

Homework #5: Hello Assimp

김준호

Abstract

본 과제에서는 Open Asset Import Library (Assimp)를 사용하여 여러 개의 모델을 불러와 여러 시점의 카메라로 렌더하는 프로그램을 작성한다.



1 OPEN ASSET IMPORT LIBRARY (ASSIMP)

Assimp는 다양한 파일 포맷으로 정의된 3D model 데이터를 손쉽게 메인메모리의 자료 구조로서 불러올 수 있는 공개형 오픈소스 라이브러리이다. 파일의 format과 관계 없이 Assimp가 정의하는 자료 구조로 import 되기 때문에 모든 format에 대해서 동일한 방식으로 프로그램을 작성할 수 있다. 보다 자세한 내용은 Assimp 공식 문서를 참고한다.

1.1 Assimp 설치

아래의 명령어를 사용해 Ubuntu 환경에 Assimp를 설치한다.

```
$ sudo apt-get install libassimp-dev assimp-utils
```

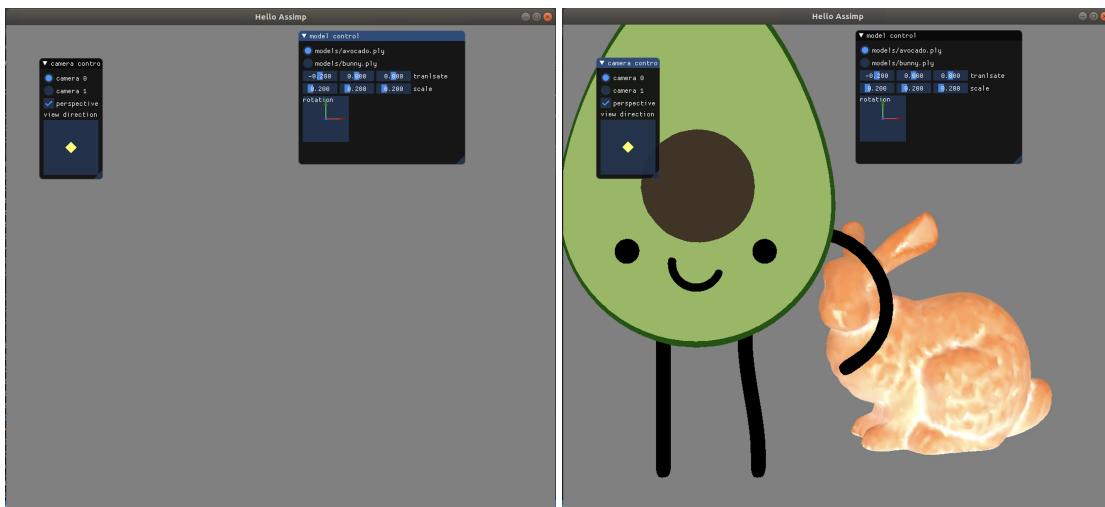
1.2 참고 링크

아래 3 개의 LearnOpenGL 투토리얼을 참고하여 과제를 완성하도록 한다. 특히 첫 번째 링크에 설명되어 있는 Assimp의 자료 구조 부분을 자세히 살펴 보면 과제를 행하는 데 큰 도움을 얻을 수 있다.

- 1) LearnOpenGL - Assimp ([link](#))
- 2) LearnOpenGL - Mesh ([link](#))
- 3) LearnOpenGL - Model ([link](#))

2 과제 가이드

Assimp를 통해 모델을 불러오면, Assimp가 정의한 자료 구조로 모든 정보가 저장된다. Object.cpp, main.cpp 코드를 완성하여 사용자의 입력에 따라 물체에 변환율 적용해보자.



(a) 프로그램 처음 실행 시

(b) 완성된 프로그램

Fig. 1: 프로그램 예시

과제로 주어진 skeleton 코드를 컴파일하여 실행하면 Fig. 1(a)와 같은 프로그램을 마주할 수 있다. Assimp를 통해 물체를 불러오는 과정은 `load_asset()` 함수에, `info.txt` 파일을 통해 장면(scene)을 초기화하는 과정은 `init_scene_from_file()` 함수에 구현되어 있다. 완성된 프로그램에는 여러 개의 모델과 여러 개의 카메라가 존재한다. ImGui위젯으로 각 모델과 카메라를 선택하고, 키보드와 ImGui위젯을 통해 선택된 모델과 카메라를 움직일 수 있게 한다. solution 코드의 `//TODO` 부분을 적절히 수정하여 Fig. 1(b)와 같은 프로그램을 완성하는 것이 이번 과제의 목표이다.

`info.txt`에는 장면 정보가 들어 있다. `main.cpp`의 `init_scene_from_file()`에서 `info.txt` 파일을 읽은 후 해당 정보로 장면을 생성할 수 있게 한다. 프로그램을 모두 작성한 후 `info.txt` 파일을 수정하여 보다 많은 모델과 카메라로 실험해보는 것을 권장한다. `info.txt`의 구조는 다음과 같다.

```

2          // 불러올 model 갯수
models/avocado.ply // 불러올 파일의 경로
0.2        // scale
-0.2 0.0 0.0 // translate
models/bunny.ply
0.5
0.2 -0.2 -0.4
2          // 생성할 카메라의 갯수
0.0 0.0 1.0 // position
0.0 0.0 -1.0 // front_direction
0.0 1.0 0.0 // up_direction
0.9 0.3 0.3
-0.806421 -0.339238 -0.484358
0.0 1.0 0.0

```

2.1 과제 세부사항

- `Object.cpp`의 `init_buffer_objects()`, `draw()`, `get_model_matrix()` 함수를 구현한다.
- `main.cpp`의 `render_object()` 함수를 구현한다.
- 현재 장면 정보를 `info.txt` 파일과 같이 텍스트 파일로 내보내는 함수를 구현하고, 키보드 또는 ImGui버튼을 통해 동작할 수 있게 한다. (어떤 키를 사용할지는 자유이다.)

2.2 과제 유의사항

- `Camera` 클래스는 이전 과제에서 본인이 작성한 코드를 사용하도록 한다.
- 키보드를 통해 선택된 물체와 카메라를 움직이는 코드는 미리 구현되어 있다.

3 과제 제출방법(매우 중요!!)

- 본 과제는 개인과제이며, 각자 자신의 코드를 완성하도록 한다.
- 공지된 마감 시간까지 과제 코드를 가상대학에 업로드하도록 한다.
- 과제 코드는 **Ubuntu 18.04 LTS 환경에서 make 명령으로 컴파일 가능하도록** 작성한다.
- 과제 코드는 다음의 파일들을 하나의 압축파일로 묶어 **tar.gz** 파일 형식이나 표준 **zip** 파일 형식으로만 제출하도록 한다. 이때, 압축파일의 이름은 반드시 'OOOOOOOO_HW05.tar.gz' (OOOOOOOO는 자신의 학번)과 같이 자신의 학번이 드러나도록 제출한다.
 - 1) 소스코드 및 리소스 파일들
 - 2) Makefile
- 과제에 관한 질문은 오피스아워를 활용하도록 한다. 오피스아워 이외의 시간에 도움을 받으려면 교육조교(teaching assistant, TA)에게 메일로 약속시간을 정한 후, 교육조교가 있는 연구실로 방문하도록 한다.