|  |  |  |  |
| --- | --- | --- | --- |
| MORPHOSIS 개발일지 13차 | | | |
| 기간 | 2019-05-30 ~ 2019-06-02 | 작성자 | 신재욱 |
| 작업 내용 | | | |
| 찾은 문제   * 메쉬의 BoneIdx가 다름. * 보간의 idx를 구하는 식에서 문제가 있음.   해결방안   * 메쉬도 같이 뽑아주기 * 식 수정하기   메쉬에서 내용을 뽑는 것은 이전에도 몇 번 시도했었었지만 그 때도 노멀 값 같은 것들은 못 얻어오고 했었다. 먼저 쓸 예정인 함수들과 클래스 먼저 정리해보자.  <http://docs.autodesk.com/FBX/2014/ENU/FBX-SDK-Documentation/index.html?url=cpp_ref/class_fbx_skeleton.html,topicNumber=cpp_ref_class_fbx_skeleton_html54afb35e-3109-452b-9316-4aa41b5ba8ea>    FbxMesh가 CtrlPoint랑 PolygonVertex 정보를 갖고 있을 예정.  GetAttributeType()으로 eMesh인지 확인한다.      CtrlPoints를 따로 배열로 저장해둔다. 얘랑 PolygonVertexIndex가 필요함.        보니까 FbxVector4를 쓰면 될 것 같다.  오늘의 목표는 무엇으로?    왜 Binormal이랑 Tangent가 안 들어갈까 했는데 아예 exporter에서 설정을 안 해주던 거였음.      잘 들어감~    굿  Mesh는 따로 빼주는게 좋을 것 같다.    그래서 따로 빼냄.    잘 들어간다.  텍스처 정보도 넣고 싶은데 그건 나중에 하기.          ???????    ?????????????? 왜 다 40만 들어가지?????? 디버깅이 안 되니까 너무 힘들다.    갑자기 됨. 야호.  파일 이상하게 만들어지던 것은 해결했으나 결과물은 의도와 달랐다. 정말 본을 띄우는 걸 먼저 해야 할 듯. 큐브 메쉬를 만들고, 이동시키고, 회전 시키고, x축이 진행방향이니까 그 쪽으로 버텍스를 옮겨주고, 렌더링하면 될 듯.  본을 그리기 위해선 무엇이 필요한가? 그릴 메쉬가 필요함.    위가 X축이다.  근데 토폴로지가 삼각형 리스트인데 저게 가능한가? 되겠지.  뒷면도 그려져야 하니까 PSO 만들 때 컬링은 None으로 해야 함.  TexCode로 색을 정해줄까 했는데 굳이 필요 없을거 같음. 하얀 색으로 하자.  Mesh는 만들어줬음. 그럼 이제 Object를 만들어 줘야 함.  Object는 Bone의 개수만큼 만들고, GlobalTransform 변환을 적용함. 그럼 그 위치로 갈 것.  길이는? 어떻게 구하지? 글로벌 트랜스폼의 위치만 구해서 그걸로 길이를 만들기?  생각해보니까 지금 그런게 중요한게 아님. ToRoot의 역행렬을 구하면 초기 상태에서 Global과 같을 것.  ToRoot 위치에 메쉬를 띄우는 것만 하자.  오브젝트를 추가하고 해당 위치에 띄우면 될 것 같다. 오브젝트 관리하는 부분을 보자. 추가해야 할게 많다.지금은 CAnimationPlayerObject \*\*ppPlayers에 할당해서 쓰고 있다. CObject 포인터로 바꿔주고 dynamic\_cast를 사용하도록 하자.    아니??????? 본은 잘 들어가는데??????? ToRoot 정상 맞는데????????? 의도한 대로 잘 움직이는데??????????????????????????????????????????????????????????    정말 무서운 상상이지만 차라리 이게 맞았으면 싶고  정말 아쉽게도 아니었음. 하지만 본 애니메이션은 잘 들어가는 것은 확인.  그럼 뭐가 문제가 될 수 있지????  본 테스트는 ToRoot를 역행렬로 만들어서 그 위치로 매번 옮겨주는 것.  맞는거 아냐? 회전이 이상한가? 위치는 맞나? 위치만 빼서 넣어볼까?  근데 애초에 본 테스트에서 위치 맞았는데 회전 문제일까?  아니면 본은 똑바로 서있는데 메쉬는 똑바로 안 서있어서 그럴 수도 있음.  메쉬 테스트가 더 쉬우니까 메쉬를 먼저 해보자.   * 메쉬는 잘 서있음. 위에서 볼 때 얼굴은 6시를 향하고 오브젝트는 12시가 정면이 맞음.   본 테스트의 메쉬는?      OFFSET, OFFSETINV, TOROOT, TOROOTINV 의 모습.  아래 두 개는 움직이고 있다. 원점의 점들을 OFFSET을 곱하거나, TOROOTINV를 곱하면 정상적인 위치로 이동한다. 근데 원래 모델은 점이 원점에 있지 않다. 그 점을 원점으로 옮긴 뒤에 다시 보내줘야 한다. 그래서 OFFSETINV를 곱해서 원점으로 보내고 TOROOTINV를 곱해 보내줬다.  결과는 이상함. 정확히는 회전의 원점이 이상해짐.        왜 회전을 저길 기준으로 할까? 다른 본들도 다 그럼. OFFSETINV의 위치만 가지고 이동을 시켜보는 것은? 위치만 곱해서 회전 없이 원점으로 보낸 다음에 TOROOTINV를 곱하면?   * 안 됨~   이동 후 회전과 회전 후 이동의 순서 문제가 아닐까?  아니 임의의 이동값을 넣어봤는데도 전혀 변화가 없었음 아니 대체????? 야!!  진짜네 이동 값은 안 들어가고 회전값만 들어가네? hlsl에서 이동값 잘 되는데?    아ㅏ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  이거 안 해줬다~~~~~~~~~~~~~~~~~~~~~    ㅋ  ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ  ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ  ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ                왜 회전방향이 반대일까? 근데 그 와중에 L Upperarm은 정상적으로 회전함.  YZ축 방향 안 바꿔줘서 그런가? 근데 그런 문제는 없었는데?  본 테스트에서도 정상적으로 나왔었잖아. 아냐 본 테스트할 때 다른 각도에서 못 봤어. 카메라 이동을 해야 제대로 볼 수 있지 않을까?  ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ  본이 메쉬랑 앞뒤가 반대였음. 그럼 어떻게 해야 앞뒤만 바꿔줄 수 있을까? 본을 Y축 기준으로 180도 회전? 애초에 메쉬를 정반대로 바꾸면? 메쉬는 바꾸기 쉬운데. (맨 처음 임포트할 때 z값만 바꿔주면 되니까) 함 해보자.    어ㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏ | | | |