

컴퓨터 그래픽스 과제

컴퓨터과학과 3학년 2020010863 조정미

조명 2개를 사용하여 구(sphere)를 그리시오.
조명의 색과 위치를 달리하여 2개의 출력 결과를 제시한다.

1) 광원이 하나인 경우



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

// 광원의 특성
void MyLightInit() {
    // 0번 광원
    GLfloat light0_ambient[] = { 1.0, 0.0, 0.0, 1.0 }; // Ia: 주변광 특성
    GLfloat light0_diffuse[] = { 1.0, 0.0, 0.0, 1.0 }; // Id: 확산광 특성
    GLfloat light0_specular[] = { 1.0, 1.0, 1.0, 1.0 }; // Is: 경면광 특성

    glLightfv(GL_LIGHT0, GL_AMBIENT, light0_ambient); // Ia 설정
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light0_diffuse); // Id 설정
    glLightfv(GL_LIGHT0, GL_SPECULAR, light0_specular); // Is 설정

    glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 1.0); // 거리 감쇠(상수)
    glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, 0.2); // 거리 감쇠(1차항)
```

```

    glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, 0.0); // 거리 감쇠(2차항)
}

// 물체의 특성
void MyMaterialInit() {
    GLfloat material_ambient[] = { 0.1, 0.1, 0.1, 1.0 }; // Ka: 주변광 계수
    GLfloat material_diffuse[] = { 0.8, 0.8, 0.8, 1.0 }; // Kd: 확산광 계수
    GLfloat material_specular[] = { 1.0, 1.0, 1.0, 1.0 }; // Ks: 경면광 계수
    GLfloat material_shiniess[] = { 25.0 }; // n:광택 계수

    glMaterialfv(GL_FRONT, GL_AMBIENT, material_ambient); // Ka 설정
    glMaterialfv(GL_FRONT, GL_DIFFUSE, material_diffuse); // Kd 설정
    glMaterialfv(GL_FRONT, GL_SPECULAR, material_specular); // Ks 설정
    glMaterialfv(GL_FRONT, GL_SHININESS, material_shiniess); // n 설정
}

void MyReshape(int w, int h) {
    if (w < h)
        glViewport(0, 0, (GLsizei)w, (GLsizei)w);
    else
        glViewport(0, 0, (GLsizei)h, (GLsizei)h);
    glMatrixMode(GL_PROJECTION); // 투상행렬
    glLoadIdentity(); // 항등행렬 로드
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0); // 평행투상
}

void MyDisplay() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_DEPTH_TEST);

    glPushMatrix();

    // 조명기능 활성화
    glEnable(GL_LIGHTING);

    // 0번 광원 배치
    glEnable(GL_LIGHT0);
    GLfloat light0_position[] = { -2.0, -2.0, 2.0, 1.0 };
    glLightfv(GL_LIGHT0, GL_POSITION, light0_position);

```

```

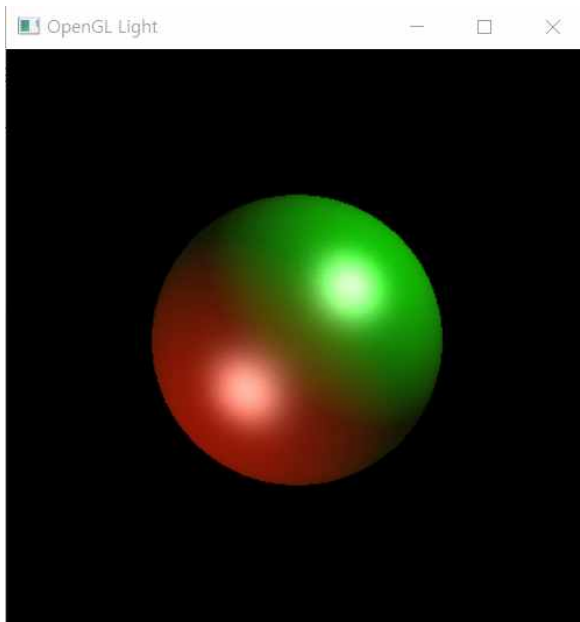
        glutSolidSphere(0.5, 50, 50);
        glPopMatrix();

        glFlush();
    }

    int main() {
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA | GLUT_DEPTH);
        glutInitWindowSize(400, 400);
        glutInitWindowPosition(500, 400);
        glutCreateWindow("OpenGL Light");
        MyLightInit();
        MyMaterialInit();
        glutDisplayFunc(MyDisplay);
        glutReshapeFunc(MyReshape);
        glutMainLoop();
        return 0;
    }

```

2) 광원이 2개인 경우



```

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

// 광원의 특성

```

```
void MyLightInit() {
```

```
    // 0번 광원
```

```
    GLfloat light0_ambient[] = { 1.0, 0.0, 0.0, 1.0 }; // Ia: 주변광 특성
```

```
    GLfloat light0_diffuse[] = { 1.0, 0.0, 0.0, 1.0 }; // Id: 확산광 특성
```

```
    GLfloat light0_specular[] = { 1.0, 1.0, 1.0, 1.0 }; // Is: 경면광 특성
```

```
    glLightfv(GL_LIGHT0, GL_AMBIENT, light0_ambient); // Ia 설정
```

```
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light0_diffuse); // Id 설정
```

```
    glLightfv(GL_LIGHT0, GL_SPECULAR, light0_specular); // Is 설정
```

```
    glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 1.0); // 거리 감쇠(상수)
```

```
    glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, 0.2); // 거리 감쇠(1차항)
```

```
    glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, 0.0); // 거리 감쇠(2차항)
```

```
    // 1번 광원
```

```
    GLfloat light1_ambient[] = { 0.0, 1.0, 0.0, 1.0 }; // Ia: 주변광 특성
```

```
    GLfloat light1_diffuse[] = { 0.0, 1.0, 0.0, 1.0 }; // Id: 확산광 특성
```

```
    GLfloat light1_specular[] = { 1.0, 1.0, 1.0, 1.0 }; // Is: 경면광 특성
```

```
    glLightfv(GL_LIGHT1, GL_AMBIENT, light1_ambient); // Ia 설정
```

```
    glLightfv(GL_LIGHT1, GL_DIFFUSE, light1_diffuse); // Id 설정
```

```
    glLightfv(GL_LIGHT1, GL_SPECULAR, light1_specular); // Is 설정
```

```
    glLightf(GL_LIGHT1, GL_CONSTANT_ATTENUATION, 1.0); // 거리 감쇠(상수)
```

```
    glLightf(GL_LIGHT1, GL_LINEAR_ATTENUATION, 0.2); // 거리 감쇠(1차항)
```

```
    glLightf(GL_LIGHT1, GL_QUADRATIC_ATTENUATION, 0.0); // 거리 감쇠(2차항)
```

```
}
```

```
// 물체의 특성
```

```
void MyMaterialInit() {
```

```
    GLfloat material_ambient[] = { 0.1, 0.1, 0.1, 1.0 }; // Ka: 주변광 계수
```

```
    GLfloat material_diffuse[] = { 0.8, 0.8, 0.8, 1.0 }; // Kd: 확산광 계수
```

```
    GLfloat material_specular[] = { 1.0, 1.0, 1.0, 1.0 }; // Ks: 경면광 계수
```

```
    GLfloat material_shiniess[] = { 25.0 }; // n:광택 계수
```

```
    glMaterialfv(GL_FRONT, GL_AMBIENT, material_ambient); // Ka 설정
```

```
    glMaterialfv(GL_FRONT, GL_DIFFUSE, material_diffuse); // Kd 설정
```

```
    glMaterialfv(GL_FRONT, GL_SPECULAR, material_specular); // Ks 설정
```

```
    glMaterialfv(GL_FRONT, GL_SHININESS, material_shiniess); // n 설정
```

```
}
```

```

void MyReshape(int w, int h) {
    if (w < h)
        glViewport(0, 0, (GLsizei)w, (GLsizei)w);
    else
        glViewport(0, 0, (GLsizei)h, (GLsizei)h);
    glMatrixMode(GL_PROJECTION); // 투상행렬
    glLoadIdentity(); // 항등행렬 로드
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0); // 평행투상
}

void MyDisplay() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_DEPTH_TEST);

    glPushMatrix();

    // 조명기능 활성화
    glEnable(GL_LIGHTING);

    // 0번 광원 배치
    glEnable(GL_LIGHT0);
    GLfloat light0_position[] = { -2.0, -2.0, 2.0, 1.0 };
    glLightfv(GL_LIGHT0, GL_POSITION, light0_position);

    // 1번 광원 배치
    glEnable(GL_LIGHT1);
    GLfloat light1_position[] = { 3.0, 3.0, 3.0, 3.0 }; // 광원의 위치
    glLightfv(GL_LIGHT1, GL_POSITION, light1_position);

    glutSolidSphere(0.5, 50, 50);
    glPopMatrix();

    glFlush();
}

int main() {
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA | GLUT_DEPTH);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(500, 400);

```

```
glutCreateWindow("OpenGL Light");  
MyLightInit();  
MyMaterialInit();  
glutDisplayFunc(MyDisplay);  
glutReshapeFunc(MyReshape);  
glutMainLoop();  
return 0;  
}
```