Computer Graphics Project3:

Bvh Viewer

2019073181 심기호

1. Which requirements you implemented

A : manipulating camera and grid line:전 과제 보고서에 있으니 자세한 설명은 생략한다.

B : Load a bvh file and render it

i.bvh file open by drag-and-drop to my viewer window

-drag callback함수를 이용해서 파일이 화면 위로 떨어졌을 때 open(path,'r')을 이용해서 파일들을 읽었고 drop_callback이 불렸는지를 알려주는 dropped변수를 만들고 f또 다시 drop_callback이 불리는지를 알려주는 변수인 first변수를 만들어서 평소에는 dropped과 first가 모두 0이다가 drop_callback이 불렸을 때는 둘다 1로 변하고 한번 렌더링 하고 나서는 first변수는 메인함수에서 rendering을 하고 0으로 바뀐다. 그리고 다시 다른 bvh file이 들어올 때는 first가 1로 변하고 새 정보를 그리게 된다.

ii. two rendering modes

-line rendering

: Line rendering은 움직이는 점들의 위치를 vao로 넘겨서 GL_LINES로 그렸다.

-box rendering

: Box rendering은 길이 1짜리 단위 cube를 만들어서 그 cube를 점과 점 사이 중앙으로 translate하고 점과 점을 잇는 벡터와 각을 구해서 rotate 시켜서 각을 맞춰준다. 그리고 y 축방향으로 점과 점사이의 길이/2만큼 scale을 해준다. Vao_cube에 normal vector들도 넣어줘서 그 벡터들도 회전하게끔 해줘서 phong illumination과 phong shading을 구현했다.

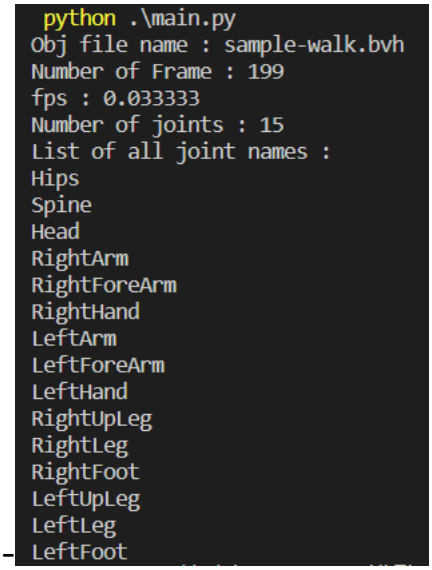
iii.bvh file reading and render the skeleton

: 처음 bvh file을 읽어들이고 node 정보들을 만들어주는 prepare_node_list()를 만들어서 node정보들을 만든다. 처음에는 움직이면 안되기 때문에 shape_transform에는 glm.mat4()를 넣어주었다. 그리고 한 joint와 붙어있는 joint의 위치정보들을 구해서 templist에 넣고 그 list를 기반으로 vao_bvh들을 만든다. 그리고 그걸 draw_node함수를 불러서 그리게 된다.

iv. Animate the loaded motion if spacebar pressed

: spacebar가 눌리면 한 frame마다 channel_list를 순회하면서 node별로 set_joint_transform함수를 불러서 위치정보나 회전정보를 계속 덮어주었다. 그리고 한프레임별로 정보를 준비하면 main함수에서 그리고 time.sleep(fps)로 fps시간만큼 자게했다. 그리고 motion정보를 다 읽게 되면 다시 motion정보의 처음부터 읽으면서 다시 움직이게 했다.

v. print out the info of the bvh file



: Bvh file을 읽을 때 정보들을 저장하고, joint 의 경우 joint_list를 만들어서 ROOT나 JOINT로 시작하는 경우에 그 다음 단어들을 list에 저장해서 출력했다.

2. Hyperlink to video

https://www.youtube.com/shorts/k9VEku3rh4Q