|  |  |  |  |
| --- | --- | --- | --- |
| MORPHOSIS 개발일지 18차 | | | |
| 기간 | 2019-07-15 ~ 2019-07-29 | 작성자 | 신재욱 |
| 작업 내용 | | | |
| 충돌처리 부분에 테스트 중인 부분 있다. 나중에 확인 바람.  라이팅으로 넘어가자. 필요한 라이트는 앰비언트 라이트, 디렉셔널 라이트, 포인트 라이트 정도  예전에 3D 겜플할 때 갖고 있던 라이트 hlsl 파일을 꺼내왔다.    유휴 과거의 유산    뚝딱뚝딱  실행해보면서 hlsl 문법적인 오류를 수정하여 적용 안한 상태에서 실행되는 것 확인(뭔가 오류 있으면 그 전에 터지니까…)  지금 쓰고 있는 컨버터와 임포터는 다행스럽게도 binormal과 tangent를 포함한 normal 정보 전반을 전부 갖고 있었고 Mesh 만들 때만 추가해주면 된다.  현재 노멀맵을 안 갖고 있으므로 노멀맵에서 읽어오는 것은 불가능하다. 일반 오브젝트는 binormal과 tangent가 필요 없다.(애니메이션 할 때나 노멀 방향 바뀌어서 써야하니까) hlsl에서 노멀맵이 있는 오브젝트와 없는 오브젝트를 구분해서 그려야 하는가? 어떻게?  일단 노멀맵 없는걸 기준으로 하고 그려보자. 예전에 받아서 쓰고 있는 박스도 노멀 정보는 갖고 있을 것. 확인하기 쉽게 메쉬 사이즈 변경하는 함수도 넣어야지.  파일 경로를 다시 예전으로 돌렸다. 내가 왜 그랬을까.  포인터 사용을 줄이려고 하던 중에 내가 객체 하나마다 메쉬나 애니메이션 컨트롤러를 하나씩 할당해주고 있었음을 깨달았다. 전역 변수로 포인터 벡터를 만들고 거기서 관리하게 하자. 임포터에서 넣어주면 될 것이다.  Extern과 관련된 문제가 있었는데 C를 배울 때 스스로 제대로 짚고 넘어가지 않았던 부분이라 계속 자질구레한 실수를 한다. Define.h에서 extern으로 선언하고 한 곳에서 그냥 선언하기. STL 컨테이너라서 헷갈린 듯 하다.  텍스처 리스트 추가 완료. 지금보니 g\_TextureList와 같이 변수명을 쓰고 있었는데 다른 전역 상수들과 너무 헷갈린다. g\_vecTexture 로 변경하고 다른 상수들 이름도 변경하는게 좋을 듯 싶다.  여기까지 했는데 뭔가 매끄럽게 진행된다 싶어서 보니까 그동안 계속 걸리던 오류 중단점들이 안 걸린다! 아니!! 설마!!!    룰루    모델 이름을 내가 직접 다 알고 있어야 한다는 점이 아쉽긴 한데(세트로 관리하고 싶다)  모델 세트를 만들까? 중요한거 아닌거 같은데 걍 나중에 생각나면 하기  애니메이션 클립과 애니메이션 컨트롤러까지 벡터로 관리하게 변경하였다.  그랬더니 다시    그 오류가 남. 사실 고쳐진 적은 없었던 거고 내가 착각하고 있던 것이 아닐까? 내가 깃으로 올려놨었는데 그거가지고 한 번 확인해보는 것은 괜찮을 듯.  이전 커밋 버전에서 브랜치를 파서 힙손상 버그가 발생하는지 테스트했고 발생하는 것을 확인. 아.  노멀값 가지고 라이팅을 해보자. 애니메이션 메쉬는 본의 움직임에 따라 normal 값을 막 변경해줘야 하니까 우선은 안 움직이는 기본 메쉬로만 라이팅 적용해보기.  메쉬를 생성할 때 노멀 값을 가져야 함. 우선 임포터로 가자.    노멀값을 넣어줌. 이젠 HLSL로 올려야 하니까 PSO 세팅하는 곳으로 가자.    노멀값을 HLSL로 올려주게 설정함. 이젠 hlsl로 가자.    라이팅 적용되는걸 볼 수 있음. 빛 방향이 뭔가 이상한데?  테스트하려고 종이박스 하나만 크기를 좀 키워볼까 하면서 과거에 UI 메쉬 크기를 자유롭게 조절할 수 있도록 작성했던 코드를 찾아서 보는데 생각보다 과거의 내가 너무 신기하게 짜서 잠시 멍하니 보고 있었다. 아예 HLSL로 올리는 정보를 덧씌우고 있었다. 생각해보니 이게 맞네. 쉽게 해결될거라 생각했었는데.    하여간 크게 렌더링 함.  밉맵 레벨이 얼마나 떨어지길래 좀만 멀어져도 텍스처 해상도가 개판나는지는 모르겠지만 일단은 넘어가자.    저 뒷면은 검고 앞면은 두 면이 같은 밝기고 윗면이 중간 정도로 어두운걸 보면 빛 방향이 (-1, 0, 1)인 모양인데 나는 그렇게 설정한 적이 없다. 계산에서 미스났든지, 노멀값에서 미스났든지, 하여간 어디서 미스가 남.  하지만 그것은 나중에 해결하자. 이젠 애니메이션 메쉬에 노멀 적용해봐야 할 때.  노멀벡터는 애니메이션 행렬로 변환할 때 뭔가 따로 처리해줘야 했던 것 같다.    룰루  보니까 스케일 변환이 있을 경우에 한정되는 듯. 나는 스케일 변환은 하지 않고 있으므로 상관 없을 것 같다.  메쉬를 생성할 때 노멀값과 혹시 모르니 바이노멀과 탄젠트까지 갖게 해줘야 한다.  좀만 자고 해야지  바이노멀이랑 탄젠트가 어디서 쓰이는거지    바이노멀 굳이 안 전해줘도 되나본데????? 일단 노멀이랑 탄젠트만 올리자.    애니메이션 메쉬에서 받도록 설정  이젠 PSO에서 HLSL로 올릴 때 이런 정보들 올라가욧 해줘야 한다. PSO로 가자.    순서는 항상 잘 맞춰줘야 한다. 이제 HLSL에서 헤더를 보자. 올리는 정보를 변경했으므로 구조체 선언도 달라져야 한다.    위의 InputLayout()에서 정해준 시멘틱 네임을 그대로 따라야 한다. 이제 실제 버텍스 쉐이더를 보자!    지옥에서 올라온 석영 같은 무언가가 되었다. PSO에서 올릴 때 오프셋 설정을 잘못 준거 같음. 이젠 덧셈도 계산기한테 아웃소싱 줘야하는 때가 됐다.    아이고  일단 라이팅 함수를 사용하여 색을 바꾸는 것까진 된다. 제대로 고칠 시간이 됐다.      너무 무서운데  일단 노멀을 탄젠트 바이노멀 노멀 행렬 만들어서 변환시킨 후 그걸로 라이팅 계산을 한 모습.  빛 받는 방향이 이상한건 노멀값이 잘못됐거나, 노멀 변환이 잘못됐거나, 라이팅이 잘못됐거나 중에 있을 것.  노멀값에는 문제 없을거라고 일단 믿고(맥스에서 뽑아오는거니까) 노멀 변환과 라이팅을 확인하도록 하자. 노멀 변환은 제대로 된 라이팅이 선행되지 않을 경우, 확인하기 어렵다. 따라서 라이팅을 먼저 보자.    까먹을까봐 설정 저장해둠.    프리 소스 정말 최고야. 빛만 테스트하려면 Lerp()에서 절반으로 보간해주는 값을 한 쪽으로 보내면 될 것.    빛 색을 좀 지저분하게 넣긴 했는데 이렇게 보니까 너무 별로라서 어이없다. 참고로 약간 노란색으로 넣었다. 보이는건 파란색이고.    빛 방향을 (0, -1, 0)으로 했는데 왜 (0, 0, 1) 방향에서 빛이 오지.    노멀값을 월드변환을 안 해줬나본데?    의심가는 부분을 일단 적어두고 찾아보자.  현재 상황은   * 오브젝트는 y축 기준 -방향으로 회전하고 있음. * 빛의 영향을 받는 부분은 오브젝트의 회전방향과 불일치함(빛은 한 방향에서 고정되어있고 오브젝트는 회전하면 빛을 받는 부분이 빛의 방향이어야 하는데 오브젝트의 회전에 따라 빛을 받는 면이 바뀜. 근데 오브젝트 회전 방향이랑 같지도 않음. 설명하기 어려운데 방향이 다르다?) * 노멀값과 빛의 방향(빛이 x축으로 +로 가면 그 반대를 노멀이랑 내적)을 반대로 한걸 내적해서 그 값이 일정 이상이면 하얗게 그리고 아니면 검게 그리게 설정하여 위와 같은 결과가 나왔다. * 빛의 방향이 위에서 아래로 내려쬐게 하면 전부 하얗게 보여서 테스트가 안 된다.(내 카메라 위치상) * 따라서 빛의 방향은 (1,0,0)   노멀값이 이상하니까 이런 문제가 생기는 것. 노멀값과 연관된 부분은 다음과 같다.   * 맥스에서 가져올 때 * FBX 파일을 컨버팅하여 내가 사용할 양식으로 변환할 때 * 내가 사용할 형식으로 재구성된 에셋 파일을 본 클라이언트에서 임포트할 때 * 임포트한 메쉬 파일을 버텍스에 저장할 때   일단 문제가 두 가지 있었는데 하나는 노멀 값을 dot 계산할 때 normalize()를 해주지 않은 것이고 다른 하나는 노멀값의 y와 z축을 바꿔주지 않은 것이다.  지금 문제는 본의 방향이 메쉬의 방향과 반대라서 클라이언트에서 메쉬의 z값을 -1 곱해주는 것도 있고    에휴  디폴트 메쉬는 회전을 시켜도 노멀이 정상 적용 된다. 이제 애니메이션 메쉬를 처리해야 함. | | | |