Obj Viewer & Drawing a Hierarchical Model

2019027001

문 석 원

1. **Single mesh rendering mode의 구현**

Single mesh rendering mode는 obj 파일을 드래그 & 드롭을 하면 해당 obj file의 도형을 render하는 mode로, 드래그 & 드롭한 obj 파일을 바탕으로 varr을 구성하는 코드는 드래그 & 드롭을 감지하는 drop\_callback() 함수에, 그 obj 파일을 render하는 함수는 obj\_render() 함수에 구현되어 있다.

drop\_callback() 함수에서는 single mesh rendering mode인지 확인한 후 맞으면, 먼저 인자로 받은 path로부터 파일명을 추출해내 출력한 수, obj 파일을 열어 한 줄씩 입력 받는다. 각 줄마다 첫 단어가 ‘v’ 이면 vertex들을 저장하는 vertices 배열에, ‘vn’ 이면 normal vector들을 저장하는 normals 배열에 추가하고, ‘f’ 이면 해당 면을 여러 개의 삼각형으로 분할한 후, 각 점의 normal vector와 vertex position을 varr에 추가하여 최종적으로 varr을 구성한다.

obj\_render() 함수에서는 색을 회색으로 설정한 뒤, drop\_callback()에서 구성한 varr과 glDrawArrays함수를 이용하여 obj file에 있는 도형을 render한다.

1. **Hierarchical model rendering mode의 구현**

Hierarchical model rendering mode는 미리 준비한 몇 개의 obj 파일들로 구성된 도형들이 각각의 부모에 대해 상대적으로 움직일 수 있게 하는 mode로, 준비한 obj 파일은 나무 기둥을 표현한 stomp.obj 파일, 나뭇가지를 표현한 branch.obj, 나뭇잎을 표현한 leaf.obj 파일의 총 3단계로 구성되어 있다. 또한 구현한 코드는 준비한 obj file들을 기반으로 각각의 varr들을 구성하는 load\_tree() 함수, 그리고 구성한 varr을 기반으로 도형을 render하는 tree\_render() 함수로 구성되어 있다.

load\_tree() 의 동작은 obj 파일들의 첫 단어가 ‘f’일 때 drop\_callback() 의 경우 ‘int//int’형식이고, load\_tree() 의 경우 ‘int/int/int’ 형식이라는 점만 제외하면 drop\_callback() 함수와 거의 동일하다.

tree\_render() 함수의 경우, glPushMatrix와 glPopMatrix 함수를 통해 부모 object에 대한 상대적 이동을 구현하였으며, 이때 부모에서 자식 순으로 보면 stomp – branch(2개) – leaf(2개) 순으로 구성되어 있고, stomp와 branch의 색은 갈색, leaf의 색은 연두색이다. stomp는 (-0.5, 0, -0.5)에서 (0.5, 0, 0.5) 사이를 직선으로 왔다 갔다 하는 움직임을 보이고, branch 하나는 stomp를 기준으로 (1, 0, 1) 방향으로 52.5~62.5도만큼 회전을, 나머지 하나는 stomp를 기준으로 (1, 0, 1) 방향으로 -62.5~-52.5도만큼 회전을 한다. 마지막으로 leaf는 각각 서로 다른 branch에 매달려, 해당 branch를 기준으로 (1, 0, 1) 방향으로 -30~30도만큼 회전을 한다.

1. **Lighting등 나머지 부분에 대한 구현**

우선 light source의 경우, (3, 4, 5) 에 point light 하나, (-3, 4, -5) 에도 point light 하나를 두었다. 이때 light color는 둘 다 (1., 1., 1.) 로 두었고, ambient light color는 (.1, .1, .1) 로 두었다. Material color의 경우 위에 언급했듯 stomp와 branch의 object color는 갈색, 정확히는 (.5859375, .29296875, 0., 1.) 로 두었고, specular object color는 (1., 1., 1., 1.) 로 두었으며, leaf의 object color는 연두색, 정확히는 (.50390625, .75390625, .27734375, 1.), specular object color는 (1., 1., 1., 1.) 로 두었다. 두 경우 모두 shininess는 10으로 설정했다.

Wireframe mode와 solid mode간의 전환에 대한 구현은 현재 mode가 무엇인지 저장해 두고, solid mode면 render 함수 시작부분에 glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FULL) 을, wireframe mode면 glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE) 을 호출하는 방식으로 구현되었다.

1. **구현 검증 영상 링크**

<https://www.youtube.com/watch?v=5KE1LuZG9Vc>