이더를 만들어줄 필요가 있다. 이용희 교수님의 예제에서는 Mesh타입을 계산해서 이 모델이 스키닝 모델인지를 판별하기 때문에 함수를 부를 때 쉐이더 인자 부분에 NULL을 넣었지만 우리는 어떤 모델이 스키닝 모델인지 잘 알고 있기 때문에 모델을 생성할 때 쉐이더를 잘 구분하여 넣도록 한다.

일단 기본적으로 픽셀 쉐이더는 두 쉐이더 모두 같은 쉐이더를 사용한다. 다른 점은 버텍스 쉐이더가 다르고 버텍스 쉐이더에서 요구하는 값이 다르기 때문에 InpuLayout도 다르다. 기본 쉐이더 클래스를 상속받는 StandardShader 클래스를 만들고 이 Standard Shader는 일반 모델을 그리기 위한 쉐이더이다. 이 StandardShader를 상속받는 SkinnedModelShader를 만들었고 이는 스키닝 모델을 그리기 위한쉐이더이다.



StandardShader에서 InputLayout, VertexShader, PixelShader를 변경하므로 가상함수로 3개의 함수를 만들고, SkinnedModelShader는 부모인 StandardShader와 PixelShader만 동일한 것을 사용하므로 InputLayou과 VertexShader에 관한 가상함수를 만들어준다.

쉐이더도 컴포넌트로 뺄 수 있을 것 같은데 이 부분은 일단 모델 띄우는 걸 완료하고 본격적으로 컴포넌트화 시킬 때 고려해 볼 것

우리 팀의 쉐이더 담당은 서연이고 현재 쉐이더 제작을 하고 있으므로 지금 모델을 띄우기 위한 쉐이더는 교수님 예제의 쉐이더를 사용한다. 나중에 서연이가 쉐이더를 완성하면 그 때 교체작업을 하도록 한다.