|  |  |  |  |
| --- | --- | --- | --- |
| MORPHOSIS 개발일지 10차 | | | |
| 기간 | 2019-04-03 ~ 2019-05-08 | 작성자 | 신재욱 |
| 작업 내용 | | | |
| 상세내용   대체 왜 3ds Max 개발자들은 Clavicle의 부모를 Neck으로 해놨으면서 Rotation Hierarchy는 Spine2를 받아와서 내가 Neck을 꺾으면 어깨도 돌아가게 만들었을까. 정말 이해할 수 없다.  3ds Max에서 직접 본의 하이라키를 수정하여 해결하였다.    Thigh의 회전값이 이상하다.  z축이 저 방향이면 -30도 회전할 때, 몸의 뒤로 가야 하나 내가 만든건 앞으로 가는 것.  왜 앞으로 가는데 저 z축 방향을 가지고 -30도가 나오는걸까. 정말 이해할 수 없다.  아니!! 아ㅏ!!! Max에서 Local 좌표계와 Parent 좌표계로 비교를 해보면서 보니까 Parent 좌표계일 때 나오는 저 0.0 -180.0 4.146 은 현재 bone의 좌표계를 얼마나 회전시켜야 Parent 좌표계의 각도와 같게 할 수 있는지 그 값이었다.  근데 현재 Local 행렬을 구하는 공식에서는 그 값만큼 회전을 시키고 있었으므로 다를 수 밖에.    ?????????????????????  M\_rotation 왜 180 0 180이야 너 0 180 0이어야지(결과는 같은데 나는 결과를 쓰는게 아니니까…)  0번째 키와 20번째 키의 40번째 본에 대한 m\_rotation 값은 각각 (0, 180, 0), (0, 180, -80)이다.(부모 좌표계로 돌아가기 위해 필요한 회전값, 정확히는 0, -180, 0 0, -180, 80 회전한게 맞다)  지금 방식은 키의 값에서 원래 본의 값을 빼고 있었기 때문에 (-180, 180, -180), (-180, 180, -260)이 된다.  애초에 좌표계가 부모와 같아진다. x축이 아래로 가야 하는데?  그럼 저 값을 +로 해서 하면? 어짜피 180은 부호 상관없이 한 바퀴 도는거니까 (180, 180, 180), (180, 180, 260)이 된다. 안 돼. 이미 좌표계가 아래가 아님. 아니지 이건 로컬좌표계에서 얼마나 회전할지를 구하는거니까 굳이 좌표계 안 같아도 됨. 회전할 값만 나오면 그만. 회전할 값은 둘이 0.5에서 보간했다 쳤을 때 (180, 180, 220)가 된다. 180으로 클램프같은걸 해볼까??? (0, 0, 40)이 나오면 되는데.  근데 문제는 179처럼 나오는 것. 아니 걍 뒤에서 앞을 빼면 (0, 0, 40)이 나오잖아.  다음 키 값에서 앞의 키 값을 빼게 하자. 완벽해짐. 해보고 와야지.  정리하자면 지금하는 거에서 뒤에서 앞을 빼고 부호를 반전.  빼는 값을 보간해서 정해야 함. Vector3::Lerp()가 있었다. 다행.  이렇게 했더니 애가 덜덜 떨기만 하고 의도한 결과는 안 나온다.  생각해보니까 앞 키와 뒷 키의 차이만큼만 회전하게 하면 계속 몇 도씩만 나올텐데 회전이 안 되는게 맞나?  앞 키의 값을 더해주면 되지 않을까?  안 됨.    맨 처음 bone의 정보를 담은 m\_vecBone의 내용과 key의 내용을 담은 m\_vecKey의 첫 번째 키의 m\_rotation 부분.  결과는 같게 나오는데 중간값이 달라서 계산할 때 값이 이상하게 나오게 된다.  Local 값 구하는 방식을 다르게 하는게 맞는 듯.    힝힝힝힝  저 값들은 모두 현재 자기 뼈 공간에서 부모 공간으로 가는 LclTranslation과 LclRotation 값.  저걸로 구한 행렬의 역행렬이 의미가 있을 것.  중간의 값은 서로 다르더라도 그걸로 만든 행렬은 같아질 것.  키의 값 (0, 180, 0)과 (0, 180, -30)은 각각 (0, -180, 0), (0, -180, 30)으로 바뀌고 그 중에서 30이 내게 필요한 값. 저것만 남길 방법은?  본의 기본값은 (180, 0, 180)  본의 기본값으로 행렬을 만들고 키의 값을 행렬로 만들고 역행렬로 변환.  둘을 곱하면 필요한 값만 남지 않을까???  그럼 테스트를 해보자.    요호~  허벅지는 정상적으로 앞으로 회전함.  근데 종아리는 여전히 허벅지와 같은 방향으로 회전함.    아…………..    대체 왜 부호가 같을까???      분명 로컬 기준은 둘의 부호가 다른데 그럼 FBX 파일에 값도 각각 30도와 -30도가 되어야 하는 것이 아닌가???      to부모 좌표계 기준  ??????????????????????????  위의 로컬 좌표계 기준에서는 허벅지가 30도, 종아리가 -30도 회전하면 의도한 결과가 나온다.  하지만 부모 좌표계 기준에서는 허벅지가 -30도, 종아리도 -30도 회전한다고 표시됨.  허벅지는 30도 회전하는게 맞다. 그리고 현재 로컬 변환 행렬을 구하는 방식도 허벅지를 30도 회전했다고 구해준다. ~~그런데 종아리는? 종아리가 뒤로 가려면 -30도 회전이 맞다. 근데 문제는 키 값은 부모 좌표계 기준으로 저장되고 따라서 허벅지와 종아리 둘 다 -30으로 저장된다는 점.~~  왜 허벅지는 위로 올렸는데 -부호고 종아리는 아래로 내렸는데 -야???  부모 좌표계와의 z축의 차이?  LclRotation은 부모 좌표계로 돌아가는 각도가 아니라 부모 좌표계 기준 돌아가는 각도였나???  LclRotation은 부모 좌표계 기준 회전.  그 회전을 내 좌표계 기준 회전으로 바꿔줘야 함.  부모 좌표계에서 (0, 180, 80) 회전했으면 내 좌표계 기준으론 (0, 180, -80) 회전해야 하는 것.  이건 정말 모르겠다……  x축이 180도 회전하면 y축과 z축의 부호가 바뀌는건 어떤 계산을 통해 나온 결과일까?    헐    ㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏ  로컬 변환 값을 본의 회전값으로 XMVector3Rotate를 해줬다.  그랬더니~~ 내 좌표계 기준 회전값이 나옴~~~~~~~  그럼 (0, 180, 80) 과 (0, 0, -100)이 로컬 변환으로 얻어진다는 것.(각각 허벅지와 종아리 회전각)  (0, 180, 0) -> (0, 180, 80)  (0, 0, 0) -> (0, 0, -100)  아………… 드디어 잘 나와……………………………    드디어……. 성불할 수 있어…………………………      이제 본 코드에 적용할 시간.  키의 회전각과 본의 toParent 각이 있을 때,  본의 toParent 각으로 쿼터니언을 만들고  키의 회전각을 그 쿼터니언으로 회전시킴  그러면 일단 로컬 변환 각이 나옴. (0, 180, 80)  근데 여기서 구해야 할 것은 (0, 0, 80)이므로 본 toParent의 역행렬을 구해서  XMMatrixMultiply(본 toParent의 역행렬, 로컬 변환 행렬)을 하면  실제로 로컬에서 회전하는 각이 나옴.  이걸 실제로 사용하려면 이전 키와 다음 키의 회전각, 이동값을 먼저 보간해서 하나로 만들고 하는게 좋을 듯.    회전 각은 이제 제대로 나온다.    ?????????????????????????????????  나는 다른 발은 건드린 적 없는데 왜 처음 포즈랑 그 다음 포즈가 다르지??  ~~심지어 왼쪽 사진은 맥스에서의 자세와 다르다. 맥스에서는 오른쪽 사진의 사진 기준 오른쪽 다리와 같은 각도로 지면에 발을 대고 있는데 애니메이션이 적용되면 갑자기 저렇게 다리를 벌리게 된다.~~  ~~오른쪽 사진의 사진 기준 왼쪽 다리를 제외한 부분들은 LclMtx를 XMMatrixIdentity()를 반환하게 해두었다.~~  ~~즉, 단위 행렬이면 정상적인 각도로 펼치고 있다는 뜻. 애니메이션이 들어가면 뭔가 약간씩 각도가 틀어진다.~~  아 이거 오른쪽 사진이 애니메이션 적용된 거였음.  즉, 드레스포즈부터 이상하게 되어있다가 애니메이션이 적용되서 단위행렬이 들어가면 잘 나온다는 뜻인데 대체…?  뭐냐 왜 이래 또..    아 제발.  지금은 translation 정보를 주지 않아서일까? 근데 로컬 변환에서 뼈의 위치가 변하는 경우는 없었다.  전부 회전만 해주었기 때문. 따라서 LclTranslation 정보는 LclMtx를 구할 때 쓸 일이 없다.  FBX File에서는 Bip001가 (0.0, 0.0, 20.9413624) 이동했다는데 Max에서는 (0.948, 0.086, 20.375) 이동했다는데????? 왜 값이 미묘하게 달라????? 아 이건 테스트용으로 따로 만든 것이니까 다를 수 있다. 휴.  왼팔의 키는 10(12)이고 대상이 되는 키는 50번과 70번 키이다.  BoneDefault  translation(2.13474846, -1.13686838e-13, 0.000000000)  rotation(6.08194685, -32.1648941, -1.76954758)  50번째 키  Translation(2.13474846, -1.13686838e-13, 0.000000000)  Rotation(6.08194685, -32.1648941, -1.76954758)  70번째 키  Translation(2.13474846, -1.13686838e-13, 0.000000000)  Rotation(7.19384909, 44.2570496, 6.51094913)  왼팔의 Clavicle을 봤는데 그대로 좌표계 회전하니까 왼팔이 아니라 오른팔 쪽 가르키는데???    Max에서 보니 일단 parent 기준 회전각이 (92.105, -90.0, 90.345)로 FBX File이랑 차이가 있음.(물론 같은 파일이 아니니까~)  또한 translation 값이 parent 좌표계 기준으로 먼저 이동하고 회전한 것으로 보임. 회전 후 이동이 아냐??  지금보니까 R Thigh도 이동 후 회전. 아 당했다!  L UpperArm도 parent 좌표계 기준 회전값은 (45.143, 81.904, 44.857)    내가 쓰는 바이패드를 본으로 바꿔서 보면 편하겠지~  일단 지금 가정은 뭘 할 때는 이동회전 순서로 하고 다른 걸 할 때는 회전이동 순서로 해야한다던지 하여간 그런 순서에 문제가 있다고 정해두고 찾아보기.  L Clavicle의 회전각이랑 이동값이 모두 같다. 맞네. 이동 먼저 하고 회전임. 그럼 돌아올 때는 회전하고 이동이겠네 맞네. 역행렬 해주면 맞네.  2.021, -0.0, 0.728  0.0, -63.241, 179.955  Spine2에서 L Clavicle로 갈 때, 이동을 하고 회전을 함.  그럼  아니 사실 이동을 먼저 하고 있는건지도 잘 모르겠다.  어떻게 저렇게 변환이 되는거지??  이동 회전 이동 회전 로컬회전 역회전 역이동 역회전 역이동  이동회전을 곱해야 하지만 다렉에선 곱하는 순서가 반대(열우선, 행우선 때문).  따라서 TR을 구하고 싶으면 XMMatrixMultiply(RT)를 해야함    보이는 것처럼 UpperArm이 안쪽으로 굽혀지려면 y축을 기준으로 회전을 해야함    중간에 Result 값 보면 분명 y축으로 회전해야된다고 잘 나오는 것도 알 수 있음.  근데 toParent 역행렬만 해주면?? x축으로 갑자기 바뀐다. 대체?    ???????????????????????????????  아!      아ㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏ????????    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  아 어이없어. 새로 함수를 만들어줘야겠다……      내 참나 어이가 없어서 진짜.    예아  로컬 변환 값이 z축 값이 들어가긴 하는데 그럼 z축이 바뀌었거나 기타 등등……  걍 다 돌리고 딱 저 부분만 저렇게 하자.  뭔가 이상해서 함수 일단 다 돌리고 인자도 돌리고 다시 확인 들어감.    x축으로 회전했다는 뜻이거든요.  MakeFromXYZAngle() 함수 원래대로 돌리고 다시 함.    y축으로 돌긴 돌았는데 -90도가 아니라 90도 돌았음.      저렇게 해줘야 원하는 값이 나옴.  아니 사실 지금 내가 뭘 하고 있는건지도 모르겠다. 일단 상황 정리하기    A는 x축으로 90도 회전하는 행렬(키 값 예시가 될 것)  B는 z축으로 90도 회전하는 행렬(이것의 역행렬을 구해서 곱할 예정이니까 -90도 될 것임)  저걸 곱하면 (0, 0, 1)의 점이 (-1, 0, 0)이 되야 함.    잘 됨. 이 때, AngleFromQuat()의 인자를 저렇게 받아야 정확한 회전각을 알 수 있음.  그니까 일단 저 함수는 맞다.  내가 지금 90도 회전이면 오른손 법칙 사용해서 회전방향 정하고 값 예상하고 하고 있는데    이렇게 해야 x축 기준 90도 회전하고 하는 값 나옴(안 그럼 반대임).  여기까지 일단 고정된 상태로 손 디버깅을 해보도록 하겠습니다…,,,   |  |  |  |  | | --- | --- | --- | --- | | RootNode |  |  |  | | 무언가… 있음… | Bip001 |  |  | |  | footstep | Pelvis |  | |  |  | L Thigth | Spine | |  |  | L Calf | Spine1 | |  |  |  | Spine2 | |  |  |  | L Clavicle | |  |  |  | L UpperArm |   일단 이론상 위에서 아래로는 이동회전 이고 아래서 위로는 회전이동임.  RootNode {x=0.000000000 y=0.000000000 z=0.000000000 }{x=0.000000000 y=0.000000000 z=0.000000000 }  Bip001 {x=0.000000000 y=-0.000000000 z=20.9413624 }{x=0.000000000 y=-0.000000000 z=-89.9999237 }  Pelvis {x=0.000000000 y=0.000000000 z=0.000000000 }{x=-89.9999237 y=-89.9999237 z=0.000000000 }  L Thigh {x=3.81469727e-06 y=-2.05546803e-06 z=-2.14222956 }{x=179.988358 y=7.94762039 z=179.923492 }  L Calf {x=8.52674103 y=1.86264515e-09 z=0.000000000 }{x=-8.33763225e-10 y=8.53773656e-07 z=-0.144021347 }  Spine {x=1.45887184 y=-0.00169828010 z=2.02551928e-06 }{x=-0.000238436623 y=-7.93944855e-05 z=0.0456256606 }  Spine1 {x=2.13401413 y=-0.00153303333 z=-4.25202273e-09 }{x=-3.45123105e-12 y=-0.000000000 z=0.000000000 }  Spine2 {x=1.92525482 y=-0.00157961808 z=-4.38122782e-09 }{x=-3.45123105e-12 y=-0.000000000 z=0.000000000 }  L Clavicle {x=2.02116394 y=-3.14824283e-05 z=0.727620244 }{x=-0.000355463650 y=-63.2410660 z=179.954697 }  L UpperArm {x=2.13474846 y=-1.13686838e-13 z=0.000000000 }{x=6.08194685 y=-32.1648941 z=-1.76954758 }  Wow….  이동 회전 이동 회전 이동 회전 Lcl 역회전 역이동 역회전 역이동 역회전 역이동 이런 순서.  여기서 이동회전을 통해 아래로 내려가는게 dressposeInv, 역회전역이동으로 통해 위로 올라가는게 toParent  역회전역이동은 이동회전의 역행렬과 같음!  Lcl x 역회전 x 역이동  = Lcl x 이동회전의 역행렬  = Lcl x XMMatrixInv(&det, XMMatrixMultiply(회전이동)  = XMMatrixMultiply(Inv, Lcl)  = toWorld  최종행렬은 dressposeInv x toWorld  = XMMatrixMultiply(toWorld, dressposeInv)  그럼 toParent는 LclTranslation과 LclRotation이고 역회전역이동이니까 XMMatrixMultiply(역이동, 역회전)  저것의 Inv가 dressposeInv가 될 것      아니 아예 반대였었는데???????? 여기까진 Lcl을 단위 행렬로 반환했을 때 아무 문제 없었음    최종 행렬 반환도 반대였네 아니 대체  이제 남은 문제는 Lcl 행렬 구하기가 전부.  내 공간에서 몇 도 회전했는가가 중요.  RThigh는 (-0.012, 172.052, -0.077) (-7.803, 178.486, -79.079) 이렇게 변화함.  저건 역회전 값. 회전값은 저 반대일 것. (7.803, -178.486, 79.079)  변화한 값이 없을 때는 결과가 단위행렬이 나와야 함.  XMFLOAT3 xmf3R1 = { -0.0116424561, 172.052383, -0.0765075684 };  XMFLOAT3 xmf3B = { 179.988358, 7.94762039, 179.923492 };  여기서 xmf3R1이랑 xmf3B랑 행렬로 만들고 나서 행렬 비교해보면 똑같은데  그럼 하나 역행렬로 만들어서 곱하면 되는거 아냐? 그럼 단위행렬 나오지 않을까?    거의 단위행렬 나옴. 비슷하네.  ???????????????????????????  R0은 회전값을 부호반전시킨 상태.  Xmf3b는 그대로.  그 둘로 행렬을 만들었는데  나는 (-90, -90, 0) 회전하는 행렬이랑 (90, 90, 0) 회전하는 행렬이랑 곱하면 단위행렬 나올 줄 알았는데…  여차저차 단위행렬 나오게는 함.  근데 문제는 다리가 0번 키(기본 위치)일 때도 단위행렬이 안 나옴. 아니 장난하나.    왜 달라………………………………….  혹시나 하고 b를 전치행렬로 해서 해보니까 그것만 오차 적어지고  -90, -90, 0으로 테스트해보니까 바로 박살남.    왜 z축의 부호가 반대지? 다른 애들도 이런지 한 번 보자.    Bip001의 경우 값은 같지만 부호가 달려있으므로 넣음.                  얘는 40번과 마찬가지로 값이 아예 다르게 들어감  {x=-179.988205 y=-7.94745159 z=179.923462 }  혹시나 하고 40번 애랑 해봤는데 걍 둘 다 다름    뭐가 됐든 처음 상태에선 단위행렬이 나와야 한다.  아니 근데 값이 다른건 37, 40번 뿐인데 나머진 정상이어야 하는거 아냐?    앗 조용히 있겠습니다.  근데 40번 행렬의 초기값과 본의 값으로 나온 결과는 y축으로 15도 회전시키는 행렬인데 저건 그런 느낌이 아닌데?  z축으로 5도 회전시키는 행렬을 만들고 Lcl에서 그걸 반환하게 하면  모든 본들이 z축 기준으로 5도 회전할 테니까 z축도 알 수 있고 변화도 알 수 있을 듯.    아무리 봐도 z축 기준 5도 회전이 아닌데.  Z축 기준으로 회전하랬더니 y축 기준으로 하고 있음.  아 이젠 주석을 좀 지워야겠다.      정말 머리로 안 되니까 컴퓨터로 빌드 때려박는 무식한 짓을 하지 말았어야 했는데  그동안 저지른게 있어서 완성하면 깃에 올릴려고 했는데 더 이상은 관리하기 너무 힘들어져서 커밋 들어감. | | | |
| 작업 예정 내용 | | | |
|  | | | |
| 기타 | | | |
|  | | | |