

Computer Graphics Project 3

Bvh Viewer

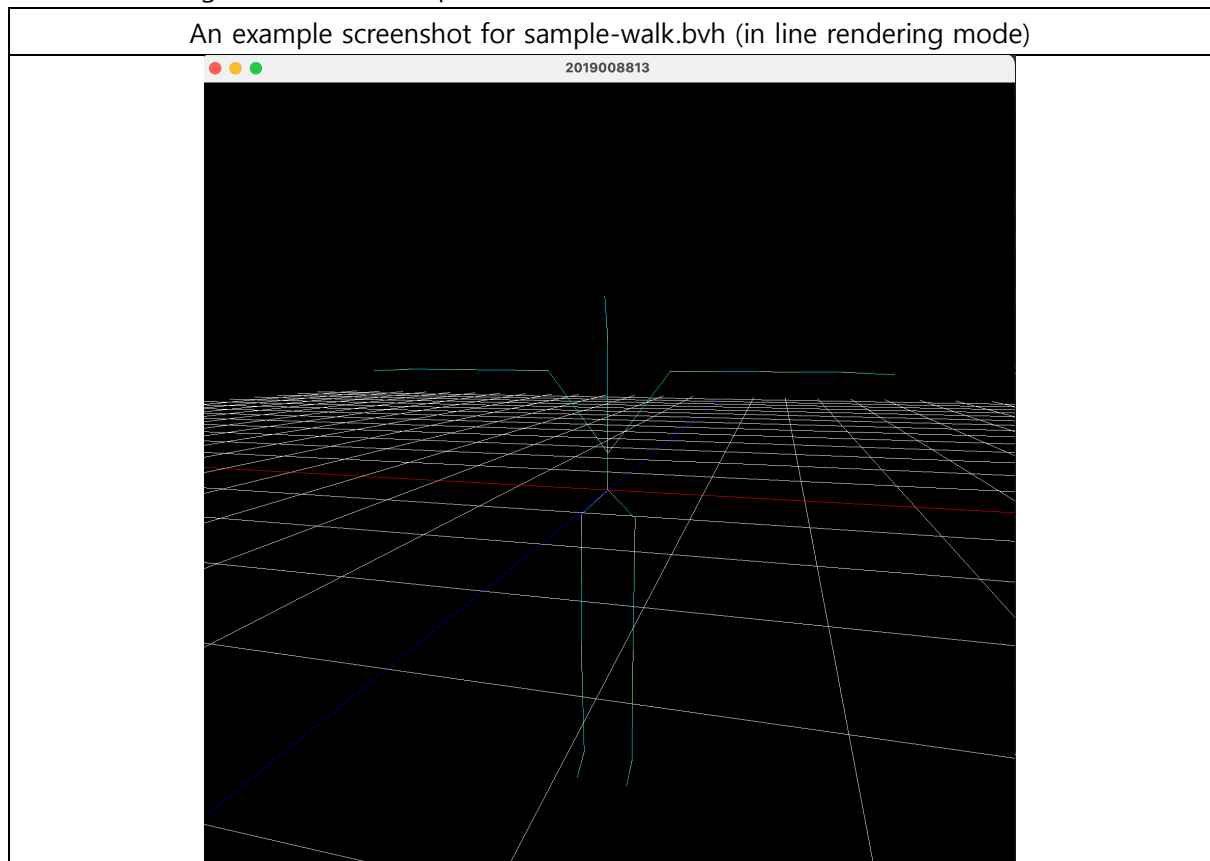
2019008813 문원찬

1. Manipulate the camera same as Project1 + draw grid plane

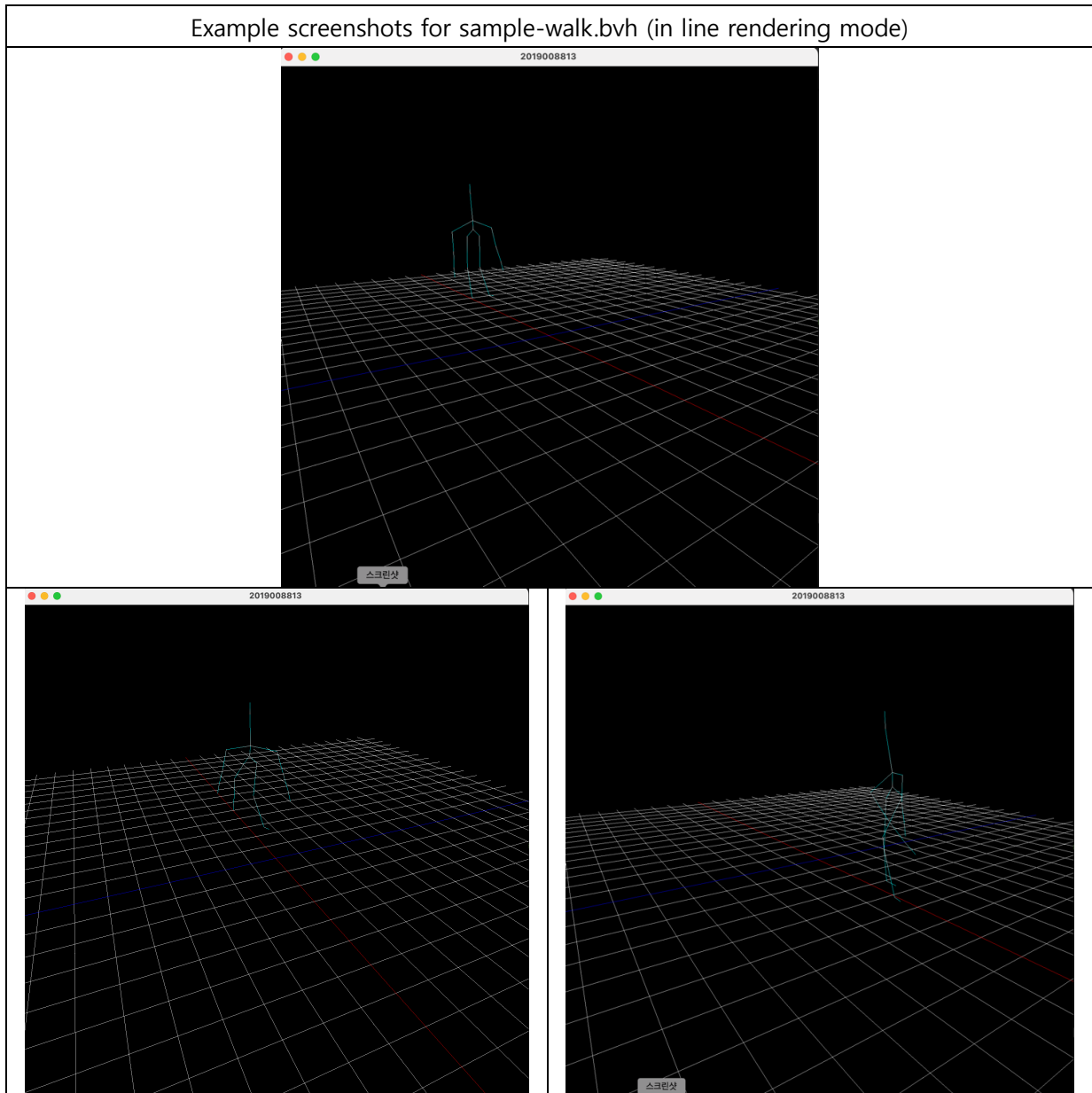
project1에서 구현한 카메라 조작 방법을 그대로 옮겨오고 grid plane 또한 project 1과 동일하게 그렸다.

2. Load a bvh file and render it (Requirements)

glfwSetDropCallback 함수를 사용해 bvh 파일을 window에 drag-and-drop하면 작동하게 구현했다. 기본적으로 Project2에서 obj 파일을 로드하는 방법과 같다. bvh 파일 drag-and-drop시 path에 따라 파일을 열어 split을 통해 렌더링에 필요한 정보를 탐색했다. 요구사항에 맞게 한 번에 하나의 파일만 렌더링하며, 새로운 파일을 넣을 시 그 파일로 렌더링한다. 렌더링은 Line rendering과 Box rendering을 지원한다. 각각 key 1, key 2를 누르면 모드에 진입한다. Box rendering에선 Phong illumination, shading을 지원하는 shader를 사용했고 Line rendering은 frame을 그릴 때 사용했던 shader를 사용했다. 두 모드는 프로그램 실행 중 언제나 바뀔 수 있게 'line_sel' 변수를 통해 구현했다. 처음 drag-and-drop시에는 skeleton을 그리게 Hierarchy부분만 사용해 Node를 구성했다. joint의 끝부분인 End site도 정상적으로 연결되도록 그렸다. 아래 그림은 line rendering mode에서의 sample-walk.bvh 파일의 모습이다.



이후 스페이스바를 누르면 'move_sel' 변수를 사용해 bvh파일의 MOTION부분을 불러와 움직이게 만들었다. glfwGetTime()함수와 glfwSetTime()함수를 적절히 활용해 시작 frame부터 frame time에 맞게 장면이 넘어가고 마지막 frame이 끝나면 반복되도록 구현했다. 다음은 sample-walk.bvh의 예시이다.



bvh 파일을 drag-and-drop시 리스트를 사용해 stdout(console)에는 File name, Number of frames, FPS, Number of joints, List of all joint names에 대한 정보가 출력되도록 구현했다.

추가적으로 bvh파일의 offset이 너무 크다면 key 3, key 4를 통해 사이즈를 조절해 파일을 확인 할 수 있도록 편의성을 더했다. key 3을 누르면 0.1배 key 4를 누르면 10배로 사이즈를 조절할 수 있다. 구현에 대한 자세한 설명은 코드의 주석에 존재한다.

3. Hyperlink to the video

<https://youtu.be/tNsKAkaOliY>