

컴퓨터 그래픽스

소속: 수학과

이 름: 문 하 영

담당교수: 최 민 규

[illegible]

실습 자료를 바탕으로 구현하였습니다.

‘1’key : Draw 36x18 data points using ‘GL_POINTS’

(0.5, 0, 0)에서 시작하여 z축을 중심으로 20도씩 시계 반대방향으로 돌려 점을 찍었습니다. 그 다음 x축으로 1만큼, y축으로 1만큼 이동시켰습니다. 18개의 점을 마찬가지로 y축을 중심으로 10도씩 시계 반대방향으로 돌려서 36x18개의 점을 찍었습니다. 36x18개의 점을 찍으면 배열 p[36][18]에 저장하였습니다.

강의에서 배운 Rotation과 Translate Matrix를 이용하여 어렵지 않게 구현할 수 있었습니다.

‘2’key : Draw the wireframe only

1번 키에서 생성했던 p[36][18]을 glBegin(GL_QUADS)을 통해 점을 시계방향으로 찍어가며, 사각형 wireframe을 그렸습니다. 사각형으로 그려 wireframe을 그릴 수 없는 경우에는 GL_LINES로 원으로 나타냈습니다. 점이 한 개 찍힐 경우에는 GL_POINTS로 점 한 개만 찍어 처리하였습니다.

1번 키와 다르게 a,s키와 j,k키를 이용할 때, 서로 인덱스 값이 달라서 계산하여 수정하는 부분이 오래 걸렸지만, 코드를 좀더 깊이있게 공부할 수 있는 시간도 되었습니다.

‘3’key : Draw the quads only

2번 키와 같은 방법으로 구현하고, glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);과 glPolygonOffset 정도만 수정하였습니다.

아직 명암이 없어 색만 칠했을 경우에는 torus의 모양이 완벽하게 보이지 않았습니다.

‘4’key : Draw the quads and the wireframe

먼저 색칠하고, wireframe을 그려 완성하였습니다.

3번 키와 달리, wireframe이 그려져서 3번 키처럼 어설픈 torus 모양은 아니었습니다.

‘5’key : Draw the normal vectors of the polygons

사각형을 그린 4개의 점에서, 인접해 있는 두 점에서 각각 대각선 방향의 점으로 가는 벡터 2개를 만들었습니다. 그 다음, 두 벡터의 외적을 구하여 normal vector를 구해 사각형 polygon의 가운데에서 나가도록 하였습니다.

torus 사이즈를 크게 잡지 않아서, normal vector 사이즈를 조절하는 부분이 조금 까다로웠지만, 강의에서 배운 내용으로 충분히 구현할 수 있었습니다.

‘6’key : Two-sided constant shading with the wireframe

설명해주신 대로 5에서 구한 normal vector와, 사각형 polygon 가운데에서 eye로 향하는 벡터를 내적시켜 값이 음수이면 빨간색으로, 양수이면 파란색으로 채우도록 하였습니다.

4번 키와 5번 키의 구현을 이용하여 조건만 변경해주었기 때문에 기능을 구현하는 데 어렵지 않았습니다.

‘a’key & ‘s’key : Around the y-axis

1번 키에서 생성했던 p[36][18]을 이용하여 키를 누른 횟수를 받아 인덱스를 조절하여 그리도록 하였습니다.

처음 2번 키를 눌렀을 때, 사각형이 그려진 wireframe이 아니라 1개의 원이 그려지도록 하기 위해서 인덱스 값을 1번과 달리 설정하여야 했습니다. 그래서 y축 대해 인덱스 값을 계산해주는 함수를 만들어 구현하였습니다.

‘j’key & ‘k’key : Around the z-axis

마찬가지로 키를 누른 횟수를 받아 인덱스를 조절하여 그리도록 하였습니다.

a,s 키와 반대로 처음 키를 눌렀을 때 제대로 동작하지만, 끝까지 sweep를 했을 때 인덱스 오류가 발생하였습니다 .그래서 위처럼 z축에 대해 인덱스 값을 계산하는 함수를 다시 만들어 구현하였습니다.