|  |  |  |  |
| --- | --- | --- | --- |
| MORPHOSIS 개발일지 15차 | | | |
| 기간 | 2019-06-11 ~ 2019-06-19 | 작성자 | 신재욱 |
| 작업 내용 | | | |
| 빠르게 고치고 들어가자. 나는 메쉬 렌더링과 애니메이션은 기본적인 오브젝트에서 다 해도 된다고 생각한다. 오브젝트는 모델을 가져야 하고 텍스처는 모델에서 관리하도록 변경한다. 모델은 하나의 메쉬와 하나의 텍스처, 하나의 충돌박스를 가지도록 한다. 충돌박스는 다이렉트 X 충돌 라이브러리의 바운딩박스와 디버그 모드에서 보여줄 테스트 메쉬를 갖도록 한다.  근데 텍스처를 모델에서 사용하게 하면 한 오브젝트가 여러 모델을 가질 때, 같은 텍스처를 사용한다고 하더라도 다시 Set을 해줘야 하는 일이 생긴다. 다이렉트 X에서 드로우콜과 셋은 성능을 많이 떨어뜨리는 원인 중 하나이므로 Batch 처리의 대상이 된다. 위의 방식을 쓰면 Batch 효율이 약간 저하되는 감이 있으나, 기존 방식과 프레임 차이가 크게 날 것으로 생각되진 않으므로 새 구조로 변경해도 괜찮을 것으로 본다.  CImporter에선 파일 이름을 받아서 해당 파일의 내용을 읽어서 반환해야 한다. ImportModel() 함수의 인자를 파일 이름만 전달해줄까 하다가 파일 하나에 여러 모델이 있을 수 있다는 점을 생각하여 파일 이름과 오브젝트의 포인터를 같이 전달해주어 안에서 여러 모델을 추가할 수 있도록 변경하였다.  Mesh는 기본적으로 텍스처를 사용하는 것을 전제로 한다. (그럼 디버그용 충돌박스를 그릴 때도 hlsl으로 텍스처 관련 정보를 보내겠네? -> 보내진 않더라도 메모리에 처음 할당할 때, UV 정보를 갖게 되는 문제가 있음 -> 즉, 메모리 사용량의 문제, 계산 속도 등으로는 문제가 없을 듯 -> 근데 생각해보면 InputLayout은 클래스의 멤버변수 선언이랑 그런 순서를 따라간다는 느낌이었는데?)  위의 의식의 흐름을 거쳐서 일단 기본 Mesh는 모양만 갖고 있는걸로 정했다. 필요한 클래스들을 정리하면 모양만 가질 가장 기본적인 Mesh, UI 용도로 텍스처를 띄우기 위한 UV만 가지면 되는 TexturedMesh, 조명 계산을 위해 Normal 값을 가져야 하는(그리고 NormalMap을 사용하게 되면 그걸 위해서 Tangent랑 Binormal값도 가져야 하는) IlluminatedTexturedMesh, 애니메이션 계산을 위해 boneIdx랑 boneWeight를 가져야 하는 AnimatedMesh가 있다. 이름이 너무 길고 쓰임새가 불분명하다. 각자 WireMesh, UIMesh, ObjectMesh, AnimatedMesh로 별명을 달아주도록 하자.  이런 식으로 별명을 사용할 수 있다. 따라서 다음과 같이 왼쪽을 부모클래스로 Mesh - TexturedMesh - IlluminatedTexturedMesh – AnimatedMesh 와 같이 상속을 받게 하자.  메쉬 부분은 일단 대충이라도 정리하였다.  임포터에서 메쉬를 만들려면 디바이스 정보랑 커맨드리스트를 갖고 있어야 한다. 따라서 생성자에서 그 둘을 받게 하고, 기본 생성자는 쓸 수 없게 ‘= delete’를 붙여주었다. 메쉬 생성을 하고 모델에 넣어주는 부분까진 문제 없었지만 텍스처를 현재 파일에서 읽어오지 못 하므로 그 부분은 지금은 어쩔 수 없이 직접 어떤 오브젝트가 어떤 텍스처를 사용할 지 넣어주어야 한다. 지금은 급하니까 텍스처를 외부에서 생성하고 해당 텍스처를 파라미터로 전해주는 방식으로 진행하자.  오브젝트는 렌더링만 되면 되는 오브젝트(UI 등), 충돌이 되야 하는 오브젝트, 충돌도 되고 이동도 되는 오브젝트가 필요하다. 관리해야 할 것이 많다. 오브젝트 매니저도 있으면 좋겠고, 레벨 데이터를 관리할 레벨 오브젝트도 필요하고, 충돌맵에서 플레이어만 충돌하거나, 팀 별로 충돌을 안 하거나 하는 것도 필요하다.  오브젝트 매니저를 만들자! 원래 있었었는데 안 쓴다고 지웠었다. (애니메이션 한다고) 오브젝트 생성 시에 필요한 수순은 다음과 같다.   * 서술자 힙을 생성한다. 서술자 힙은 서술자를 저장하는 연속적인 메모리 영역으로 간단하게 서술자들을 저장하고 있다고 생각하면 된다. * 서술자 힙을 생성하려면 서술자 힙 설명 구조체의 내용을 채운 뒤, 그걸 넘겨줘야 하는데 그 구조체에서 서술자의 개수는 오브젝트의 개수와 쓸 애니메이션 행렬의 개수, 텍스처 써야하니까 쉐이더 리소스 뷰까지 추가해서 정해주면 된다. * 상수버퍼를 생성해준다. 이 때, 버퍼 사이즈는 무조건 256단위로 맞춰야 한다. 저 때 말고는 맞춰줄 필요 없음. ID3D12Resorce 포인터 변수에 CreateBufferResource()를 해준다. 그 리소스 변수를 다른 포인터 변수에 Map을 해준다. 그럼 그 Map 된 변수로 접근 가능하다. 거기에 값을 쓰면 그게 VRAM에 올라가는 듯. * 그 다음엔 상수 버퍼 뷰를 생성해준다. 상수버퍼 뷰는 오브젝트 개수만큼 반복하며 서술자 힙의 시작 주소에서 자기 위치만큼 이동한 주소를 가지고 CreateConstantBufferView()를 하면 된다. * 그 다음엔 오브젝트 만들어주고 그 오브젝트에게 아까 만든 상수 버퍼 뷰의 주소를 준다. 얘를 가지고 SetGraphicRootDescriptorTable()에 사용할 거임.   지금은 다른 생각 하지 말고 매니저에서 레벨 오브젝트랑 프롭 오브젝트랑 플레이어 오브젝트랑 투사체 오브젝트만 관리하게 하자. 근데 애니메이션 들어가는 오브젝트는 애니메이션 행렬 올리게 전부 64개씩 갖고 있어야 하는거 아냐? CB\_OBJECT 내용을 바꿀까?  아니 근데 솔직히 오브젝트가 모델 갖고 있으면 충돌 처리 할 수도 있고, 애니메이션 할 수도 있고, 이동할 수도 있고, 그냥 오브젝트에 다 때려박으면 안 돼? 된다.  하다보니 너무 보기 싫어서 오브젝트의 이동 방식을 변경하기로 함. 키 입력을 받는 부분과 처리하는 부분을 따로 둬야 한다. 모델 별로 충돌 박스를 갖는게 제일 좋지만 그럴 경우, 오브젝트 이동 시에 어떻게 이동시켜줘야 하는지가 문제가 된다. 모델 별로 Offset을 갖게 하면? 근데 문제는 스키닝 메쉬는 메쉬가 하나라서 충돌박스도 하나만 나온다. 아 이게 문제가 되네……………………….. 일단 지금은 모델의 충돌박스를 신경쓰지 말고 하자.  지금 보니까 오브젝트랑 플레이어 오브젝트랑은 따로 둬야겠음.  지금 어느 정도까지 된건지 잘 모르겠다.  쉬고 싶어서 구상도 작업함.    확실히 그려보니까 엄청 좁은게 느껴짐. 하지만 튜토리얼 용이라고 하면 모든 것이 해결될 것이다. 필요한 오브젝트 종류와 기타 등등. 경사로 처리가 관건이 될 것. (경사로 처리는 아래쪽에)  문제가 생기는데 이게 무엇이 원인인지도, 어떤 결과가 나오는지도 잘 모르겠다. 고질적인 문제였던 WindowUpdate()에서 걸리는 오류에 추가로 ifstream으로 파일을 읽어올 때 일정 확률로 안 되는 문제가 있다. 같은 코드를 두 번 돌려보면 한 번은 되고 한 번은 안 되는 문제.  어떻게 저런 일이 생길 수 있을까. 초기화가 안 된 변수가 문제? 동적할당하는 부분에서 문제?  검색하는데 아무런 영양가 없는 한국어 포스트만 잔뜩 나오는게 너무 싫어서 검색엔진 기본 언어를 영어로 바꾸고 비주얼 스튜디오 언어 설정도 영어로 변경했다. 이제 오류 문구를 검색하기 훨씬 편해졌다. 실제로 검색했더니 바로 결과가 나왔다.  <https://stackoverflow.com/questions/6811882/0xc0000005-access-violation-reading-location-0x00000008>  null 포인터를 역참조 한다는 것 같다. (아니면 내가 접근할 수 없는 곳을 접근하려 했다던지) 또는 포인터 객체 중에 무언가가 null을 반환한다던지. 아니면 array 범위를 넘어갔다던지.    이 부분을 수정하고    이 부분을 수정했다. (주석 처리된 부분이 수정 전)  인덱스 문제가 틀림없다. 감이 쎄게 온다.  그 뒤로 인덱스 관련된 부분들을 손 봐줬다. PSO가 애니메이션 전용을 쓰고 있어서 텍스처 전용으로 변경해줬다. (지금 애니메이션 안 쓰니까)  생각해보니까 프롭이 애니메이션이 들어가는거랑 안 들어가는거랑 구분해서 쓴다 쳐도 FBX에서 애니메이션 없는 것은 키프레임 하나에 단위 행렬을 넣어줘야 하는데 너무 귀찮다. 프롭은 애니메이션 없는 걸로 하자. 나중에 추가하면 되지. 아니면 플레이어만 애니메이션을 갖는게 아니라 애니메이션 되는 프롭이랑 아닌거랑 따로 따로 나눠놓고 써도 되고. 후자가 좋은 생각인 것 같다. 하여간 여기서 다시 한 번 커밋.  텍스처가 초록색으로 보여지는 문제가 있다. 나는 픽셀 쉐이더에서 초록색으로 띄우라고 강제한 적 없다. 따라서~ 이것은 텍스처를 불러올 때의 문제거나, 불러온 텍스처를 로드할 때 문제거나, 메쉬의 UV 좌표의 문제거나 많은 가능성이 있다. 근데 거기서 사용한 텍스처는 갈색 바탕이라서 초록색이 생길 일이 없다.    이렇게 생김.    파일에 UV 좌표도 다 잘 들어가는걸로 봐선 텍스처를 Set할 때 문제가 있는게 틀림없음. 기존에 되던 텍스처로도 시험해 봤는데 그것도 초록색으로 나오는걸 보면 그 쪽 잘못이 맞다.  테스트하다가 카메라 오프셋 고정되어 있는 것도 찾았다.  알고보니 텍스처를 쉐이더 리소스 뷰로 안 만들어주고 있었다.    색만 들어간 느낌인데 이런 결과가 나올 수가 있나.  다른 텍스처를 넣어줬는데도 같은 결과가 나옴.  지금 CAnimatedMesh로 만들어주고 있는데 그걸 InputLayout에서 넘기면서 다른 값을 읽어서 보내는게 틀림없음. 그래픽 디버깅을 해보려고 했는데 비주얼 스튜디오에서 지원 종료했다고 한다. 아니.  <https://devblogs.microsoft.com/pix/download/>  저기서 그래픽 디버거를 받을 수 있다.  하지만 유의미한 결과는 못 얻었다. (쓰기 어려워서)  근데 텍스처는 띄움. (?)    혹시나하고 확인해봤는데 CIlluminatedTexturedVertex 의 멤버변수 선언 순서는 위치, 노멀, UV좌표 순이었다. 따라서 CTexturedIlluminatedPSO는 위치, 노멀, UV좌표 순으로 하는게 맞다. (지금 쓰고 있는 PSO) 근데 CAnimVertex는 위치, UV좌표, 본 가중치, 본 인덱스 순으로 변수 선언이 되어있고, 지금은 CAnimVertex를 쓰고 있으므로 PSO의 InputLayout의 순서와 다를 수 밖에 없다. 실제로 InputLayout은 클래스의 변수 선언 순서에 따라 값을 읽어가고 있었던 것. 책에는 안 나와있던데 아니 대체. 하여간 순서를 변경하여 해결하였다.  여기서 커밋 또 함.  애니메이션이 동시에 될 수 있게 하자! 이제 관건은 오브젝트마다 애니메이션 행렬을 따로 올리는 것. 이걸 위해선 오브젝트를 손봐야 한다.  렌더 걸기 전에 pd3dCommandList -> SetGraphicsRootConstantBufferView( 3, interpolatedMatrixResource -> GetGPUVirtualAddress()); 를 해줘야 한다. 애니메이션 올릴 때 뭐가 필요하더라.  일단 오브젝트에서 애니메이션을 쓸 수 있도록 클립을 받게 변경한다. 임포터의 구조를 변경하면서 애니메이션 클립을 받아오는 함수를 잠시 주석처리 해놨었는데 그 부분을 수정한다.  수정해줘야 하는 것은 오브젝트별 XMMATRIX \*pMapped = new XMMATRIX[64]랑 루트 시그니처에서 행렬을 서술자 테이블로 변경해줘야 한다.  행렬을 hlsl로 올려주는게 또 문제가 됐다. 어떻게 해야 할까? 텍스처도 하나만 띄워지는 문제가 있다.   * 텍스처 하나만 띄워지는건 예전 코드 참고해서 해결하였다.   서술자 테이블이 아닌데 어떻게 했지. 한 번 올리고 하나 그리고 이런 식으로 했나. 근데 그럼 익스큐트 할 때 한 번에 빠져나가서 안 될텐데.    아. 드디어.  맨 위에 원인을 잘 모르겠는 문제가 다시 발생했다.  DLL과 EXE 관련된 문제라는 말도 있던데(MT와 MD 이런 내용) 내가 겪고 있는 문제의 현상과 그걸로 짐작가는 원인과 해결법이 분명하지 않으니 너무 어렵다. 일단 계속 보자.  Windbg 같은 툴도 있다고 하던데 설치해서 봤으나 지금 쓰기에는 너무 어려워보인다.  힙 문제는 나중에 하고 지금은 레벨을 먼저 만들자. 좋은 생각.  투사체도 이제 되지 않을까?  애니메이션 컨트롤러에서 이전 클립을 저장해둘 것. 그래야 현재 클립의 IsLoop가 false일 때 그걸로 돌아간다. 애니메이션 클립 저장할 때 IsLoop도 따로 저장해줘야 할 것.  "PlayerIdle"  "PlayerRun"  "PlayerFire"  "PlayerStartJump"  "PlayerEndJump"  "PlayerDied"  일단 조건 줘서 애니메이션 클립 변경하는 것도 잘 된다. 투사체 먼저 만들고 충돌박스 정리한 다음에 레벨 만들자.  투사체는 미리 만들어두고 IsAlive로 관리하는 걸 목표로 하자. 투사체는 이동하는 속력, 방향, 중력의 영향을 받는지 여부 등이 필요하다.  계속 종료할 때 릴리즈에서 걸리는 문제가 있는데 잘 모르겠다.    한 클래스로 작업할 생각은 없었지만 계속 하다보니 한 클래스에 온갖게 다 들어가서 CObject를 CPlayer와 CProjectile로 나눴다. 마우스를 누르면 미리 만들어둔 풀에서 가져오게 만들었는데 잘 된다. 기쁨!  이제 방향으로 이동하게 해야 한다. 팀을 정할 변수도 필요할 것 같다.  지금 플레이어와 투사체를 만드는 방식은 플레이어 풀을 만들고, 투사체 풀을 플레이어 수에 비례해서 만들고 하는 식이다. 그렇다면 미리 팀도 다 정해둘 수 있지 않을까? 팀은 변경되지 않지 않을까? 뭔가 막 반사하고 그런게 아니라면 중간에 팀이 바뀔 일은 없을 것이다.  갑자기 궁금해서 constexpr와 #define의 차이점을 검색해보고 왔다.  <https://stackoverflow.com/questions/42388077/constexpr-vs-macros>  이동도 잘 된다. 이제 충돌하면 없어지거나, 일정 시간이 지나면 없어지거나, 일정 거리를 가면 없어지거나 하는 걸 만들어주고 거기에 추가로 플레이어에게 fireRate 같은걸 줘서 발사 수 제한을 둬야 할 것.  일단은 일정 시간이 지나면 없어지게 만들자. 마음 같아선 스레드를 하나 추가하고 타이머를 관리하고 싶으나 멀티스레드 프로그래밍을 지금 바로 하기엔 내가 너무 부족하다. 타이머 변수를 만들고 Update()로 관리하도록 하자.  이제 해야 하는 것은 충돌 처리, 생성 위치 등.  드디어 대망의 충돌처리를 살릴 때가 왔다.    하고 싶었던 것은 플레이어가 많은 충돌박스를 갖고 그게 본의 움직임을 따라 움직이는 거였는데 지금 거기까지는 무리인 것 같다. 지금은 충돌체를 모델에 넣어놨었는데(방금 그거 하려고) 일단 객체로 다시 옮기도록 하자.  충돌체 구조체는 충돌체(바운딩 박스나 바운딩 스피어 등)와 그걸 보여줄 메쉬를 가지게 해야 한다.  충돌체는 객체의 중심을 기준으로 존재한다.    이런 문제가 생긴다.  이제와서 모델의 피봇을 수정하는 것은 힘들다. 충돌체를 위로 올려야 한다.  이동할 때 대각선으로 움직이면 속도가 빨라지길래 정규화한 값으로 움직이게 했다.  카메라 회전 시에도 문제 없도록(오프셋이 고정이라 가까워지거나 하는 문제가 있었다) 수정했다.  충돌 시에 이동은 무효되지만 충돌박스를 옮겨주던 것은 되돌리지 않던 점을 수정하였다.      게임의 기본적인 UnitStandard가 정해졌다. 20이 1 Unit이다.  달릴 때 충돌 박스 너무 안 맞는데 지금은 어쩔 수 없다. 패스.  BoundingBox 등에서 쓰이는 Extents는 반지름이 아니라 지름의 의미로 이해하면 좋다.  하여간 이제는 충돌도 이전보다는 정확하게 된다.  "Bip001 R Hand" 를 사용해서 투사체가 나가는 방향이나 기타 등등을 정하게 수정하였다. 지금 보니까 애니메이션이 좌우가 반대인데??  사람이 100미터를 10초에 뛴다고 치고 키 큰 사람이 대충 2미터로 보면 저기서 1유닛이 20이고 캐릭터 키가 2유닛이다. 그럼 1유닛이 1미터고 1초에 대략 10유닛은 뛰어야 한다. 이동속도를 바꿔주었더니 애니메이션이 좀 더 자연스러워졌다.    일단 테스트를 위해 간단하게 만든 레벨 메쉬.  맵을 띄웠더니 이젠 디버그 모드에서 실행 한 번 하기가 너무 어려워졌다. (실행이 안 된다는 뜻) 릴리즈 모드에서는 다행히 아무런 문제가 없으나 아마 드러나지만 않은 모양이고 속에서 터지고 있을 것.  김재경 교수님과 윤정현 교수님과 상담하였다.  지금 뭐가 가장 중요한지, 그걸 하려면 뭘 해야 하는지 등등을 생각해보자. 지금처럼 구멍난 곳을 떼우기만 하는 식으로는 완성 못 할 듯. 무엇을 해야 하는지 찾고, 어떻게 해야 하는지 찾아야 한다. 너무 무작정 들이박는 방법으로는 시간만 낭비.  게임이 아니라 프로그램으로서 기본인 것들:   * ~~오브젝트 렌더링~~ * ~~오브젝트 충돌처리~~ * 씬 변경 * 레벨 충돌처리 * UI 렌더링 * UI 피킹(버튼) -> 이거 생각해보니까 굳이 피킹 안 해도 됨. 화면에서 비율로 마우스 클릭 찾아내기. * 사운드 재생   기술적으로 필요한 요소나 고려해야 할 것:   * NPC를 할건지 네트워크를 할건지 정해야 함. (기간 내에 할 수는 있나?) * NPC든 네트워크든 어떤 방식으로 사람들이 구현했는지 알아보고 도움을 구하기   송인희 교수님과 상담하였다.  나는 몰랐는데 UI는 다른 중요한 것들에 비해 우선순위가 떨어진다. 빨리 맵 충돌처리랑 그런걸 해야 게임이 된다고 하셨다. | | | |