
컴퓨터 그래픽스 기말고사 대체 레포트

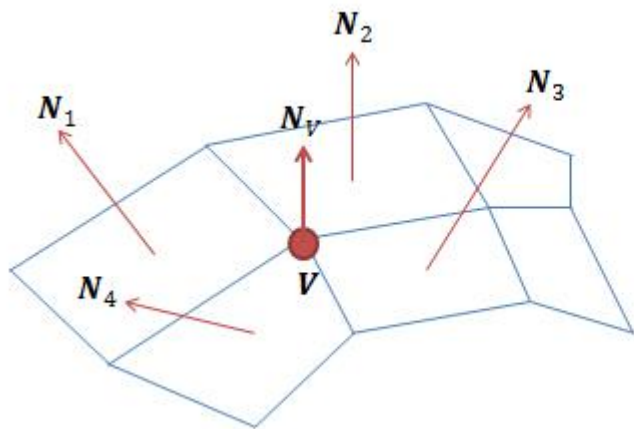
20204656 김지혜

필수구현



1 조명 구현

폴리곤의 외적 벡터 구하기



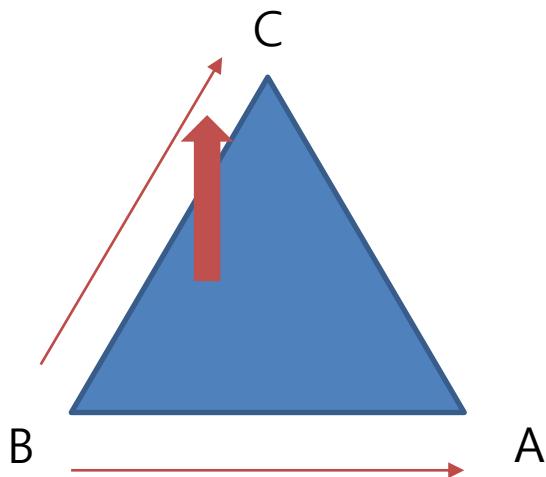
$$N_v = \frac{\sum_{k=1}^n N_k}{\left| \sum_{k=1}^n N_k \right|}$$

조명을 구현하기 위해선,
폴리곤들의 법선벡터가 필요함.

1 조명 구현

폴리곤의 외적 벡터 구하기

```
//외적값을 구하기 위한
Point cnormal(Point a, Point b, Point c) {
    Point p, q, r;
    double val;
    p.x = a.x - b.x; p.y = a.y - b.y; p.z = a.z - b.z; //p벡터 (BA벡터)
    q.x = c.x - b.x; q.y = c.y - b.y; q.z = c.z - b.z; //q벡터 (BC벡터)
    r.x = p.y * q.z - p.z * q.y; //법선벡터 x값
    r.y = p.z * q.x - p.x * q.z; //법선벡터 y값
    r.z = p.x * q.y - p.y * q.x; //법선벡터 z값
    val = sqrt(r.x * r.x + r.y * r.y + r.z * r.z);
    r.x = r.x / val; r.y = r.y / val; r.z = r.z / val; //법선벡터 정규화
    return r;
}
```



법선벡터는 평면 위 두 변의
각 꼭짓점 좌표를 뺌으로서 두 방향
벡터를 만들고, 이 둘을 외적시킨다.

폴리곤의 삼각형으로부터 받아온
꼭짓점이 차례대로 A,B,C라 할 때
BAXBC가 법선벡터가 되도록 설계함.

1 조명 구현

폴리곤의 외적 벡터 구하기

```
//face기록
fprintf(fout, "FACE = %d\n", (2 * n + s) - 4);
for (int i = 0; i < n-1; i++) {
    for (int k = 0; k < s - 1; k++)
    {
        fprintf(fout, "%d %d %d\n", i * s + k + 1, i * s + k, (i + 1) * s + k + 1);
        fprintf(fout, "%d %d %d\n", (i + 1) * s + k, (i + 1) * s + k + 1, i * s + k);
    }
}
for (int k = 0; k < s - 1; k++)
{
    fprintf(fout, "%d %d %d\n", m - s + k + 1, m - s + k, k+1);
    fprintf(fout, "%d %d %d\n", k, k + 1, m - s + k);
}

for (int k = 0; k < n - 2; k++)
{
    fprintf(fout, "%d %d %d\n", 0, (k + 2) * s, (k + 1) * s);
    fprintf(fout, "%d %d %d\n", s - 1, (k + 3) * s - 1, (k + 2) * s - 1);
}
fclose(fout);
```

한 면을 삼각형 2개로 쪼개되, 두 삼각형의 평면벡터는 같아야함.

윗 삼각형 아랫삼각형
모두 같은 방향으로 face 입력

1

조명 구현

폴리곤의 외적 벡터 구하기

```
//조명설정
void InitLight() {
    GLfloat mat_diffuse[] = { 0.5, 0.4, 0.3, 1.0 };
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat mat_ambient[] = { 0.5, 0.4, 0.3, 1.0 };
    GLfloat mat_shininess[] = { 15.0 };
    GLfloat light_specular[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat light_diffuse[] = { 0.8, 0.8, 0.8, 1.0 };
    GLfloat light_ambient[] = { 0.3, 0.3, 0.3, 1.0 };
    GLfloat light_position[] = { 0, 6, 3.0, 0.0 };
    glShadeModel(GL_SMOOTH);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
}
```

조명기능을 위해
InitLight함수를 호출한다.
조명의 설정값은 다음과 같다.

2 키보드 콜백

키보드로부터 다양한 함수 호출 가능

```
//ShowModel에서 마우스와 키보드 함수
void keyboard(unsigned char key, int x, int y)
{
    printf("key %d\n", key);
    switch (key)
    {
        case 'w':
            status = WIRE; glutPostRedisplay(); break;
        case 's':
            status = SHADE; glShadeModel(GL_SMOOTH); glutPostRedisplay(); break;
        case 'f':
            status = SHADE; glShadeModel(GL_FLAT); glutPostRedisplay(); break;
        case '4': //2를 누르면 FRONT FACE
            glCullFace(GL_FRONT); glutPostRedisplay(); break;
        case '6': //4를 누르면 BACKFACE
            glCullFace(GL_BACK); glutPostRedisplay(); break;
        //목표점 변환
        case '7': //7를 누르면 vx+=0.01
            vx += 1; glutPostRedisplay(); break;
        case '9': //9를 누르면 vx-=0.01
            vx -= 1; glutPostRedisplay(); break;
        case '1': //1을 누르면 vy+=0.01
            vy += 1; glutPostRedisplay(); break;
        case '3': //3을 누르면 vy-=0.01
            vy -= 1; glutPostRedisplay(); break;
        case '2': //2를 누르면 BACKFACE
            vz += 1; glutPostRedisplay(); break;
        case '8': //8을 누르면 BACKFACE
            vz -= 1; glutPostRedisplay(); break;
        //카메라 위치 변환
        case 'q': //q를 누르면 cx+=0.01
            cx += 1; glutPostRedisplay(); break;
        case 'e': //e를 누르면 cx-=0.01
            cx -= 1; glutPostRedisplay(); break;
        case 'a': //a를 누르면 cy+=0.01
            cy += 1; glutPostRedisplay(); break;
        case 'd': //d를 누르면 cy-=0.01
            cy -= 1; glutPostRedisplay(); break;
        case 'z': //z를 누르면 BACKFACE
            cz += 1; glutPostRedisplay(); break;
        case 'c': //c를 누르면 BACKFACE
            cz -= 1; glutPostRedisplay(); break;
    }
}
```

키보드 콜백 함수인 keyboard를 정의함.

W를 누르면 와이어모드,
S를 누르면 셰이드 모드로 바뀐다.
(f를 누르면 SMOOTH로 셰이딩이 되어야
하나,SMOOTH는 구현하지 못했다.)

또한, BackFace Culling도 가능함.

그 밖에 카메라 위치, 시점위치
(바라보는 점) 또한 키보드 콜백으로 설정
하였다.

3 마우스 콜백

마우스로부터 다양한 함수 콜백

```
void mouse(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        moving = 1;
        mousebegin = x;
    }
    if (button == GLUT_LEFT_BUTTON && state == GLUT_UP) {
        moving = 0;
    }
    if (button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN) {
        scaling = 1;
        mousebegin = x;
    }
    if (button == GLUT_RIGHT_BUTTON && state == GLUT_UP) {
        scaling = 0;
    }
}

void motion(int x, int y) {
    if (scaling) {
        scalefactor = scalefactor * (1.0 + (mousebegin - x) * 0.0001);
        glutPostRedisplay();
    }
    if (moving) {
        angle_x = angle_x + (x - mousebegin);
        angle_y = angle_y + (y - mousebegin); mousebegin = y;
        glutPostRedisplay();
    }
}
```

마찬가지로 마우스 콜백함수
mouse를 정의함.
왼쪽 클릭으로는 스케일 조절을,
오른쪽 클릭으로는 물체의 회전각
을 조절함.

(*motion 함수는
마우스 클릭이 된 상태에 관련한
함수임)

감사합니다.
