

Homework #4: Camera

김준호

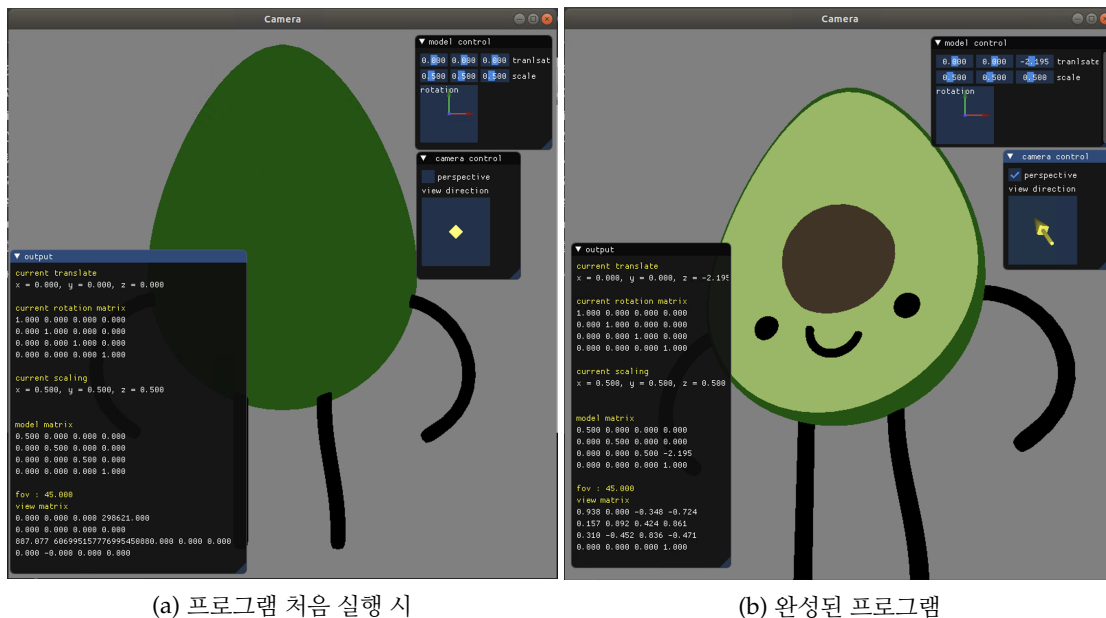
Abstract

본 과제에서는 사용자의 입력에 반응하여 카메라를 조작하는 그래픽스 프로그램을 작성한다. GLFW의 콜백 함수와 ImGui를 사용하여 사용자 입력을 통해 카메라의 위치, 회전, 시야각을 바꾸는 방법을 익힌다.

1 과제 소개

과제로 주어진 skeleton 코드를 수정하여, 사용자에게 입력에 따라 카메라 이동(translation), 회전(rotation), 시야각(field of view)을 변경할 수 있는 프로그램을 완성한다. GLFW의 콜백 함수를 사용하여 키보드 입력과 마우스 입력을 처리하고, 이에 더해 ImGui로 사용자 인터페이스(User Interface)를 구성하는 방법을 실습한다.

2 과제 가이드



(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. 현재 프로그램은 model matrix를 조작하는 기능만 포함되어 있다. 본 과제에서는 주어진 소스 코드를 적절히 수정하여 Fig. 1(b)와 같은 프로그램을 완성하는 것이 목표이다.

Camera.cpp의 함수들과 main.cpp의 `//TODO`를 완성하여 다음 기능들이 프로그램에 구현될 수 있도록 한다. 키보드를 통한 카메라 조작은 FPS 게임의 카메라 조작 방식과 유사하게 구현된다.

- 키보드 D/A 키를 통한 카메라 $\pm x$ 축 방향 이동 조작
- 키보드 S/W 키를 통한 카메라 $\pm z$ 축 방향 이동 조작
- `imGuIZMO.quat` 위젯을 통한 카메라 회전 조작
- ImGui 위젯을 통한 카메라 perspective / orthographic 모드 토글링
- 마우스 scroll을 통한 카메라 시야각 (FoV) 조작 (추가 점수)

카메라는 처음에 (0.0, 0.0, 0.5) 위치에 자리해 있으며, front direction은 $-z$, up direction은 $+y$ 방향으로 설정되어 있다. 키보드 입력 혹은 ImGui 입력에 맞게 카메라의 up, front, right vector들을 적절히 수정하여 view matrix를 생성하면 본 과제를 해결할 수 있다. 과제와 관련된 GLM 함수들은 다음과 같다. 보다 자세한 방법은 LearnOpenGL의 카메라 튜토리얼 (link)을 참고한다.

```

template<typename T >
detail::tmat4x4< T > lookAt (detail::tvec3< T > const &eye,
                             detail::tvec3< T > const &center,
                             detail::tvec3< T > const &up)

template<typename T >
detail::tmat4x4< T > ortho (T const &left, T const &right,
                           T const &bottom, T const &top,
                           T const &zNear, T const &zFar)

template<typename T >
detail::tmat4x4< T > perspective (T const &fovy, T const &aspect, T const &near, T const &far)

```

추가 구현

카메라 스크롤을 통해 zoom in/out을 구현하면, 추가 점수를 부여한다. 관련된 GLFW 함수는 다음과 같다.

```

glfwSetScrollCallback(window, scroll_callback);

void scroll_callback(GLFWwindow* window, double xoffset, double yoffset)
{
    // do something
}

```

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 가능**하도록 작성한다.
- 과제 코드는 다음의 파일들을 하나의 압축파일로 묶어 **tar.gz** 파일 형식이나 표준 **zip**파일 형식으로만 제출하도록 한다. 이때, 압축파일의 이름은 반드시 'OOOOOOOOO_HW04.tar.gz (OOOOOOOOO은 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.