|  |  |  |  |
| --- | --- | --- | --- |
| MORPHOSIS 개발일지 22차 | | | |
| 기간 | 2019-09-10 ~ 2019-10-08 | 작성자 | 신재욱 |
| 작업 내용 | | | |
| 이 참에 다 바꾸자.  밖의 파일을 읽어서 초기화를 하도록 하자.  렌더링은 프리펩 방식을 사용한다.  이미 다 터질대로 터졌으므로 하고 싶은걸 하고 그걸 포트폴리오로 쓰자.  이쁜걸 렌더링 할 수 있으면 좋겠다.  겸사겸사 그게 되면 나머지도 해결 되거나 적어도 큰 문제는 없어질 것.  하고 싶은건 씬을 만들고 프리펩을 읽어오고(무슨 메쉬를 쓰는지, 무슨 텍스처를 쓰는지) 씬에 두고 렌더링하는 것.  <https://blog.naver.com/PostView.nhn?blogId=cccani&logNo=221232979712&proxyReferer=https%3A%2F%2Fwww.google.com%2F>  공부할 때 참고했던 블로그.  ~~프리펩 데이터 내용~~  ~~헤더~~  ~~애니메이션 컨트롤러 이름(있으면 이름, 없으면 비워두든지 NULL 하든지)~~  ~~모델 리스트~~  ~~메쉬 이름~~  ~~텍스처 이름~~  ~~텍스처 이름~~  ~~메쉬 이름~~  ~~텍스처 이름~~  ~~텍스처 이름~~  ~~메쉬 이름~~  ~~텍스처 이름~~  ~~충돌체 리스트~~  ~~충돌체 정보(offset, extent, orientation)~~  ~~충돌체 정보~~  디퓨즈맵, 노멀맵, 스페큘러맵, 글로스맵 이런 선은 텍스처 추가하고 기존 HLSL에서 만져주면 될 것 같음.  맥스 라이선스가 끝났어~~~      스케일 맞췄고  일단 노멀맵부터 확인  범프매핑하고 싶어  한 모델에서 여러 텍스처를 쓸 수 있게 만ㄴ들어야 함.  지금은 한 모델이 하나의 메쉬와 하나의 텍스처만 쓰고 있음.  메쉬 하나당 여러 텍스처를 쓸 수 있어야 함.  그래야 알비도, AO, 하이트, 노멀, 러프니스 등 다양하게 쓸 수 있으니까.  지금 모델 클래스는 SetMesh(Mesh\*)와 SetTexture(Texture\*)를 받고 있는데  직접적으로 포인터 주지 말고 index로 관리하게 변경(저번에 했던걸 다시)해주고  SetMesh(int) //g\_vecMesh[idx]  SetAlbedo(int) // g\_vecTexture[idx]  로 하는게 좋을 것 같음.  아니면 TextureSet 클래스를 만들고 텍스처 세트 하나에 여러 텍스처를 두는 것도 좋을 것 같음.  그러면 g\_vecTextureSet[idx] 해서 idx를 얻어온 다음에 그리면 되니까.  메쉬 렌더링 할 때마다 Set 함수를 많이 부르면 성능이 떨어지지 않을까?  프리펩에 모두 같은 텍스처 사용 옵션을 주고 그게 True면 렌더링할 때  처음에 한 번만 셋 하고 그 뒤론 걍 메쉬만 업로드해도~~  이런건 나중에 생각하자.  지금 해야 하는 것은 여러 텍스처를 한 모델에 넣는 것.  텍스처 세트는 좋은 생각 같다.  가장 작은 크기의 기본 텍스처들을 0, 1, 2, 3, 4 순으로 올려두고 디폴트 값으로 주는게 좋을 것 같다.  다 집어치우고 일단 디퓨즈랑 노멀을 같이 올리는걸 목표로 하기  동시에 두 텍스처를 올릴 수 있는가??  지금 텍스처를 올리는 방식은 루트 시그니처에서 SRV를 하나 만들어서 디퓨즈 맵을 올려주고 있는 상황(레지스트 T2로). 적당히 레지스터 t5 쯤에 하나 넣어주면 되지 않을까??  DescriptorTable의 registerSpace 값을 어떻게 잘 주면 HLSL에서 register(t3, space0); register(t3, space1); 과 같이 쓸 수 있다고 책에는 적혀있는데 저걸 해서 뭘 어떻게 하는건지 모르겠다.  Space 쓰지 말자! 선배한테 여쭤봤는데 서술자를 늘려서 넣었다고 하셨다. 그렇군~~~~        위의 코드는 내 프로젝트에 있던 부분. 아마 내용이 크게 다르지는 않을 것이다.  선배 코드의 맨 아래 부분을 보면 서술자의 개수가 10개로 되어있는데 그럴 경우 HLSL에서 다음과 같이 사용한다.    이렇게 쓸 경우 마테리얼 정보를 따로 업로드를 해주고 이번 메쉬는 어떤 마테리얼을 쓰는지, 그 마테리얼은 몇 번째 텍스처를 쓸건지 등을 알고 있어야 한다. 마테리얼 업로드 부분은 귀찮아서 정리해버렸는데 이렇게~~  텍스처를 t4로 빼고 다른걸 2번에 넣은 다음에 거기서 서술자 개수를 많이 주고 사용하는게 좋을 것 같다.  나는 어짜피 매번 렌더할 때마다 업로드를 하게 될 것 같으므로.  (위에서 저렇게 10개로 미리 배열을 만들어 둔 이유는 최대 10개까지만 사용할 것 같아서 라고 하심)    기존의 코드를 수정함. GetType() 함수는 Albedo인지 AO인지 Normal인지 등을 반환한다.  Mesh랑 Model이랑 Prefab이랑 Texture랑 다 같은 헤더에서 관리하고 싶다.      노멀맵을 추가하고 범프매핑까지 진행했다.  반사된 값이 뭔가 이상하길래 야매로 넣어주고 있던 탄젠트 방향을 정반대로 바꿔주자 해결되었다. | | | |