컴퓨터그래픽스 과제 - 광원 애니메이션

컴퓨터과학과 2020010863 조정미


```
#include <GL/glut.h>
#include <GL/GLU.h>

static int SpinAngle = 0;
GLfloat Ka = 0.4, Kd = 0.9, Ks = 1.0, Sn = 25.0;

void InitLight() {
	// 조명 특성
	GLfloat light0_ambient[] = { 0.5, 0.4, 0.3, 1.0 };
	GLfloat light0_diffuse[] = { 0.8, 0.7, 0.6, 1.0 };
	GLfloat light0_specular[] = { 1.0, 1.0, 1.0, 1.0 };

	// 물체 특성
	GLfloat material_ambient[] = { Ka, Ka, Ka, 1.0 };
	GLfloat material_diffuse[] = { Kd, Kd, Kd, 1.0 };
	GLfloat material_specular[] = { Ks, Ks, Ks, 1.0 };
	GLfloat material_shiniess[] = { Sn };

	glShadeModel(GL_SMOOTH); // 구로셰이딩
	glEnable(GL_DEPTH_TEST);   // 깊이 버퍼 활성화
	glEnable(GL_LIGHTING);       // 조명 기능 활성화

	glEnable(GL_LIGHT0);
```

```
        glLightfv(GL_LIGHT0, GL_AMBIENT, light0_ambient);      // Ia: 주변광
        glLightfv(GL_LIGHT0, GL_DIFFUSE, light0_diffuse);      // Id: 확산광
        glLightfv(GL_LIGHT0, GL_SPECULAR, light0_specular);   // Is: 경면광
}
void MyDisplay() {
        GLfloat LightPosiion[] = { 0.0, 0.0, 1.5, 1.0 };
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

        // 1. 이전의 변환 행렬을 저장
        glPushMatrix();
                // 2. 좌표계가 z축 방향으로 -5만큼 이동
                glTranslatef(0.0, 0.0, -5.0);
                // 3. 이동한 좌표계 저장
                glPushMatrix();
                        // 4. 현재 좌표계의 x축을 중심으로 좌표계를 회전
                        // - SpinAngle값 변화에 영향
                        glRotatef(SpinAngle, 1.0, 0.0, 0.0);
                        // 5. 광원을 z축을 따라 1.5만큼 떨어진 곳으로 위치
                        glLightfv(GL_LIGHT0, GL_POSITION, LightPosiion);
                        // 6. 좌표계를 z축으로 1.5만큼 이동
                        glTranslatef(0.0, 0.0, 1.5);
                        // 7. 조명 비활성화 - 광원 자체를 직접 그리기 위해
                        glDisable(GL_LIGHTING);
                        glColor3f(0.9, 0.9, 0.9);
                        // 8. 백색 와이어 프레임으로 광원의 모습을 그림
                        glutWireSphere(0.06, 400, 400);
                        // 9. 조명 활성화
                        glEnable(GL_LIGHTING);
                glPopMatrix();
                // 10. 실제 조명을 받을 원구를 그림
                glutSolidSphere(1.0, 400, 400);
        glPopMatrix();
        glFlush();
}
void MyReshape(int w, int h) {
        glViewport(0, 0, (GLsizei)w, (GLsizei)h);
        glMatrixMode(GL_PROJECTION);   //투상행렬
        glLoadIdentity();                       //항등행렬 로드
        gluPerspective(40.0, (GLfloat)w / (GLfloat)h, 1.0, 20.0);
        glMatrixMode(GL_MODELVIEW);
```

```
	glLoadIdentity();
}
void MyMouse(int button, int state, int x, int y) {
	switch (button) {
	case GLUT_LEFT_BUTTON:
		if (state == GLUT_DOWN) {
			SpinAngle = (SpinAngle + 15) % 360;
			glutPostRedisplay();
		}
		break;
	default:
		break;
	}
}
int main() {
	glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
	glutInitWindowSize(500, 500);
	glutInitWindowPosition(100, 100);
	glutCreateWindow("OpenGL Sample Drawing");
	InitLight();
	glutDisplayFunc(MyDisplay);
	glutReshapeFunc(MyReshape);
	glutMouseFunc(MyMouse);
	glutMainLoop();
	return 0;
}
```