|  |  |  |  |
| --- | --- | --- | --- |
| 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……………………….. | | | |