Computer graphics
제3장 셰이딩과 셰이딩

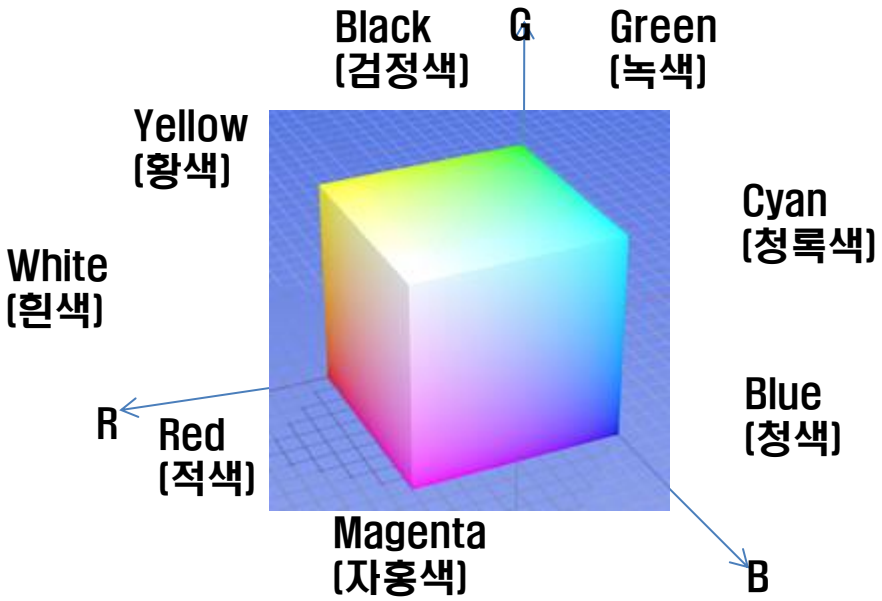
2015년 2학기

OpenGL과 셰이딩

- 차례
  - 컬러
  - 쉐이딩
  - 상태 설정
  - 은면 제거
  - 조명 모델
  - 재질 설정

컬러와 조명 효과

- openGL에서의 컬러 모델: RGB color model



쉐이딩

- 그림색 결정
  - glColor<x><t> (red, green, blue, alpha);
    - x: 인자의 개수, t: 인자의 자료형
    - 현재 드로잉 색상을 설정, 이 명령 뒤에 드로잉 되는 모든 물체는 이 색상을 사용한다.
      - glColor3f, glColor3ub…
    - glColor 함수는 이 명령 뒤에 그려지는 모든 버텍스에 적용되는 색상을 지정하는데 사용된다.
  - 각 버텍스에 다른 색상을 지정하면 색상이 부드럽게 변환된다.

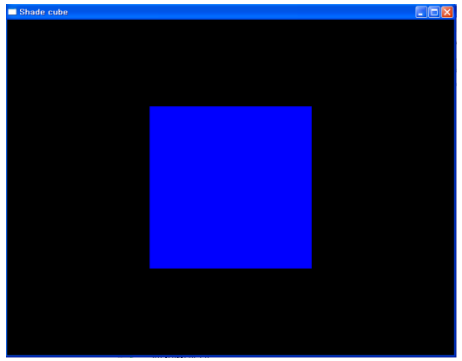
쉐이딩

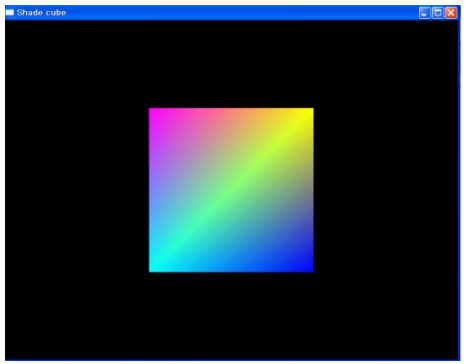
- 음영 (Shading) 넣기
  - 쉐이딩: 음영 또는 표면 렌더링으로 물체 면의 색을 부여하는 것
  - 쉐이딩 모델 설정
    - glShadeModel (GLenum mode);
      - 기본 쉐이딩, 폴리건 내부의 색은 각 정점에 설정된 색을 보간한 값으로 지정된다. Flat 이나 smooth 으로 결정한다.
      - GLenum mode:
        - » GL_FLAT: 평면 쉐이딩 (flat shading) – 마지막 정점에 대해 설정된 색으로 칠해지게 된다.
        - » GL_SMOOTH: 부드러운 쉐이딩 (smooth shading) –지정된 색들이 보간법에 의해 점차로 변해가게 칠해진다.

Shading

- 예) 부드러운 쉐이딩이 적용된 사각형 그리기

```
glShadeModel (GL_SMOOTH);           // 또는 glShadeModel (GL_FLAT)
glBegin (GL_QUADS);
                glColor3f (1.0f, 1.0f, 0.0f);       // Yellow
                glVertex3f (100.0, 100.0, 0.0);
                glColor3f (1.0f, 0.0, 1.0f);        // Magenta
                glVertex3f (-100.0, 100.0, 0.0);
                glColor3f (0.0f, 1.0f, 1.0f);       // Cyan
                glVertex3f (-100.0, -100.0, 0.0);
                glColor3f (0.0f, 0.0f, 1.0f);       // Blue
                glVertex3f (100.0, -100.0, 0.0);
glEnd ();
```





상태 설정

- OpenGL에서의 상태 및 상태 관리
  - 대부분의 상태들은 (라이팅, 텍스처링, 은면 제거, 안개 효과 등) 디폴트로 비활성화(disable)되어 있다.
  - 상태를 활성화(켜거나)하거나 비활성화(끄는)하는 명령어
    - Void **glEnable** (GLenum cap);
      - 지정한 기능을 활성화한다.
    - Void **glDisable** (GLenum cap);
      - 지정한 기능을 비활성화 한다.
  - 활성화 여부를 체크하는 명령어
    - GLboolean glIsEnabled (GLenum cap);

상태 설정

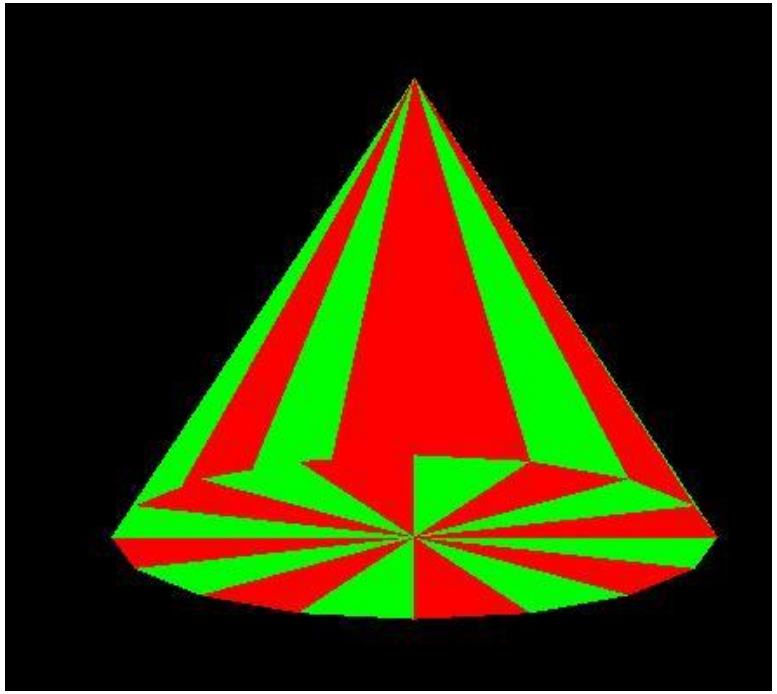
- glEnable(GLEnum cap)/glDisable(GLEnum cap)
  - OpenGL이 제공하는 다양한 기능들을 켜거나 끄거나 하는 함수
  - Cap:
    - GL_ALPHA_TEST : 알파값을 테스트 (glAlphaFunc)
    - GL_BLEND: 픽셀 블렌딩 연산을 수행 (glBlendFunc)
    - GL_CULL_FACE: 앞면 혹은 뒷면을 향하는 폴리곤을 선별 (glCullFace)
    - GL_DEPTH_TEST: 깊이를 비교
    - GL_DITHER: 컬러의 디더링 수행
    - GL_LIGHTx: 조명x를 켠다
    - GL_LIGHTING : 조명 연산을 켠다 (glLight)
    - GL_LINE_SMOOTH: 선의 안티알리아싱 효과
    - GL_STENCIL_TEST: 스텐실 테스트
    - GL_TEXTURE_CUBE_MAP: 큐브맵 텍스처링
    - GL_TEXTURE_1D: 2D 텍스처링이 생성되지 않을 때 1D 텍스처 맵핑 수행 (glTextImage1D)
    - GL_TEXTURE_2D: 2D 텍스처 맵핑 수행 (glTextImage2D)
    - …

은면 제거

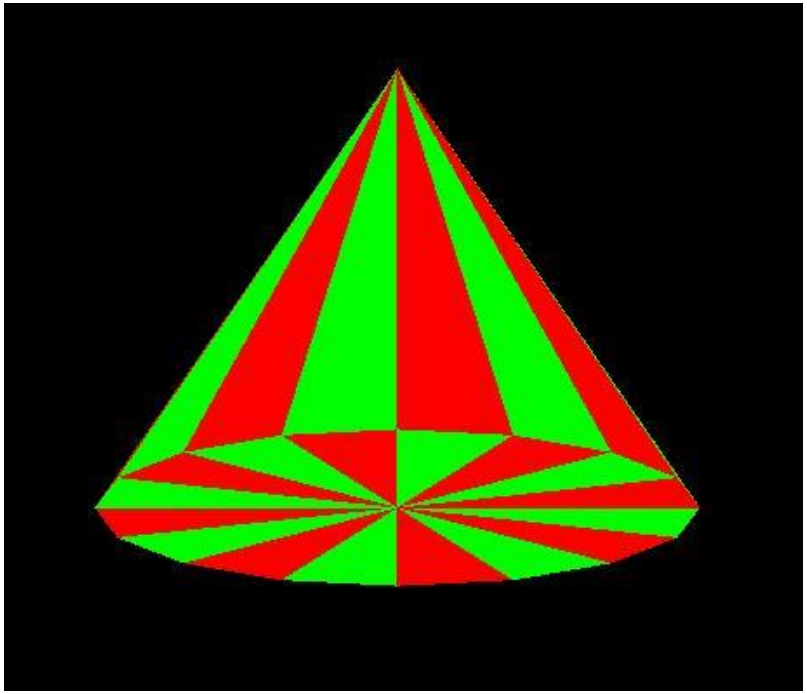
- 은면 제거
  - 3차원 장면을 2차원 평면에 투영시키면 물체들이 중첩될 수 있는데, 관측자의 시점에서 가까운 면은 보이고 깊이가 큰 면은 가려져 보이도록 은면 제거를 한다.

  - **깊이 검사 (depth test)를 사용한다.**
    - 윈도우 초기화 시 깊이 검사 모드 설정
      - glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB | ***GLUT_DEPTH***)
    - 깊이 버퍼를 클리어한다
      - glClear (GL_COLOR_BUFFER_BIT | ***GL_DEPTH_BUFFER_BIT***);
    - 깊이 검사를 설정:
      - glEnable (**GL_DEPTH_TEST**);
    - 깊이 검사를 해제:
      - glDisable (**GL_DEPTH_TEST**);

은면 제거



Depth test를 안 했을 때



Depth test를 했을 때

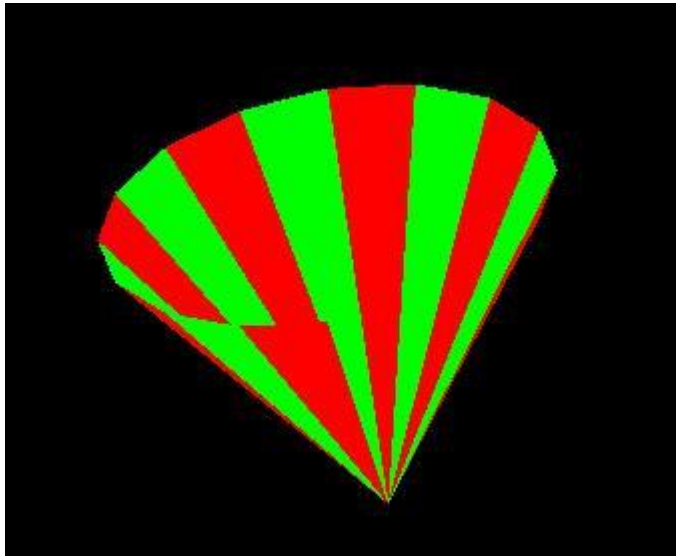
은면 제거

- 컬링 (culling)
  - 후면을 선별(backface culling)하여 **뒷면을 모두 제거**할 수 있다.
    - Winding을 이용하여 폴리곤의 앞면과 뒷면을 구분한다.
    - 시계 반대방향으로 winding되는 폴리곤이 앞면이다.

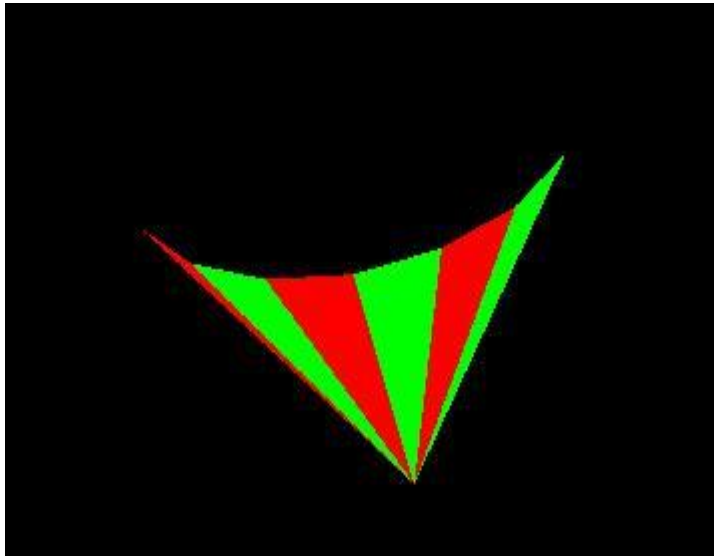
      - 컬링 설정: glEnable (**GL_CULL_FACE**);
      - 컬링 해제: glDisable (**GL_CULL_FACE**);

      - void **glFrontFace** (Glenum mode);
        - » 폴리곤의 어느 면이 앞면 또는 뒷면인지 정의한다.
        - » 장면이 닫힌 객체로 구성되어 있을 때 그 객체의 내부 연산은 불필요한데, 폴리곤의 어느 면이 앞면인지를 결정할 수 있다.
        - » Glenum mode: GL_CW – 시계방향, GL_CCW – 반시계 방향
        - » glFrontFace (GL_CW): 시계 방향을 앞면으로
        - » glFrontFace (GL_CCW): 반시계 방향을 앞면으로

은면 제거



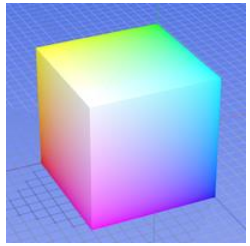
Culling을 안 했을 때



Culling을 했을 때

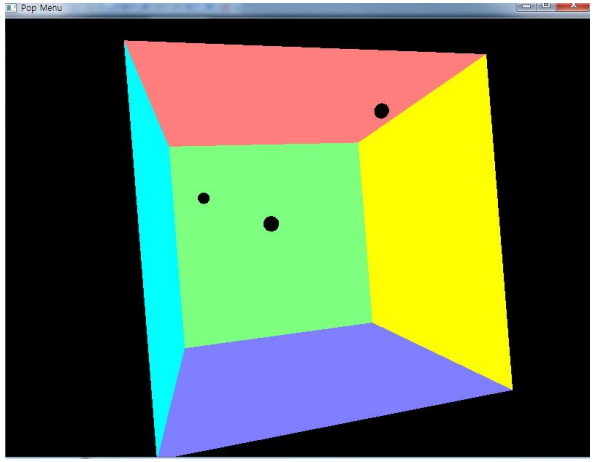
실습 19

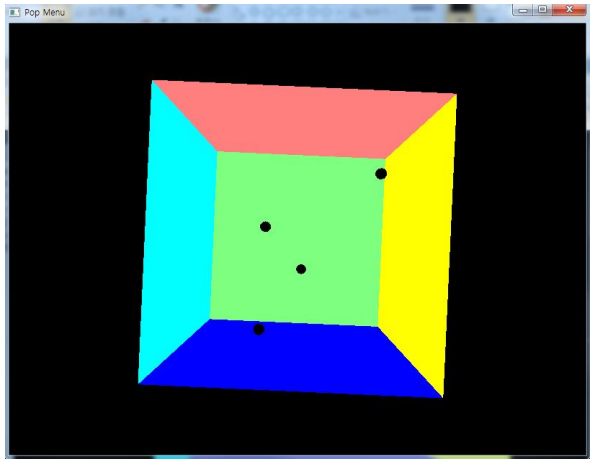
- RGB 컬러 모델 만들기
  - 육면체를 만들어서 각 꼭지점에 Red, Green, Blue, Yellow, Cyan, Magenta, White, Black의 값을 대응하여 컬러 모델을 만들기
  - 육면체는 GL_QUAD를 사용하여 그리기
  - 초기에 X축으로 30도, y축으로 45도로 회전 돼있고, y축에 대하여 회전하고 있도록 한다.
  - 팝업 메뉴를 추가하여 은면 제거와 컬링, 쉐이딩을 추가한다.
    - 은면 제거: on / off
    - 컬링: on / off
    - 쉐이딩: flat / smooth
    - 윗면: yes / no (윗면을 그린다 / 윗면을 안 그린다)
    - 앞면: yes / no (앞면을 그린다 / 앞면을 안 그린다)



실습 20

- 3차원 공 튕기기
  - 원근 투영/은면제거를 사용한다.
  - 공을 튀길 공간을 평면을 이용하여 직육면체의 형태로 만든다.
    - 각 면은 다른 색으로 표현한다.
    - 왼쪽, 오른쪽, 아래쪽, 위쪽, 뒤쪽의 면을 그린다.
    - 실습 19는 육면체의 바깥쪽이, 실습 20에서는 안쪽이 보이는 면이다.
  - 공간 내부에서 공(sphere)이 이동한다.
  - 직육면체의 면에 맞으면 맞은 면은 색을 바꾸고, 공은 방향을 바꿔서 이동을 한다.
  - 키보드 명령
    - z/Z: z축으로 양/음 방향으로 이동
    - y/Y: Y축에 시계/반시계 방향으로 회전
    - B: 볼이 새로 생겨서 튕기기 시작한다 (최대 5개)
  - 마우스 명령
    - 마우스를 왼쪽으로 옮기면 육면체가 왼쪽으로 회전
    - 마우스를 오른쪽으로 옮기면 육면체가 오른쪽으로 회전

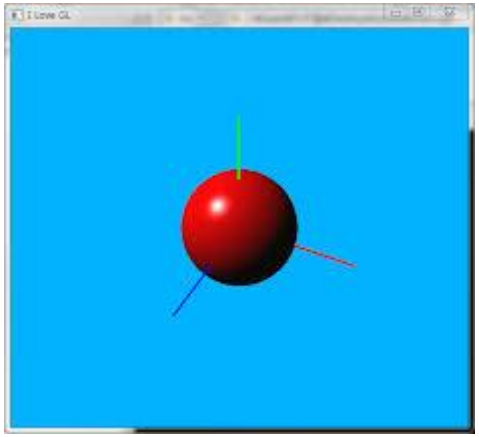


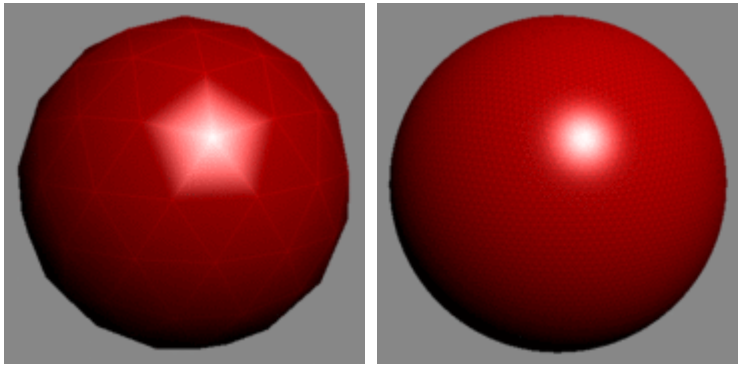


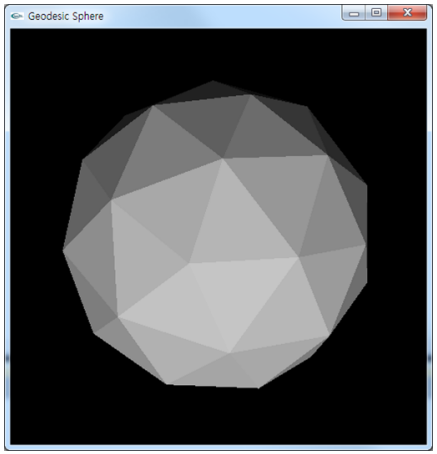
실습 21

조명 모델

- OpenGL에서 빛:
  - 빨간색, 초록색, 파란색의 성분
    - 광원의 색은 광원에서 방출하는 R, G, B 성분의 양에 따라 결정
  - 표면의 재질은 표면에 들어왔다가 다양한 방향으로 반사되는 빛의 RGB 성분의 비율로 결정
  - 즉, openGL의 조명은 **광원 (Lights), 재질(Materials), 면의 법선벡터(Normal)**에 의해서 결정된다.

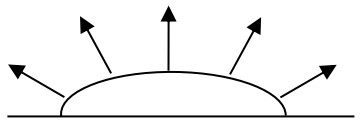


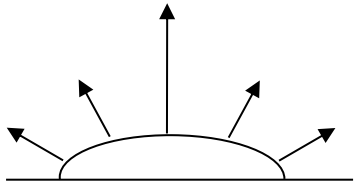


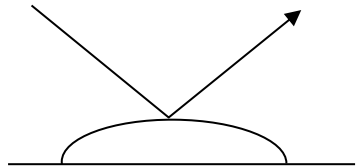


조명 모델

- OpenGL에서 지원하는 광원
  - 주변 조명 (Ambient light)
    - 빛은 모든 방향에서 고르게 비춘다.
    - 배경 조명
  - 산란 반사 조명 (Diffuse light)
    - 방향과 위치를 가짐으로 빛이 비치는 물체의 면이 밝아지는 조명
    - 일정한 방향으로 빛이 들어와서 물체의 표면에서 여러 방향으로 분산된다.
    - 빛을 받는 표면은 그렇지 않은 부분에 비해 밝게 보인다.
  - 거울반사 조명 (Specular light)
    - 특정한 방향으로 들어와서 한 방향으로 완전히 반사되는 빛
    - 하이라이트가 생긴다.
  - 위의 3종류의 빛을 합쳐서 조명 모델을 결정한다.
  - OpenGL에서 한 장면에 동시에 8개의 광원 사용 가능







조명 모델

- 광원 설정
  - 광원의 속성은 glLight ()함수에 의해 지정된다.
  - 광원은 위치(position)와 색(color)를 갖는다.
  - 광원의 강도는 색의 강도에 의해 결정된다.
  - 광원은 방향성 광원(directional light, infinite light)이거나 위치성 광원(positional light, local light) 이다.
    - 방향성 광원의 모든 빛은 같은 방향을 갖는다.
    - 위치성 광원은 공간의 특정 지점에서 오는 빛이다.
      - 광원의 위치의 4번째 값이 0이면 방향성 광원이고, 1이면 위치성 광원이다.
  - 광원의 위치는 현재 변환에 영향을 받는다.
    - 카메라 변환 후에 지정하는 것이 좋다.
    - 광원은 객체와 같이 변한 행렬에 의해 움직일 수 있다.

조명 모델

- 재질의 속성
  - 재질의 속성은 표면이 빛을 어떻게 반사하는지를 나타낸다.
  - 조명이 활성화 되면, glColor는 무시되고 재질이 대신 사용된다.

조명 모델

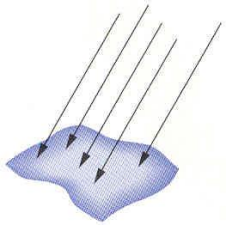
- 조명과 음영 넣기
  - 조명 기능 활성화 (Enable lighting)
  - 광원 정의 (Specify lights)
  - 음영 모드 정의 (Shading mode)
  - 법선 벡터 정의 (surface normal)
  - 재질 특성 정의 (surface material)
  - 조명모델 정의 (lighting model)

조명 모델

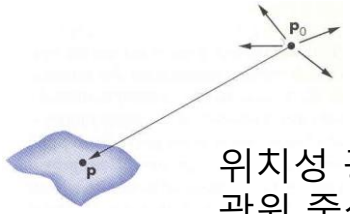
- 조명 기능 활성화
  - 조명 기능을 활성화 해야 한다.
    - **glEnable (GL_LIGHTING);**
      - 조명 사용이 가능하도록 한다.
      - 장면 내에 있는 각 꼭지점의 색상을 결정할 때 재질 속성과 조명 인자를 계산에 넣게 된다.
      - 조명기능이 활성화 되면,
        - » **물체의 색은 광원과 물체의 특성에 의해서 결정**
        - » glColor에 의해 정의된 색은 무시
      - 이러한 인자를 지정하는 부분
        - » 그리기가 이루어지기 직전에 수행

조명 모델

- 광원 정의
  - 광원: 세기, 색, 특정 위치, 방향
  - OpenGL에서는 독립적인 광원을 8개까지 사용할 수 있다.
    - 광원 번호: GL_LIGHTx (0 ≤ x ≤ 7)
    - 주변광 성분: GLfloatf ambientLight[ ];
    - 산란반사광 성분: GLfloatf diffuseLight[ ]
    - 거울반사광 성분: GLfloatf specularLight[ ]
    - 광원의 위치: GLfloatf lightPos[ ];
      - 마지막 값이 0.0이면 – 원점에서 좌표 값을 향하는 벡터 방향의 방향성 광원
      - 마지막 값이 1.0이면 – 지정된 좌표가 광원의 위치인 위치성 광원



방향성 광원
방향만 고려



위치성 광원
광원 중심으로 모든 방향에 영향

  - 광원을 켠다: **glEnable (GL_LIGHTx);**
  - glLight 함수로 광원의 속성을 설정한다.

조명 모델

- **glLight**: 사용 가능한 8개의 광원 중 하나의 광원 인자 지정
  - glLightf (GLenum lightID, GLenum pname, GLfloat param);
  - glLighti (GLenum lightID, GLenum pname, GLint param);
  - glLightfv (GLenum lightID, GLenum pname, const GLfloat *param);
  - glLightiv (GLenum lightID, GLenum pname, const GLint *param);
  - Parameters:
    - lightID : 어느 광원을 수정할 지 지정한다.
      - GL_LIGHT0 ~ GL_LIGHT7
    - pname: 사용할 조명 인자를 지정
      - GL_AMBIENT: 주변 조명의 세기 (r, g, b, a)를 지정 (초기값은 (0, 0, 0, 0))
      - GL_DIFFUSE: 산란 반사 조명의 세기 (r, g, b, a)를 지정 (1, 1, 1, 1))
      - GL_SPECULAR: 거울 반사 조명의 세기 (r, g, b, a)를 지정 (초기값은 (1, 1, 1, 1))
      - GL_POSITION: 조명의 (x, y, z, w) 값 지정 (초기값은 (0, 0, 1, 0))
      - GL_SPOT_DIRECTION: 스포트라이트의 방향 벡터 (초기값은 (0, 0, -1))
      - GL_SPOT_CUTOFF: 스포트라이트의 확산 각도(초기값은 180.0)
      - GL_SPOT_EXPONENT: 스포트라이트 지수 (초기값은 0)
      - GL_CONSTANT_ATTENUATION: 빛이 점점 흐려지는 감쇠율 (초기값은 1)
      - GL_LINEAR_ATTENUATION: 빛이 점점 흐려지는 감쇠율 (1차 계수, 초기값은 0)
      - GL_QUATRATIC_ATTENUATION: 빛이 점점 흐려지는 감쇠율 (2차 계수, 초기값은 0)
    - param: pname의 값

조명 모델

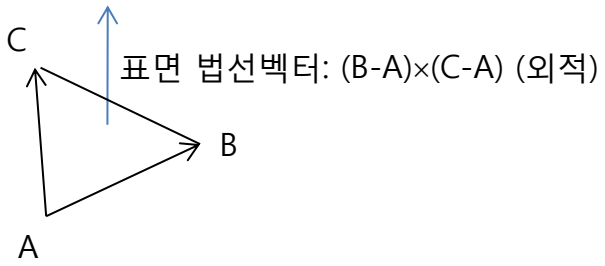
- 예) (1, 2, 3)에서 주변조명은 녹색, 산란 반사 조명은 적색, 반사광은 백색인 광원 0를 좌표 (1, 2, 3)에 설치하기

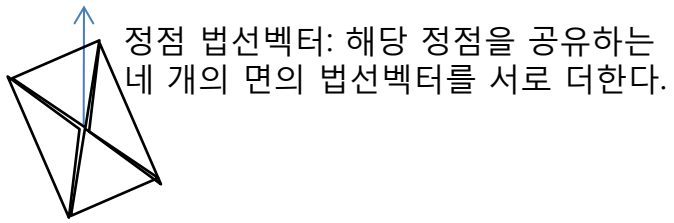
```
GLfloat AmbientLight[] = {0.0f, 1.0f, 0.0f, 1.0f};
GLfloat DiffuseLight[] = {1.0f, 0.0f, 0.0f, 1.0f};
GLfloat SpecularLight[] = {1.0, 1.0, 1.0, 1.0};
Glfloat lightPos[] = {1.0, 2.0, 3.0. 1.0};
…
// 조명을 사용하도록 설정
glEnable (GL_LIGHTING);

// 조명 설정
glLightfv (GL_LIGHT0, GL_AMBIENT, AmbientLight);
glLightfv (GL_LIGHT0, GL_DIFFUSE, DiffuseLight);
glLightfv (GL_LIGHT0, GL_SPECULAR, SpecularLight);
glLightfv (GL_LIGHT0, GL_POSITION, lightPos);
glEnable (GL_LIGHT0);
```

조명 모델

- 음영 모드 정의
  - glShadeModel() 함수를 사용하여 쉐이딩을 설정한다.
    - GL_FLAT 또는 GL_SMOOTH 사용하여 쉐이딩 설정
    - 지엘은 플랫 셰이딩과 구로 셰이딩 지원

- 법선 벡터 정의
  - 반사광의 세기를 계산하기 위해 법선 벡터 사용
  - 정점의 법선 벡터와 표면 법선 벡터는 수동으로 설정해야 한다.



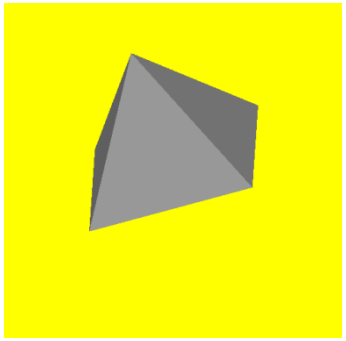


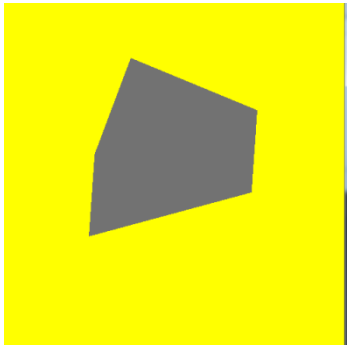
조명 모델

- 법선 벡터 설정
  - **glNormal3f (GLfloat x, GLfloat y, GLfloat z);**
    - 법선 벡터는 단위벡터라야 한다.
    - 곡선일 때는 각각의 정점에 법선 벡터를 설정한다.
    - 사용 예)

```
glBegin (GL_TRIANGLE);
    glNormal3f (0.0, 1.0, 0.0);  // 아래의 3개의 정점 모두에 적용
    glVertex3f (0.0, 0.0, -1.0);
    glVertex3f (0.0, 1.0, 0.0);
    glVertex3f (-1.0, 0.0, 0.0);
glEnd ();
```

  - 단위벡터로 만들 때: **glEnable (GL_NORMALIZED);**





조명 모델

- 물체면 특성 정의 (Material)
  - 물체의 특성: 물체면의 색과 물체면의 매끄러움
    - 물체면의 색: 주변조명, 산란반사조명, 거울반사조명 등 반사의 종류별로 물체면에서 어떤 크기로 빛을 반사하는지를 의미
    - 물체면의 매끄러움: 거울반사광의 광택 계수
  - 표면이 어떤 물체로 만들어질 것인가를 결정한다.
  - 물체가 얼마나 빛을 반사 할 것인가를 결정한다.
  - 적당한 조명 효과를 만들기 위해서 재질의 특성을 결정해야 한다.
    - Ambient: 특정한 방향이 없이 들어오고, 모든 방향으로 반사
    - Diffuse: 특정한 방향에서 들어오고, 모든 방향으로 반사
    - Specular: 특정한 방향에서 들어오고, 특정 방향으로 뚜렷하게 반사
    - Shininess: 특정한 방향에서 들어오고, 특정 방향으로 같은 양이 반사
    - Emissive: 물체 자체가 발산하는 빛

재질 속성 정하기

- 물체면의 재질 속성 설정하기
  - **glMaterial** 함수 사용: 조명 모델에서 사용할 재질 인자를 결정
    - void glMaterialf (GLenum face, GLenum pname, GLfloat param);
    - void glMaterialI (GLenum face, GLenum pname, GLint param);
    - void glMaterialfv (GLenum face, GLenum pname, const GLfloat *param);
    - void glMaterialiv (GLenum face, GLenum pname, const GLint *param);

      - face: 재질 속성이 폴리곤의 어디에 적용할지를 결정
        - » GL_FRONT / GL_BACK / GL_FRONT_AND_BACK
      - pname: 재질 속성 설정
        - » GL_AMBIENT: 주변 반사에 대한 물체색 (초기값은 (0.2, 0.2, 0.2, 1.0))
        - » GL_DIFFUSE: 산란 반사에 대한 물체색 (초기값은 (0.8, 0.8, 0.8, 1.0))
        - » GL_SPECULAR: 거울 반사에 대한 물체색 (초기값은 (0.0, 0.0, 0.0, 1.0))
        - » GL_SHINIESS: 거울 반사의 광택 계수 (초기값은 0.0, 0 ~ 128사이의 값)
        - » GL_EMISSION: 자체적 빛 방출 (초기값은 (0.0, 0.0, 0.0, 1.0))
        - » GL_AMBIENT_AND_DIFFUSE: 주변 반사와 산란 반사에 대한 물체색
      - params: pname으로 설정된 인자의 값

재질 속성 정하기

- 예)

```
GLfloat gray[] = {0.75f, 0.75f, 0.75f, 1.0f};
GLfloat  specref[] =  { 1.0f, 1.0f, 1.0f, 1.0f };


glMaterialfv (GL_FRONT, GL_AMBIENT_AND_DIFFUSE,  gray);
glMaterialfv(GL_FRONT, GL_SPECULAR, specref);
glMateriali(GL_FRONT,  GL_SHININESS, 64);
glBegin (GL_TRIANGLES);
        glVertex3f (-15.0f, 0.0f, 30.0f);
        glVertex3f (0.0f, 15.0f, 30.0f);
        glVertex3f (0.0f, 0.0f, -50.f);
glEnd ();
```

재질 속성 정하기

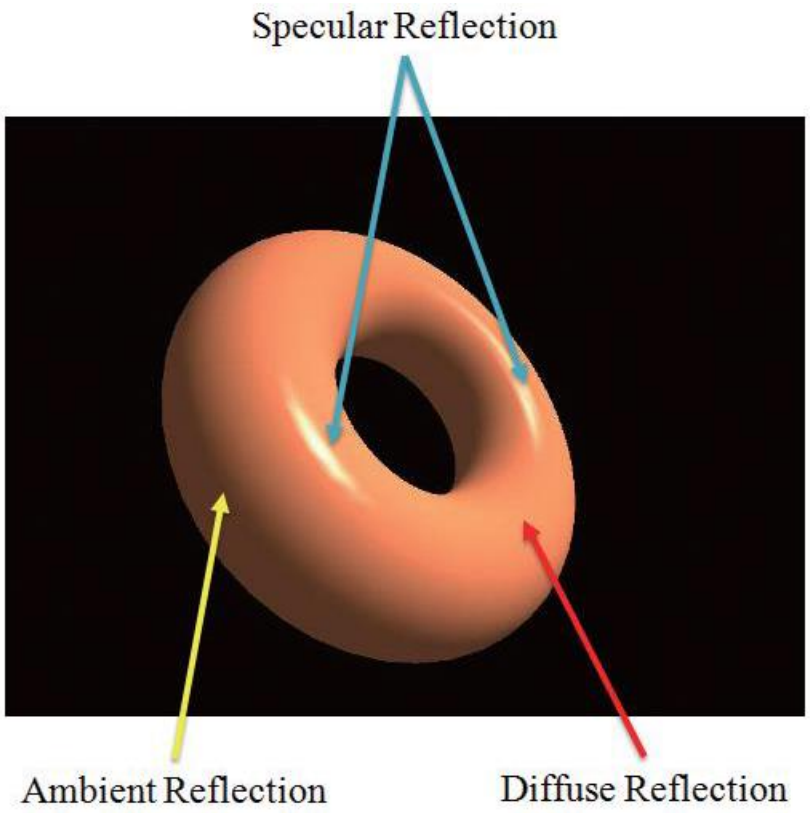
- **glColorMaterial**: glMaterial 함수를 호출하지 않고, glColor를 이용하여 재질 색을 설정. 현재 색대로(glColor로 설정된 색상) 재질 속성을 설정.
  - glEnable 함수로 GL_COLOR_MATERIAL 설정하고 사용
  - glColorMaterial (GLenum face, GLenum mode);
  - Parameter:
    - Face: 현재 설정할 면
      - » GL_FRONT / GL_BACK / GL_FRONT_AND_BACK
    - Mode: 어떤 재질 속성이 현재 색상 설정의 영향을 받을 것인지를 결정
      - » GL_EMISSION / GL_AMBIENT / GL_DIFFUSE / GL_SPECULAR / GL_AMBIENT_AND_DIFFUSE

재질 속성 정하기

- 예)

```
GLfloat ambientLight[] = {1.0f, 1.0f, 1.0f, 1.0f};
GLfloat  specref[] =  { 1.0f, 1.0f, 1.0f, 1.0f };

// 조명 효과를 설정한다.
glEnable (GL_LIGHTING);
glLightModelfv (GL_LIGHT_MODEL_AMBIENT, ambientLight);

// 재질 컬러 트래킹을 설정한다.
glEnable (GL_COLOR_MATERIAL);
glColorMaterial (GL_FRONT, GL_AMBIENT_AND_DIFFUSE);

glMaterialfv(GL_FRONT, GL_SPECULAR, specref);
glMateriali(GL_FRONT, GL_SHININESS, 128);

glColor3f (0.75f, 0.75f, 0.75f);
glBegin (GL_TRIANGLES);
        glVertex3f (-15.0f, 0.0f, 30.0f);
        glVertex3f (0.0f, 15.0f, 30.0f);
        glVertex3f (0.0f, 0.0f, -50.f);
glEnd ();
```

재질 속성 정하기


Specular Reflection
Ambient Reflection
Diffuse Reflection

조명 모델

- 조명 모델 정의
  - 조명 모델 전반에 관한 변수 조절
- 조명 효과 모델 설정
  - **glLightModel**: openGL에서 사용하는 조명 모델 인자를 설정, 특정 번호의 조명이 아니라 전역 설정이므로 장면 전체에 적용
    - glLightModelf (GLenum pname, GLfloat param);
    - glLightModeli (GLenum pname, GLint param);
    - glLightModelfv (GLenum pname, const GLfloat *params);
    - glLightModeliv (GLenum pname, const GLint *params);
  - Parameters:
    - pname: 조명 모델 인자를 지정
      - GL_LIGHT_MODEL_AMBIENT: 전역 주변광 설정 (초기값은 (0.2, 0.2, 0.2, 1.0))
        - » 위의 모델 중 다음 2개만 사용 가능: glLightModelfv, glLightModeliv
      - GL_LIGHT_MODEL_LOCAL_VIEWER: 반사광 각도 계산 (초기값은 GL_FALSE)
        - » 산란반사광의 각도가 –z축 방향으로 평행
        - » GL_FALSE(0.0): 조명이 평행하게 비친다.
        - » GL_TRUE(1.0): 정점의 방향에 따라 비친다.
      - GL_LIGHT_MODEL_TWO_SIDE: 폴리곤의 양면이 조명을 받을지를 결정 (초기값은 GL_FALSE)
        - » GL_FALSE(0.0): 폴리곤의 앞면만 빛을 받는다
    - params: 빛의 값 (RGBA 값)

조명 모델

- 사용 예)

```
GLfloat ambientLignt[] = {0.2f,  0.2f,  0.2f,  1.0f};

glEnable (GL_LIGHTING);
glLightModelfv (GL_LIGHT_MODEL_AMBIENT, ambientLight);

glLightModelf (GL_LIGHT_MODEL_LOCAL_VIEWER, 0.0);

glLightModelf (GL_LIGHT_MODEL_TWO_SIDE, 0.0);
```

모두 합쳐서 조명 효과 프로그래밍 하기

- 조명을 위한 변수 값 설정하기
  - Ambient light: GLfloat ambientLight[] = {…, …, …, …};
  - Diffuse light : GLfloat diffuseLight[] = {…, …, …, …};
  - Specular light : GLfloat specular[] = {…, …, …, …};
  - Light position: GLfloat lightPos[] = {…, …, …, …};
  - Specular Reflectance: GLfloat specref[] = {…, …, …, …};

- 조명 효과 설정하기
  - glEnable (GL_LIGHTING)

- 조명 모델 설정하기
  - glLightModelfv(GL_LIGHT_MODEL_AMBIENT, ambientLight);

- 조명 켜기 (0 ~ 7)
  - 광원 설정하기
    - glLightfv (GL_LIGHTx, GL_AMBIENT, ambientLight); // Ambient light
    - glLightfv (GL_LIGHTx, GL_DIFFUSE, diffuseLight); // Diffuse light
    - glLightfv (GL_LIGHTx, GL_SPECULAR, specular); // Specular light
  - glEnable (GL_LIGHTx)

모두 합쳐서 조명 효과 프로그래밍 하기

- 물체 표면의 재질 설정하기
  - 컬러 트랙킹 설정하기
    - glEnable (GL_COLOR_MATERIAL)
  - 현재 색의 트랙킹을 위해서 재질 속성 설정하기
    - glColorMaterial (GL_FRONT, GL_AMBIENT_AND_DIFFUSE);
  - 조명 모델을 위한 재질 변수 값 설정하기
    - 특정 면에 대해 재질 값을 설정한다.
      - 앞면:

        glMaterialfv(GL_FRONT, GL_SPECULAR,specref);

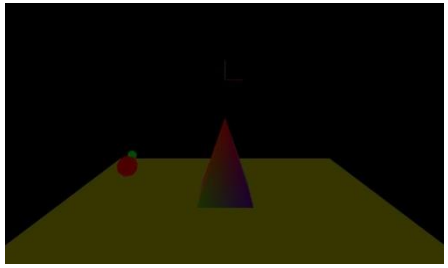
        glMateriali(GL_FRONT, GL_SHININESS,128);

        - GL_SPECULAR / GL_SHININESS / GL_AMBIENT / GL_DIFFUSE…
      - 뒷면: glMaterialfv (GL_BACK, …, …);
      - 앞뒷면: glMaterialfv (GL_FRONT_AND_BACK, …, …);

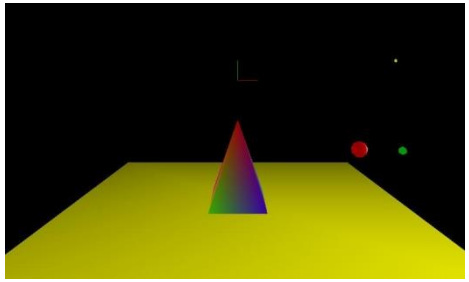
실습 22

- 조명 2개를 사용 해 본다.
  - 가운데에 피라미드를 그리고 그 주위를 지구가, 지구 주위를 달이 공전한다.
  - 화면의 좌우에 조명을 각각 1개씩 놓는다.
  - 명령어
    - A/a: 광원의 ambient light를 높게 / 낮게
    - D/d: 광원의 diffuse light를 높게 / 낮게
    - S/s: 광원의 specular light를 높게 / 낮게
    - 1: 1번 조명을 켠다/끈다
    - 2: 2번 조명을 켠다/끈다
  - 조명의 위치에 작은 콘을 그려 조명을 표시한다.

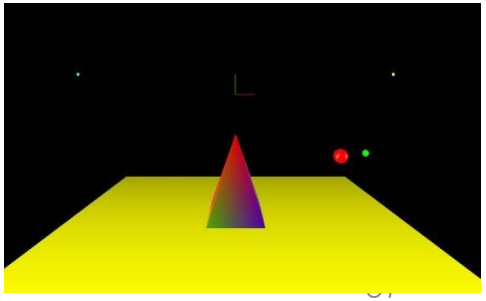
달, 조명이 초기에는 꺼져있음



조명 1이 켜졌음

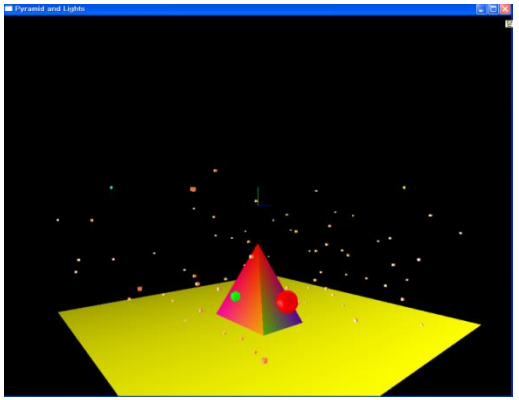


조명 2가 켜졌음



실습 23

- 눈 내리는 피라미드 애니메이션 구현하기
  - 원근 투영을 사용한다.
    - 바닥: 평면을 사용하여 바닥을 그린다.
    - 피라미드: 평면에 2개의 피라미드를 그린다.
    - 피라미드 주변을 각각 지구와 달이 공전한다.
    - 바닥과 피라미드는 smooth 쉐이딩을 한다.
    - 은면 제거를 한다.
  - 바닥의 네 꼭지점 위에 4개의 조명을 넣는다.
    - 조명을 각각 조정할 수 있도록 한다. (실습 22)
  - 화면의 위 부분에서 눈이 내린다.
    - 눈: 오픈지엘이 제공하는 3차원 도형 사용
      - Cone으로 눈 모양을 만들 수도 있다.
    - 눈은 임의의 높이에서 아래로 떨어진다.
    - 눈이 바닥에 닿으면 다시 위에서 내린다.
  - 화면 애니메이션
    - 초기 화면에서 줌인을 (z축으로 이동) 시켜서 가까이 간다.
    - 특정 위치에 도달하면 y축으로 4바퀴 회전한다.
    - 회전이 끝나면 다시 제자리로 줌아웃한다.



실습 24

실습 25

실습 26

실습 27