

Computer Graphics Project 2

Obj viewer & drawing a hierarchical model

2019008813 문원찬

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

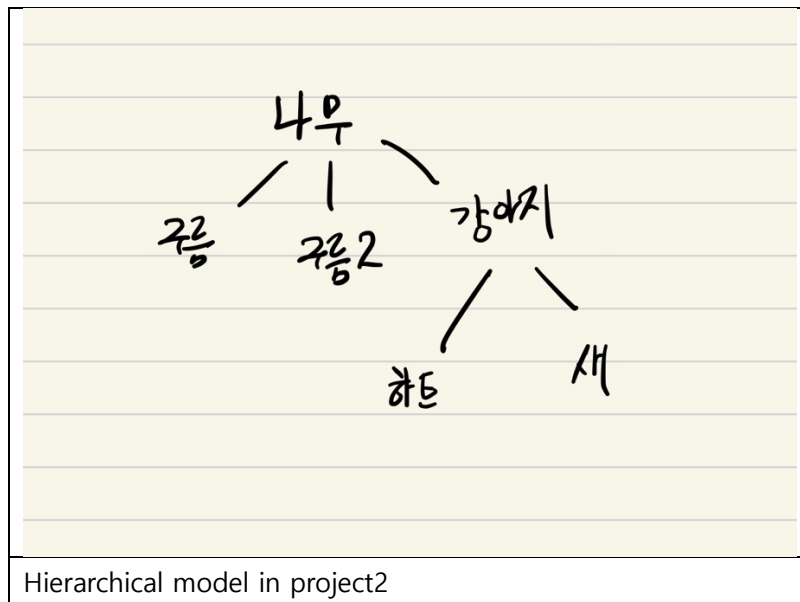
project1에서 구현한 카메라 조작 방법을 그대로 옮겨오고 grid plane 또한 project 1과 동일하게 그렸다.

2. Single mesh rendering mode

glfwSetDropCallback 함수를 사용해 obj파일을 window에 drag-and-drop하면 작동하게 구현했다. drag-and-drop시 파일경로를 따라가 파일을 오픈하여 split을 통해 렌더링에 필요한 정보를 탐색했다. 요구사항에 맞게 한 번에 하나의 파일만 렌더링하며, 새로운 파일을 넣을 시 그 파일로 렌더링한다. 렌더링은 vertex positions, vertex normals, faces information을 가지고 작동하게 만들었다. obj 파일을 drag-and-drop시 terminal에는 Obj file name, Total number of faces, Number of faces with 3 vertices, Number of faces with 4 vertices, Number of faces with more than 4 vertices에 대한 정보가 출력되도록 구현했다.

3. Animating hierarchical model rendering mode

'h' 키를 누르면 해당 모드로 작동하게 설정했다. 프로그램을 처음 실행 시 Single mesh rendering mode이다. 요구사항에 맞게 'h'키를 누르면 해당모드를 실행하기에, 해당 모드 중 다시 'h'키를 눌러도 Single mesh rendering mode로 toggle되지 않는다. 하지만 해당 모드 작동 중 obj 파일을 drag-and-drop시 다시 Single mesh rendering mode로 바뀐다. 해당 모드에선 최소 3가지의 다른 obj 파일들로 작동하게 해야하므로, '나무', '구름', '강아지', '하트', '새' 총 5가지의 다른 obj 파일을 다운로드하고 사용했다. 파일을 불러올 땐 상대주소를 사용해 불러왔다. 계층 구조는 아래 사진과 같다.



총 3 level로 구성되어있으며 leaf 노드가 아니면 최소 2개의 자식 노드를 가지게 했다. '나무'가 root 노드로 1 level에 위치한다. 2 level엔 '구름', '구름2', '강아지'가 위치하고 '강아지'에겐 3 level에 위치한 '하트', '새'를 자식 노드로 갖는다. 모든 노드는 보이도록 했다. 모든 노드들은 부모 노드에 재귀적으로 종속되어 움직이도록 구성하였다. '나무'는 가운데서 y축에 대해 회전하며 '구름'과 '구름2'는 '나무' 위에서 y축 위 아래로 움직인다. '강아지'는 '나무' 옆에서 z축에 대해 앞 뒤로 움직이며 '새'는 '강아지'보다 z축으로 앞, y축으로 위에 존재하여 y축 위 아래로 움직인다. '하트'는 '강아지'와 '새' 사이에 존재하여 scaling을 통해 커졌다 작아졌다는 반복한다. 해당 움직임은 local transformation으로 프로그램에선 parent body에 종속되어 자동으로 움직이도록 구현했다.

4. Lighting & Etc

모든 물체는 Phong illumination, Phong shading으로 렌더링되었다. 'z'키를 누를 시 wireframe mode와 solid mode를 toggle한다. 프로그램을 처음 킨다면 solid mode로 작동한다. 해당 mode 변경은 single mesh, animating hierarchical model 두 렌더링에서 모두 작동한다.

광원은 총 4개 사용하였다. 전부 흰색 광원으로 좌표는 각각 (20,30,0), (-20,30,0), (0,-30,20), (0,-30,-20)이다.

5. Hyperlink to the video

https://youtu.be/rT_k7moHS7M