Computer Graphics 과제 2 보고서

컴퓨터 공학부

2004-11881 고우종

# 개발 환경

1. Microsoft Windows 7 Professional K (64bit)
2. Microsoft Visual Studio 2010 Professional
3. Mac OS X 10.6.7

# 실행 방법

Hw.sln 파일을 열어서 hw2 프로젝트 선택 후 Release나 Debug 모드에서 Run (F5)을 하시면 컴파일 및 실행이 됩니다.

그런데 혹시 **hw1** 프로젝트가 선택되어 컴파일 된다면, 왼쪽 사이드 바에서 hw2 프로젝트에서 오른쪽 버튼을 클릭하여 “Set as StartUp project”를 클릭 후 다시 컴파일 및 실행해주시기 바랍니다. 설정하여 제출하였는데 이상하게 가끔씩 풀리는 경우가 있는 것 같습니다.

# Virtual trackball을 이용한 3D Viewer 만들기

3D 공간 상의 오브젝트를 회전, 이동, 확대/축소시키면서 볼 수 있는 3D Viewer를 만들었습니다. 3D 공간 상에서 회전을 구현하기 위하여 Virtual trackball을 구현하였습니다.

Virtual trackball은 강의 슬라이드에 나온 것처럼 화면 너머에 화면에 꽉 차는 크기의 구가 있다고 가정하고 마우스 커서의 위치를 구 위의 점에 매핑하였습니다. 그리고 마우스의 움직임에 따라 처음 점과 끝 점을 구의 중심으로부터의 벡터로 보고 이 두 벡터 사이의 회전을 계산하여 회전 대상 오브젝트에 동일하게 적용하였습니다. 이를 통해 3D 공간 상의 오브젝트는 구가 마우스 커서에 붙어서 돌아가는 것에 맞춰 동일하게 회전을 하게 됩니다. 그리고 구가 없는 화면의 네 모서리를 클릭하면 그 점에서 가장 가까운 구 표면 위의 점으로 매핑되도록 했습니다. 따라서 그 부분을 클릭하면 구의 가장 바깥쪽 라인, 즉 z = 0인 원 위로 매핑됩니다.

Dolly in/out, 즉 forward, backward 이동은 ‘d’ (dolly in), ‘D’ (dolly out) 키를 입력하시면 됩니다. 이 키를 입력하면 world 좌표계의 원점으로 dolly in/out 합니다. 그리고 screen plane 상에서 평면 이동을 하기 위해서는 위아래, 좌우 이동 각각 ‘[‘ (위). ‘/’ (아래). ‘;’ (왼쪽), ‘’’ (오른쪽) 키를 입력하시면 됩니다. Enter 키 왼쪽에 있는 기호들을 십자 모양으로 보고 각각 화살표의 좌우와 위아래 동작을 하게 배치하였습니다.

# 추가 구현

과제 문서에 제시된 추가 스펙을 모두 구현하였습니다. Show all, Zoom in/out 그리고 Seek 기능을 구현하였습니다.

## Show all

키보드의 ‘s’키를 누르면 모든 오브젝트들이 한 화면에 보이는 위치와 거리로 카메라가 이동합니다. 이 기능은 매우 구현하기 어려웠는데, 그 이유는 앞으로 화면에 배치될 오브젝트들에 대해서도 동적으로 가장 잘 보이는 위치와 방향을 계산해내야 했기 때문입니다.

이 기능을 위해서 모든 오브젝트들의 위치와 크기 정보를 수집해야 했습니다. 자세히 설명하면 vertex들을 이어 붙여 만들어진 오브젝트의 경우에는 각 vertex의 좌표들을 모았고 구, 큐브, 실린더 같은 primitive들은 그 primitive를 포함하는 크기의 bounding box를 씌운 후 8개의 모서리 좌표를 구했습니다.

이 좌표들을 통해서 지금 공간 안의 모든 오브젝트들이 차지하는 공간 정보를 얻을 수 있었습니다. 이를 이용하여 모든 좌표들을 합하고 총 개수로 나누어서 모든 vertex들의 중점을 구했습니다. 이 중점을 중심으로 가장 멀리 있는 vertex의 좌표까지의 거리를 구해, 이 거리를 반지름으로 하는 가상의 구를 그린 후 이 구가 화면에 모두 들어오도록 하면 어느 오브젝트도 잘리지 않고 한 화면에 들어오는 카메라 거리를 구할 수 있습니다.

하지만 이렇게만 구현할 경우 오브젝트가 서로 가려서 많이 안 보이는 경우가 발생할 수 있습니다. 따라서 조금 더 보기 좋게 하기 위해서 오브젝트들이 퍼져있는 평면의 normal vector를 구해 이 위치에 카메라를 배치하기로 했습니다. 물론 정확하게 계산하는 알고리즘은 아니지만 그래도 그냥 거리만 지정하는 것보다는 더 효과적으로 오브젝트들을 보여주었습니다.

이 알고리즘은 간단하게 설명하면 위에서 계산한 중점으로부터 각 vertex의 좌표까지의 거리를 구해서 가장 긴 것과 그 다음으로 긴 것을 구해 이 두 벡터를 cross product한 벡터의 방향을 카메라가 보는 방향으로 지정하는 방식입니다. 따라서 물체가 균일하게 분포하지 않고 어느 정도 평면 형태로 펼쳐져 있다면 그 평면을 옆에서 평행하게 보는 게 아니라 정면으로, 즉 수직으로 바라보게 되어 더 많은 물체가 표시될 가능성이 높아집니다.

이 기능을 구현하였는데 결과를 보니 Show all을 하기 전에 virtual trackball로 회전을 좀 시킨 다음에 Show all을 하면 원래 구도가 위아래 뒤집혀 보이는 경우가 종종 발생했습니다. 이는 이미 회전이 적용된 상태에서 Show all은 카메라 위치를 한번의 회전으로 위에서 말한 normal vector 방향으로 옮기기 때문에 카메라의 up vector가 바뀌는 경우입니다.

이를 해결하기 위해서는 단순히 Show all 함수에 회전 매트릭스를 identity matrix로 초기화하는 코드를 한 줄 넣으면 되는데, 그렇게 해보니 Show all시 매번 같은 위치에 같은 구도가 보여지도록 되어서 마치 미리 정해둔 위치로 카메라가 이동하는 느낌이었습니다. 그래서 이 코드는 주석 처리하였고 대신 이 점 덕분에 회전을 이리저리 막 해보다가 Show all을 하면 그때그때마다 다른 구도를 보여주는 위치로 카메라가 이동하게 되었습니다. 버그가 아니라 Show all이 미리 작성자가 지정해준 좋은 위치와 방향으로 카메라를 옮기는 게 아니라 계산을 통해 동적으로 그때그때마다 다른 위치와 방향으로 이동한다는 것을 보여주기 위해서 이렇게 했습니다.

## Zoom in/out

Zoom in/out은 키보드 ‘z’ (zoom in), ‘Z’ (zoom out) 키를 입력하시면 사용할 수 있습니다. Zoom in/out은 dolly in/out과 달리 gluPerspective()의 Fov (Field of view)가 바뀌어서 확대/축소되는 효과를 냅니다. 따라서 다가갈수록 원근법이 더욱 뚜렷해지는 Dolly in/out과 달리 확대 시에도 원근법이 강조되지 않고 똑같은 모습으로 화면에 크기만 커지게 됩니다.

## Seek

Seek 기능은 화면 상에서 마우스 오른쪽 버튼을 누르면 마우스 커서가 위치한 공간 상의 표면 위의 점의 좌표를 기준으로 회전의 중심점이 바뀌게 됩니다. 즉 구 위에서 마우스 오른쪽 버튼을 누르면 구 안의 좌표가 아니라 구 표면 위의 점이 회전의 중심점이 됩니다. 주의하실 점은 배경인 검은색에는 오른쪽 클릭하시면 얻어오는 좌표가 굉장히 먼 곳이 되므로 회전이 예측할 수 없게 됩니다. 따라서 꼭 눈에 보이는 오브젝트 위에만 찍으시는 것이 좋습니다.

Seek 시에 회전 중심점뿐만 아니라 시선의 중심도 이 점으로 이동해야 되나 고민했는데 딱히 스펙에 정해진 이야기가 없어서 일단 회전 중심점만 이 점으로 바꾸게 하였습니다. 시선 중심점도 이 점으로 옮기는 코드는 주석 처리하였는데, 그 이유는 시선의 중심은 screen plane 상에서의 평면 이동 등으로도 변경할 수 있기 때문에 이렇게 처리하였습니다.

## 각 관절 움직여보기

이 기능을 사용해보기 위해서는 일단 루프를 돌고 있는 애니메이션을 정지시켜야 합니다. Void idle(void) 함수의 3번째 줄에 주석 처리된 “//return;”의 주석을 풀고 다시 실행시키시면 됩니다.

1. ‘1’, ’!’ - 로봇의 팔꿈치를 구부렸다 폈다 하는 동작
2. ‘2’, ‘@’ - 로봇이 팔을 어깨를 기준으로 비틀어 회전하는 동작 (사람은 불가능)
3. ‘3’, ‘#’ - 로봇이 팔을 어깨를 기준으로 위아래 올렸다 내렸다 하는 동작
4. ‘4’, ‘$’ - 로봇이 손을 쥐었다 폈다 하는 동작

작동이 안 될 경우 이미 정해진 한계 각도에 도달한 경우가 대부분이므로 반대키 (대문자)를 입력해보시기 바랍니다. 로봇 팔 역시 공유하는 변수가 몇 개 있으므로 같이 움직일 수 있습니다.