

# Computer Graphics Project 3

2019092306 구본준

## 1. Implemented requirements

### - Camera and grid

카메라의 경우 project1과 동일한 viewing matrix로 구현하였으며 grid는 빛의 영향을 받지 않도록 하기 위하여 별도의 shader를 사용해 렌더링하였다.

### - Loading a bvh file

*glfwSetDropCallback* 함수를 이용하여 파일을 드래그 후 창에 올려놓았을 때 그 파일의 위치를 알아내어 읽을 수 있도록 구현하였다. 먼저 캐릭터의 크기를 일정하게 유지하기 위하여 rest pose 상태의 키(최고 높이 - 최저 높이)를 알아내 scale factor을  $3/(\text{캐릭터 키})$ 로 만들어 모든 offset에 곱해주도록 하였다. Node class를 이용하여 hierarchical한 구조를 생성하였는데 이때 offset 값을 *link\_transform* 값으로 각각 설정해주었으며 이를 위해 stack을 이용해 bvh 파일의 hierarchy 부분을 탐색하며 계층적 구조를 연결하였다. Motion의 경우 Node class 변수에 리스트 형태로 저장을 하여 이후에 사용할 수 있도록 만들었다. 파일을 불러온 이후 파일 이름, 프레임 수, FPS, 관절의 개수, 관절의 개수를 출력하도록 하였으며 관절에는 root까지 포함하여 출력한다.

### - Two rendering mode

Line rendering mode와 box rendering mode를 '1', '2' 키를 누르면 각각 바뀌도록 구현하였다. Line rendering mode의 경우 캐릭터 구조의 각 점을 매 프레임마다 리스트 형태로 가져와 VBO에 복사하는 방식으로 설계하였다. 이를 위해 파일을 열었을 때 한번 VAO를 초기화 한 이후 애니메이션이 실행될 때 VBO를 계속 업데이트 해주는 함수를 호출한다. Box rendering의 경우 각 joint 위치에 box를 렌더링 하는 것을 기본으로 각 box를 parent joint의 방향으로 scale, translate하여 관절을 잇는 형태의 box처럼 보이도록 구현하였다. 이를 위해 box를 원하는 벡터 방향으로 회전해주는 *box\_rotation* 함수를 만들었으며 함수의 인자로써 원하는 방향 벡터인 (parent joint point) - (child joint point) 값을 대입하였다. Box rendering 모드에서는 phong illumination과 phong shading을 이용하여 (10, 10, 10)과 (-10, 10, -10) 위치에서 빛을 비추주는 광원 두 개를 만들었다.

### - Rendering the skeleton (rest pose)

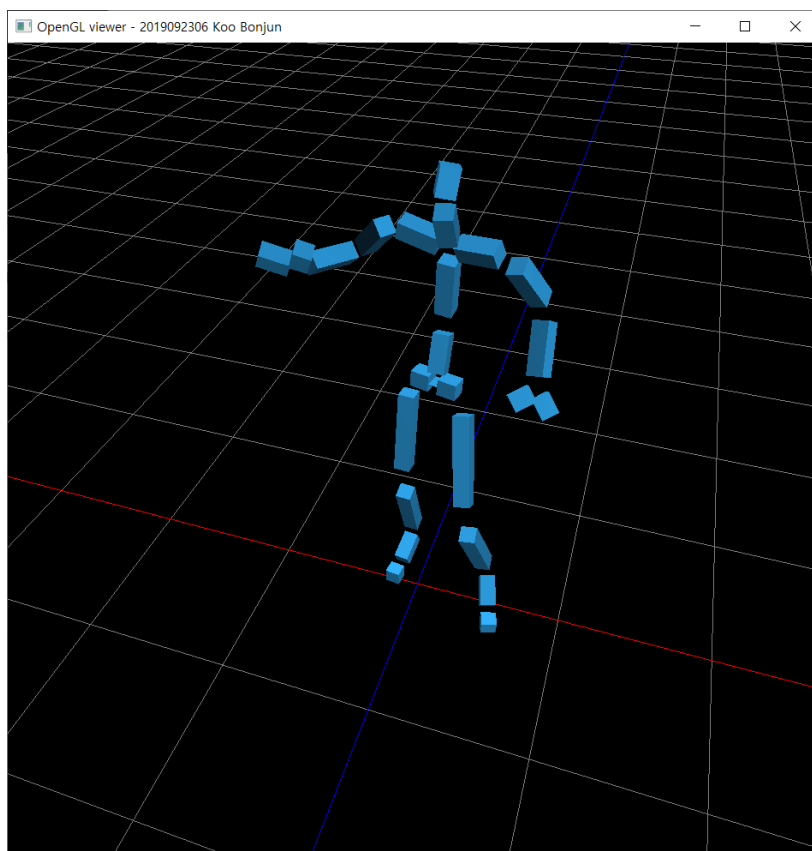
파일을 drag-and-drop하면 joint transform 값을 identity matrix로 설정하여 offset 만으로 표현되는 rest pose를 렌더링하도록 구현하였다. Rest pose 상태에서도 line rendering과

box rendering을 모두 지원한다. 이미 한 bvh 파일이 실행 중이더라도 새로운 bvh 파일을 drop 한다면 이전 파일의 캐릭터는 사라지고 새로운 파일의 캐릭터만 rest pose 상태로 렌더링된다.

### - Animating the loaded motion

Rest pose 상태에서 스페이스바를 누르게 되면 animation이 시작된다. 매 프레임마다 joint transform 값을 motion list의 있는 값을 이용하여 업데이트한 후 캐릭터를 화면에 그려준다. 만약 motion list의 끝에 도달했다면 다시 처음으로 돌아가도록 하여 동작을 반복하도록 구현하였다. Frame time을 맞춰주기 위하여 프레임 마지막에 시간을 측정해 만약 정해진 frame time보다 경과된 시간이 부족한 경우 while문을 통해 기다린 후 다음 프레임으로 넘어가도록 만들었다. 추가적으로 명세에 있지는 않았지만 동작 상태를 자세히 확인할 수 있도록 동작 중에 스페이스 바를 다시 누르면 캐릭터의 동작이 고정되게 구현하였다. 다시 스페이스바를 누르면 pause 상태가 풀리고 동작을 재개한다.

## 2. The animating hierarchical model video.



<https://youtu.be/61HVFox1wo0>