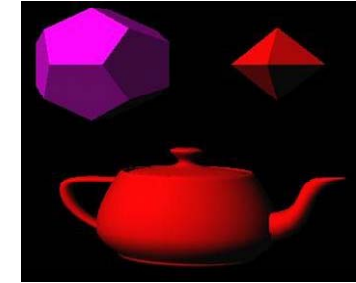
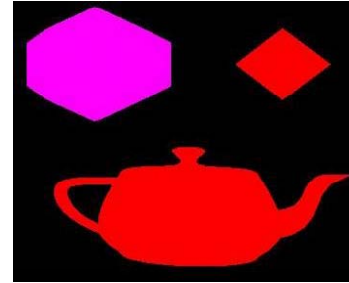


Lighting

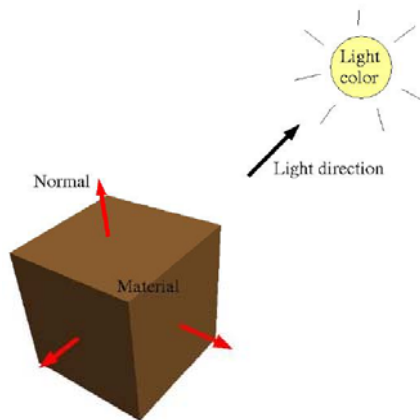
321190
2011년 봄학기
5/17/2011
박경신

OpenGL Lighting

- OpenGL의 조명에는 3가지 요소가 필요
 - 광원 (Lights)
 - 재질 (Materials)
 - 면의 법선벡터 (Normals)



OpenGL Lighting



OpenGL Lighting

- OpenGL에서 제공하는 조명모델
 - 환경광/주변광 (ambient lights)
 - 점광원 (point lights)
 - 방향성광원/평행광원/원거리광원 (directional lights)
 - 점적광원 (spot lights)
- 거리에 따른 빛의 세기 감소 (attenuation)
 - Physical attenuation: $I(P) = \frac{I(P_0)}{\|P - P_0\|^2}$
 - OpenGL attenuation:
 - Default $a = 1, b = 0, c = 0$

$$I(P) = \frac{I(P_0)}{a + bd + cd^2}$$

OpenGL Lighting

- OpenGL에서 lighting을 사용하려면 GL_LIGHTING을 활성화해야 한다.
`glEnable(GL_LIGHTING);`
- 또한, 적어도 하나의 광원을 활성화해야 한다.
`glEnable(GL_LIGHT0);`
- OpenGL에서는 최대 8개의 광원을 지정할 수 있다
`GL_LIGHT0, ..., GL_LIGHT7`
- 최대 광원수 얻기:
`glGetIntegerv(GL_MAX_LIGHTS, GLint *num_lights);`
- OpenGL에서 lighting 비활성화
`glDisable(GL_LIGHTING);`

OpenGL Light Sources

- 광원의 속성은 `glLight(..)` 함수에 의해 지정된다.
- 광원은 **위치**(position)와 **색**(color)을 갖는다.
- 광원의 **강도** (intensity)는 색의 강도에 의해 결정된다.

```
GLfloat position[] = {0, 1, 0, 0}; // directional light
GLfloat white_color[] = {1, 1, 1, 1};
glEnable(GL_LIGHT0);
glLightfv(GL_LIGHT0, GL_POSITION, position); // light position
glLightfv(GL_LIGHT0, GL_DIFFUSE, white_color); // light color
```

OpenGL Light Source Position

- 광원은 infinite light (혹은 directional light)이거나 local light (혹은 positional light)이다.
 - Infinite light source의 모든 빛은 같은 방향을 갖는다.
 - Local light source는 공간의 특정지점에서 오는 빛이다.
 - 광원의 위치의 4번째 값이 0이면 infinite light (directional light)이고, 1이면 local light (point light)이다.
- 예제:
`// directional light comes from above`
`GLfloat position0[] = {0, 1, 0, 0};`
`glLightfv(GL_LIGHT0, GL_POSITION, position0);`
`// point light comes from point (0, 1, 0)`
`GLfloat position1[] = {0, 1, 0, 1};`
`glLightfv(GL_LIGHT1, GL_POSITION, position1);`

OpenGL Light Source Position

- 광원의 위치는 현재 변환 (transformation)에 영향을 받는다.
- Light은 세계 좌표계 (world coordinates system)에서 정의될 수 있도록 카메라 변환 후에 지정하는 것이 좋다.
- Light은 또한 객체와 같이 변환 행렬에 의해 움직일 수 있다.
- 예제:
`// make a point light comes from position (5, 0, 7)`
`GLfloat position[] = {0, 0, 0, 1};`
`glPushMatrix();`
`glTranslatef(5, 0, 7);`
`glLightfv(GL_LIGHT0, GL_POSITION, position);`
`glPopMatrix();`

OpenGL Light Sources

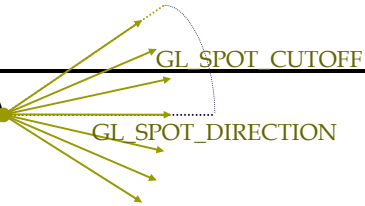
□ 점적/집중 광원 (spot light source)

- 방향지정 벡터 (spot light direction)
- 절단 각도 (spot light cutoff)
- 높은 계수 (spot light exponent) 를 사용하면 보다 많은 spot light 생김

```
GLfloat direction[] = {0.0, -1.0, 0.0};  
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, direction);  
glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 15.0f);  
glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, 10.0f)
```

□ 빛의 감쇠 (attenuation): $1/(a + bd + cd^2)$

- Default a = 1, b=0, c=0
- ```
glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 2.0f);
glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, 1.0f);
glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, 0);
```



## OpenGL Multiple Lights

- OpenGL에서는 GL\_LIGHT0, GL\_LIGHT1, .. 등으로 지정함으로써 여러 개의 광원을 동시에 활성화할 수 있다.
- 최대 8개까지의 광원을 사용할 수 있다.  

```
GLfloat position[] = {-1, 1, 2, 0};
glEnable(GL_LIGHT1);
glLightfv(GL_LIGHT1, GL_POSITION, position);
```
- 너무 많은 light을 사용하면 계산량이 늘어나게 되고 따라서 렌더링 속도를 늦출 수 있다.

## OpenGL Materials

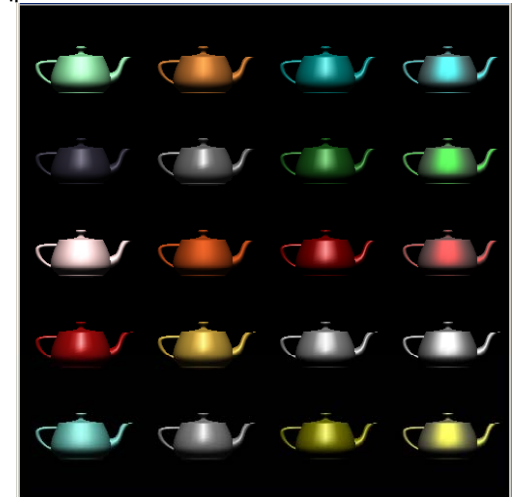
- 재질 속성은 표면이 빛을 어떻게 반사하는지를 나타내준다.
- 재질은 기본적으로 표면의 색을 말한다.
- 조명이 활성화되면, glColor는 무시되고 재질(material)이 대신 사용된다.
- 재질 속성은 glMaterial 함수를 사용하여 지정한다.

```
glMaterialfv(face, property, value)
```

- Face: e.g. GL\_FRONT, GL\_BACK, GL\_FRONT\_AND\_BACK
- Property: e.g. GL\_AMBIENT, GL\_DIFFUSE, GL\_SPECULAR, GL\_SHININESS, GL\_EMISSION, etc
- Value: the reflection coefficients
- Nonzero GL\_EMISSION makes an object appear to be giving off light of that color

## Example

- Teapot의 재질예제



## OpenGL Light Color

- OpenGL에서 광원은 색(color)과 강도(intensity)를 가지고 있으며, 기본값은 백색이다.
- 우리가 보는 객체의 색은 빛의 색 (light color)과 재질의 색 (material color) 둘 간의 합성으로 결정된다.
  - 빛의 색은 얼마나 RGB 빛이 객체에서 빛나는지를 말한다.
  - 재질의 색은 얼마나 RGB 빛이 객체에서 반사되는지를 말한다.
  - White light \* red material = red
  - Red light \* white material = red
  - Red light \* blue material = black
  - Yellow light \* cyan material = green

## OpenGL Ambient/Diffuse/Specular Light

- OpenGL 조명 모델에서는 3가지 형태의 반사광을 고려함.
- Ambient light (환경광): 다른 일반 표면에서 반사되어 나오는 빛. 장면을 전반적으로 밝게 함.
- Diffuse light (난반사광): 특정 방향으로 진행하다가 표면에 닿으면 모든 방향으로 동일하게 반사됨. 관찰자의 위치와 무관함. 가장 일반적인 형태임.
- Specular light (정반사광): 특정 방향으로 진행하다가 표면에 닿으면 한 방향으로 강하게 반사됨. 반짝이는 표면을 모델링할 때 이용됨.

## OpenGL Ambient Light

- 환경광은 간접적으로 들어오는 빛을 시뮬레이션함.
- 표면의 방향과는 상관없이 사방에 일정한 밝기의 빛이 고르게 퍼져있다고 가정함.
- 환경광 효과는 광원의 환경광색과 재질의 환경광색으로 지정됨.

```
GLfloat gray[] = {0.1, 0.1, 0.1, 1};
GLfloat white[] = {1, 1, 1, 1};
glLightfv(GL_LIGHT0, GL_AMBIENT, gray);
glMaterialfv(GL_FRONT, GL_AMBIENT, white);
```
- 또한 glLightModel 함수를 사용하여 전역적 환경광을 지정할 수 있음.

```
glLightModelfv(GL_LIGHT_MODEL_AMBIENT, gray);
```

## OpenGL Specular Light

- 정반사광은 표면에 반짝이는 highlights를 생성함.
- 정반사광의 3가지 속성:
  - 재질의 정반사 색
  - 재질의 반짝임 (shininess)
  - 광원의 정반사 색

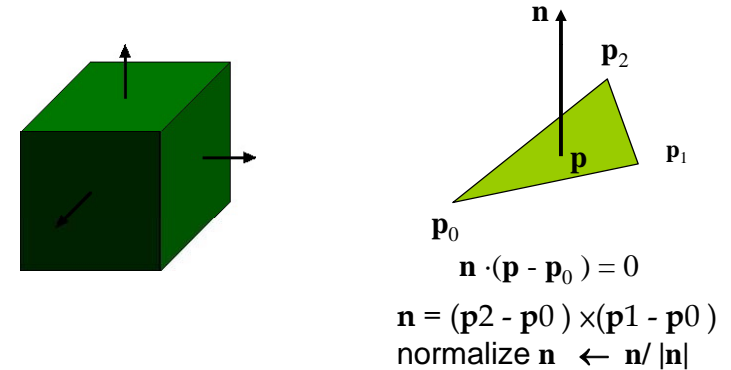
```
GLfloat white[] = {1, 1, 1, 1};
glLightfv(GL_LIGHT0, GL_SPECULAR, white);
glMaterialfv(GL_FRONT, GL_SPECULAR, white);
glMaterialfv(GL_FRONT, GL_SHININESS, 50);
```
- GL\_SHININESS값이 높으면 작은 면적의 highlight이 생성
- 정반사광을 비활성화하고자 하면, 정반사광색을 검정색 (0, 0, 0)으로 지정함.

## Surface Orientation

- Lighting에서 빛이 표면에서 얼마나 반사되는지 계산하기 위해서 OpenGL은 표면이 어느 방향을 향하고 있는지 알아야 한다.
- 표면의 방향 (surface orientation)과 빛의 방향은 빛의 난반사광 (diffuse light)의 밝기를 결정짓는다.
- 표면의 방향 (surface orientation), 빛의 방향 (light direction), 관찰자를 향하는 방향벡터 (direction to the viewer)는 정반사광 (specular light)의 밝기를 결정짓는다.

## Surface Normal

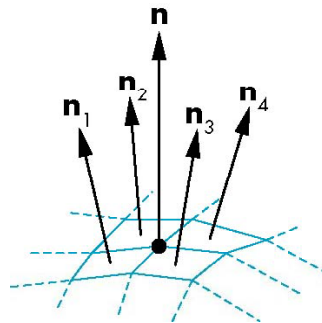
- 표면의 방향 (orientation)은 표면에 수직을 이루는 법선벡터에 의해 정의된다.
- 면의 법선 벡터는 (벡터의 길이가 1인) 단위벡터 (unit vector) 이어야 한다.



## Vertex Normal

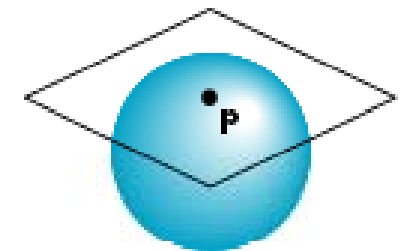
- 정점의 법선벡터
  - 그 정점을 공유하는 근접한 면의 법선벡터의 평균치를 계산하여 사용함.

$$\mathbf{n} = \frac{\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 + \mathbf{n}_4}{|\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 + \mathbf{n}_4|}$$



## Normal to Sphere

- Implicit function of Sphere:
  - $f(x,y,z)=0$
- Unit Sphere:
  - $f(\mathbf{p})=\mathbf{p}\cdot\mathbf{p}-1=0$
- 구의 법선벡터
  - $\mathbf{n} = [\partial f/\partial x, \partial f/\partial y, \partial f/\partial z]^T = \mathbf{p}$



## OpenGL Normal

- OpenGL에서 면의 법선벡터는 glNormal 함수로 지정
- OpenGL의 조명 계산에서는 각 정점의 법선벡터를 사용함
- 예제:

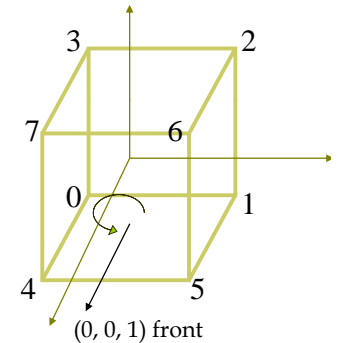
```
glBegin(GL_TRIANGLES);
glNormal3f(0, 1, 0);
glVertex3f(-1, 0, 0);
glNormal3f(0, 1, 0);
glVertex3f(1, 0, 0);
glNormal3f(0, 1, 0);
glVertex3f(0, 0, 2);
glEnd();
```

## OpenGL Normal

- Shaded Cube를 그린다.

```
GLfloat vertex[][3] = {
 {-1.0,-1.0,-1.0}, { 1.0,-1.0,-1.0},
 { 1.0, 1.0,-1.0}, {-1.0, 1.0,-1.0},
 {-1.0,-1.0, 1.0}, { 1.0,-1.0, 1.0},
 { 1.0, 1.0, 1.0}, {-1.0, 1.0, 1.0}
};

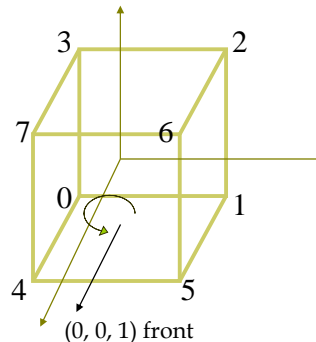
GLfloat normal[][3] = {
 { 1.0, 0.0, 0.0}, // right
 { 0.0, 1.0, 0.0}, // top
 { 0.0, 0.0, 1.0}, // front
 {-1.0, 0.0, 0.0}, // left
 { 0.0, -1.0, 0.0}, // bottom
 { 0.0, 0.0, -1.0}, // back
};
```



## OpenGL Normal

```
void polygon(int n, int b, int c, int d)
{
 glBegin(GL_POLYGON);
 glNormalfv(normal[n]);
 glVertex3fv(vertex[a]);
 glVertex3fv(vertex[b]);
 glVertex3fv(vertex[c]);
 glVertex3fv(vertex[d]);
 glEnd();
}

void cube()
{
 polygon(0, 5, 1, 2, 6); // right
 polygon(1, 6, 2, 3, 7); // top
 polygon(2, 6, 7, 4, 5); // front
 polygon(3, 4, 7, 3, 0); // left
 polygon(4, 4, 0, 1, 5); // bottom
 polygon(5, 0, 3, 2, 1); // back
}
```



## Advanced OpenGL Lighting

- OpenGL에서 제공하는 lighting관련 기능들
  - Spotlights
  - Attenuation
  - Emissive materials
  - Two-sided lighting model
  - Color-material mode
  - Secondary specular color
  - Automatic rescaling of normals