

Homework #2 : TriangleMesh

김준호

Abstract

본 과제에서는 버퍼 객체를 이용해 물체를 그리는 간단한 모던 OpenGL 프로그램을 작성한다. 모던 OpenGL의 정점 버퍼 객체 (vertex buffer object, VBO)와 인덱스 버퍼 객체 (index buffer object, IBO)를 사용하는 예제를 실습하여 OpenGL 버퍼 객체를 활용한 모델 데이터의 표현법을 익힌다.



1 과제 소개

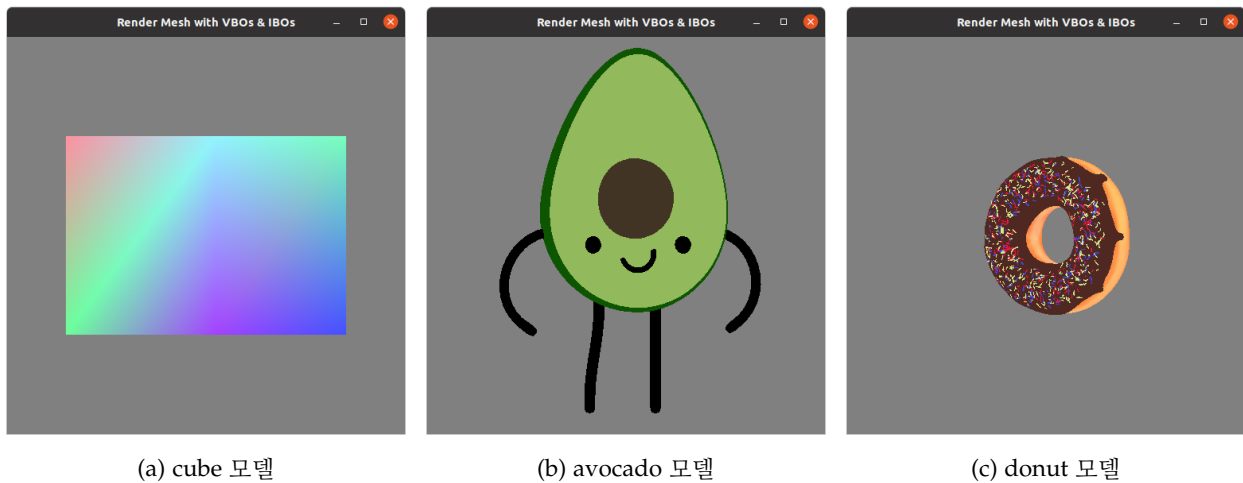


Fig. 1: 완성된 프로그램 예시

과제로 주어진 skeleton 코드를 수정하여, 그림 1과 같이 세 개의 모델에 대해 i) triangle soup와 ii) vertex list & triangles 두 개의 방식으로 렌더링하는 프로그램을 완성한다.

2 과제 가이드

과제로 주어진 skeleton 코드를 컴파일하면 회전하는 cube를 렌더링하는 프로그램을 확인할 수 있다. 현재 cube 모델은 triangle soup 방식으로 VBO를 사용하여 렌더링되고 있다. 본 과제는 main.cpp의 `//TODO` 부분을 적절히 수정하여 키보드 입력에 따라 각기 다른 모델을 각기 다른 방식으로 렌더링하는 프로그램을 완성하는 것을 목표로 한다.

과제로 주어진 skeleton 코드의 models/에는 세 개의 모델에 대해 triangle soup 방식, vertex list & triangles 방식으로 정의된 데이터가 헤더파일로 형태로 코딩되어 있으며, 이들은 모두 main.cpp 파일에 아래의 코드 조각과 같이 포함되어 있다. 이 때, triangle soup 방식은 {model}_triangle_soup.hpp 파일에, vertex list & triangles 방식은 {model}_vlist_triangles.hpp 파일에 해당된다.

```
// model.hpp
#include "models/cube_triangle_soup.hpp"
#include "models/avocado_triangle_soup.hpp"
#include "models/donut_triangle_soup.hpp"
#include "models/cube_vlist_triangles.hpp"
#include "models/avocado_vlist_triangles.hpp"
#include "models/donut_vlist_triangles.hpp"
```

주어진 모델들을 이용하여 아래의 키보드 입력에 따라 각기 다른 방법으로 렌더링하는 프로그램을 완성하도록 한다. 키보드 입력을 처리하는 코드는 main.cpp에 미리 구현되어 있으며, 본 과제는 `//TODO` 부분을 적절히 수정하여 아래와 같이 각기 다른 모델과 렌더링 방식을 선택할 수 있도록 하는 것이 목표이다.

- key 1 : cube 모델을 triangle soup 방식으로 렌더링
- key 2 : avocado 모델을 triangle soup 방식으로 렌더링
- key 3 : donut 모델을 triangle soup 방식으로 렌더링
- key 4 : cube 모델을 vertex list & triangles 방식으로 렌더링
- key 5 : avocado 모델을 vertex list & triangles 방식으로 렌더링
- key 6 : donut 모델을 vertex list & triangles 방식으로 렌더링

아래는 키보드 입력에 따라 렌더링 모드 설정을 변경하는 것과 관련된 코드 조각이다. `key_callback()` 함수 안에서 `g_mesh_model`와 `g_mesh_type`이 어떻게 설정되는지만 참고하고, `key_callback()` 함수의 코드는 수정하지 않는다.

```
enum MeshModel { kCube, kAvocado, kDonut };
enum MeshType { kTriangleSoup, kVlistTriangles };

MeshModel g_mesh_model = kCube;
MeshType g_mesh_type = kTriangleSoup;

// ...

void key_callback(GLFWwindow* window, int key, int scancode, int action, int mods)
{
    if (key == GLFW_KEY_1 && action == GLFW_PRESS)
    {
        g_mesh_model = kCube;
        g_mesh_type = kTriangleSoup;
        update_buffer_objects();
    }
    // 이하 생략
    // ...
}
```

과제는 `main.cpp`의 두 함수, `void update_buffer_objects()`, `void render_object()`에서 `//TODO` 부분을 적절히 수정하면 쉽게 해결할 수 있다. 다만, triangle soup 방식으로 렌더링 시 VBO를, vertex list & triangles 방식으로 렌더링 시 VBO + IBO를 사용해야만 한다.

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_HW02.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.