**2018180020 박재우(클라이언트) 6주차 기록**

**모델 띄워보기**

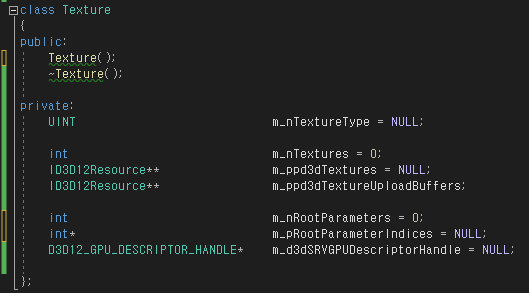
우리 프로젝트에 샀던 에셋을 추출한 파일을 읽어 화면에 띄워 보려 한다.

현재 진행 상황으로는 육면체 몇 개가 자전하는 모습만 나오는 상황이다. 여기서 일단 모델을 띄우기 위해 필요한 요소들을 정리한다.

1. 모델이 포함한 정보를 저장할 객체를 만들어야 한다. (텍스쳐, 머터리얼, 애니메이션 등등..)
2. 모델이 포함한 정보를 읽을 함수를 만들어야 한다.
3. 모델을 렌더링할 때 사용할 쉐이더를 만들어야 한다.
4. 모델이 가지고 있는 리소스를 쉐이더로 넘겨줘야 한다.

**모델 띄우기 1번**

머터리얼, 텍스쳐, 애니메이션 이 세가지의 클래스가 필요하다. 머터리얼은 이미 만들어져 있고 애니메이션은 따로 더 생각해야 할 부분이기 때문에 우선 텍스쳐를 만들어 준다.

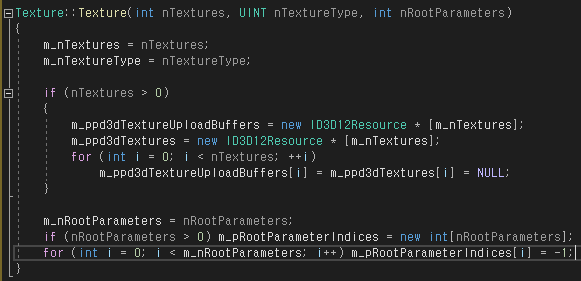


우선 간단하게 리소스 타입, 텍스쳐 개수, 텍스쳐 리소스, 쉐이더에 넘겨주기 위한 버퍼, 루트파라미터의 인덱스 번호와 핸들을 만들어 주었다. 루트 파라미터의 인덱스의 배열이 필요한 이유는 모델의 텍스쳐가 하나가 아니라 노멀맵, 메탈릭 맵 등등 여러 텍스쳐가 존재할 수 있기 때문이다. 서로 다른 텍스쳐는 각각 다른 루트파라미터에 넘겨주기 위해 인덱스의 배열로 선언한다.

텍스처 객체를 생성할 때 텍스쳐 리소스 개수, 텍스쳐 타입, 루트파라미터 인덱스를 넘겨 받아서 멤버 변수들을 설정해 주어야 한다. 따라서 생성자의 매개변수를 다음과 같이 설정해준다.

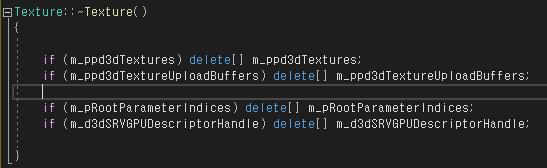


그에 따른 정의는 다음과 같이 작성해 주었다.



텍스쳐 개수와 타입, 루트파라미터의 개수는 바로 대입하여 저장해 주고 텍스쳐 개수에 맞는 리소스와 버퍼를 선언을 해주며 전부 NULL로 초기화 해준다. 루트 파라미터의 인덱스도 개수만큼 생성을 하고 인덱스 번호는 0번 부터이기 때문에 -1로 전부 초기화를 해준다.

소멸자에서는 동적으로 생성된 모든 값을 지워주는 작업을 해준다.

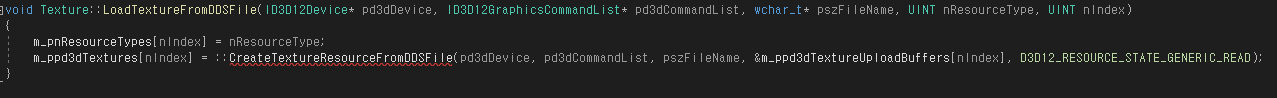


각 변수에 NULL이 들어 갈 수 있으므로 if문으로 한번 체크를 해준다.

다음은 텍스쳐를 로드하는 부분이다. 텍스쳐로 사용하는 파일은 dds파일이고 이 파일을 로드하는 기능은 3D게임프로그래밍 시간에 제공되었던 DDSTextureLoader12 파일을 이용했다.

LoadTextureFromDDSFile의 선언과 정의는 다음과 같고 CreateTextureRescourceFromDDSFile 함수는 stdafx.cpp에 정의가 되어있다.





이 함수는 DDS파일 자체를 열어 읽는 함수 이므로 실제 모델의 정보를 읽어 해당 모델과 맞는 텍스쳐를 찾는 일도 해야한다. 따라서 LoadTextrueFromFile이라는 함수를 만들어 bin파일을 읽는 작업을 해준다. 이 함수는 모델을 로드하는 함수에서 텍스쳐를 로드할 때 부를 함수이다.



선언은 이와 같이 했다. 함수의 구조는 먼저 <AlbedoMap>과 같이 텍스쳐의 부분을 나타내는 문구가 나오면 호출이 될 것이고 다음 문구가 텍스쳐 파일의 이름이다. 텍스쳐는 항상 Model/Texture/… 경로에 저장이 될 것이기 때문에 텍스쳐 파일 앞에 해당 경로를 붙히고 뒤에 .dds 를 붙혀 파일 경로를 완성한다. 이 경로를 앞서 만든 LoadTextureFromDDSFile함수를 이용해 파일을 읽는다. 파일 이름 앞에 @가 붙은 경우가 있는데 이 경우는 파일이 이미 한번 열렸고 계층 구조를 따라 내려가서 또 같은 이름의 텍스쳐를 열어야 할 때 붙는다. 이미 한번 읽은 텍스쳐는 이미 SRV가 만들어져 있으므로 이 텍스쳐를 넘겨주는 루트 파라미터의 인덱스 번호를 찾아 넘겨주면 된다.