

# 조명 결합

어떤 예제에도 조명 처리를 추가  
할 수 있다.

# 시작

- 실습자료 7-4 Model-View-Projection
- 제공된 주석을 해제하면 다음이 가능
  - 기본 카메라 동작 (키보드)
  - Idle callback을 이용한 애니메이션
  - 계층 구조를 이용한 관절 동작
  - Wireframe 렌더링
- 할 일
  - 조명 추가
    - 재질 변경
  - Solid Model로 변경
    - 깊이 버퍼 사용

# 조명 추가

- 실습자료 8-3
  - void InitLight() 함수 부분만을 복사해서 추가
  - 조명의 색, 위치, 재질을 설정
- 문제: InitLight() 함수를 main에서 호출해야 하는데, 적절한 위치는?
  - cam.set(2,3,4, 0,0,0, 0,1,0);
  - InitLight(); // 전역 좌표 기준의 조명을 설정
  - glutDisplayFunc(MyDisplay);

# Solid Model로 변경

- `gluQuadricDrawStyle(cyl, GLU_LINE);`
  - `GLU_LINE` 을 `GLU_FILL`로 변경
- `glutWireCube`, `glutWireSphere`
  - `glutSolidCube`, `glutSolidSphere`로 변경
- 깊이 버퍼 활성화
  - `glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA | GLUT_DEPTH);` // main 함수
  - `glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );` // MyDisplay 함수
  - `glEnable(GL_DEPTH_TEST);` // main 함수에 추가

# 완성

- 배경화면을 검정색으로
  - `glClearColor(0.0, 0.0, 0.0, 1.0);` // main 함수
- 만약 물체마다 다른 색으로 그리고 싶다면
  - 물체를 그리기 전에, 재질(material)변경
- `void MyDisplay()` // 재질 설정 없으면 조명 불가
  - `float mat_diffuse1[4] = {0.5,0.4,0.3,1};` // 추가
  - `float mat_diffuse2[4] = {0.5,0.9,0.3,1};` // 추가
  - // 1. 몸통
  - `glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse1);` // 추가
  - `glPushMatrix();` // 이하 생략
  - // 2. 팔
  - `glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse2);` // 추가
  - `glPushMatrix();` // 이하 생략

# 물체마다 조명을 다르게

- void MyDisplay()
  - glDisable(GL\_LIGHTING); // 추가
  - glutSolidSphere(0.2, 20,20);
  - glEnable(GL\_LIGHTING); // 추가
- 조명 연산이 활성화되면, 물체 기본색상 glColor는 무시된다.
  - Material을 원하는 색으로 부여하고, 흰색 조명을 사용하면 대략 원하는 색상을 얻을 수 있다.
  - 그래도 꼭 순수한 glColor를 사용하고 싶다면, 그 물체를 그릴 때만 glDisable(GL\_LIGHTING);

# 결론 – 조명 추가 과정

- Wireframe으로 잘 동작하는 것을 확인
- 조명 효과 추가
  - 문제가 있으면, 조명의 좌표 또는 색상 확인
- Solid 모델로 변경
  - 문제가 있으면, Depth buffer 활성화 되어 있는지 확인
  - Clear color 색상 확인
- 각 물체마다 다른 재질/조명 부여
- 참고:
  - 물체 재질/조명 속성도 효율적인 push / pop 방법이 있으나, 수업의 범위를 넘어선다.