

0. Parsing

기본적으로 모든 기능 구현에 앞서서 obj파일에서 내가 원하는 정보를 읽어와야 vao와 vbo에 정보를 넘겨주어 렌더링을 할 수 있기 때문에 obj파일의 정보를 parsing하는 함수를 구현했다. 항상 drop을 했을 때는 가장 마지막으로 drop된 파일만 렌더링을 해줘야 하므로 obj_file의 정보를 담고있는 list들을 초기화 시킨다. Parsing 함수는 Single mesh rendering mode에서 terminal에서 출력해주어야 하는 정보를 parsing과정에서 계산하며, drawarrays를 할 때 필요한 삼각형의 개수 또한 계산한다. 가장 중요한 파싱의 경우에는 v,vn,f 세 개의 문자에 대하여 입력이 들어왔을 때 각각 vertex pos, vertex norm, face info에 저장하게 하였다. Face의 경우에는 올바른 렌더링을 위하여 저장 순서를 고려하여 주었다.

1. Single mesh rendering mode

Screen에 obj파일을 drop하면 파일을 받아들이기 위해 glfwSetDropCallback 함수를 이용하여 미리 만들어진 drop_callback 함수를 호출하게 했다.

<Draw_callback 함수>

drop이 되면 만약 animaiting 모드였다면 그것을 꺼주고 드랍 받은 파일을 열어주며, 그 파일을 parsing 함수로 넘겨준다. 그리고 parsing이 종료되면 parsing된 obj파일의 정보를 과제 명세에 맞게 출력해준다.

<prepare_vao_obj 함수>

Parsing된 정보를 바탕으로 vertices matrix를 초기화해주고 vao와 vbo를 구성해준다.

<main 함수>

이제 완성된 vao와 parsing할 때 계산했던 삼각형의 개수를 이용하여 화면에 rendering 해준다.

2. Animating hierarchical model rendering mode

실행영상 링크 : <https://youtu.be/xnl7clYoaUA>

제가 제출한 과제 파일의 형식에서는 obj_files에 제가 과제를 구현할 때 사용한 obj 파일들이 들어있습니다. 상대주소로 소스코드가 있는 디렉토리에 obj_files가 있으면 실행될 수 있게 구현해 두었습니다. 감사합니다.

우선 이것을 구현하기 위하여 초기화해주어야 할 두 가지 과정이 있다. 첫번째는 h버튼을 눌렀을 때 animating mode로 렌더링하는 화면을 보여주는 것이고, 두번째는 렌더링할 객체들을 미리 vao에 넣어주는 것이다.

<key_callback 함수>

H버튼의 입력이 들어오면 만약에 drop된 파일이 rendering 중이었다면 그것을 꺼주고 animating을 활성화해준다.

<prepare_vao_'Nodename'> nodename에는 각 node의 이름이 들어간다

Draw때와 동일하게 parsing을 해주고 vao와 vbo를 구성해준데, 삼각형 개수가 각 node별로 다르므로 list에 값을 저장해준다.

<main 함수>

Sing mesh rendering을 해줄 때에는 drop을 했을 때 파일을 받고 vao를 구성해야 하므로 그 때에 vao를 만들어주지만, animating 모드에서는 내가 가지고 있는 obj 파일을 가지고 하는 것이므로 while 문이 시작되기 전에 미리 vao를 다 만들어 둔다. 그리고 node를 hierarchical 구조로 만들어준다. Tree가 가장 부모이고 자식으로 table과 maple을 갖는다. Table은 pikachu와 umbreon을 자식으로 가지고, maple은 fist와 foot을 자식으로 갖는다.

Node별 움직임

Tree – 제자리에서 빙글빙글 돈다.

Table – Tree를 기준으로 빙글빙글 돈다.

Maple – Tree 위에서 위아래로 움직인다

Pikachu, umbreon – table을 기준으로 빙글빙글 돈다.

Fist – maple을 기준으로 빙글빙글 돈다.

Foot – maple을 기준으로 빙글빙글 도는데, 위 아래로도 움직인다.

3. Lighting & ETC

Shader에 phong illumination과 phong shading을 이용하여 객체에 색을 칠해준다. Single에 경우에는 기본적으로 초록색으로 출력해주게 만들어 두었다. Animating에 경우에는 객체별로 node를 만들 때 색을 지정해 두었다. 그리고 여러 개의 광원을 이용하라는 과제의 명세를 따르기 위하여 light pos를 (3,2,4), (-3,2,-1) 2곳에 지정해 주었다.

<key_callback 함수>

z버튼의 입력이 들어오면 default로 1로 설정 되어있는 wire 변수를 1과 -1로 변화시켜준다.

<main 함수>

Wire값이 1일 때는 solid mode를 지원하고 -1일 때는 wireframe mode를 지원한다.
