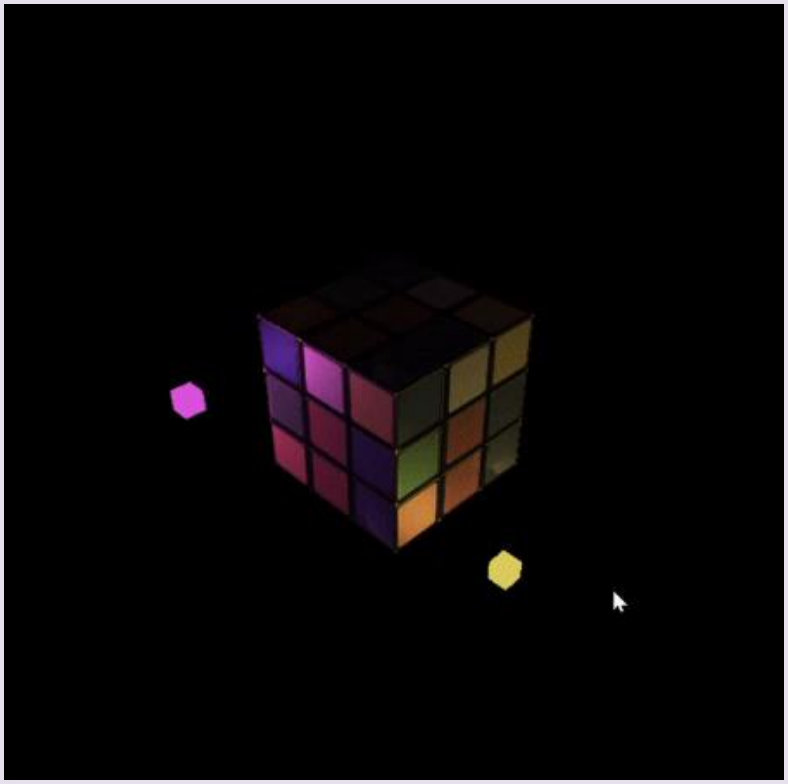
디지털 그래픽스 강의 12주차

# Lighting & Shading *(1)*

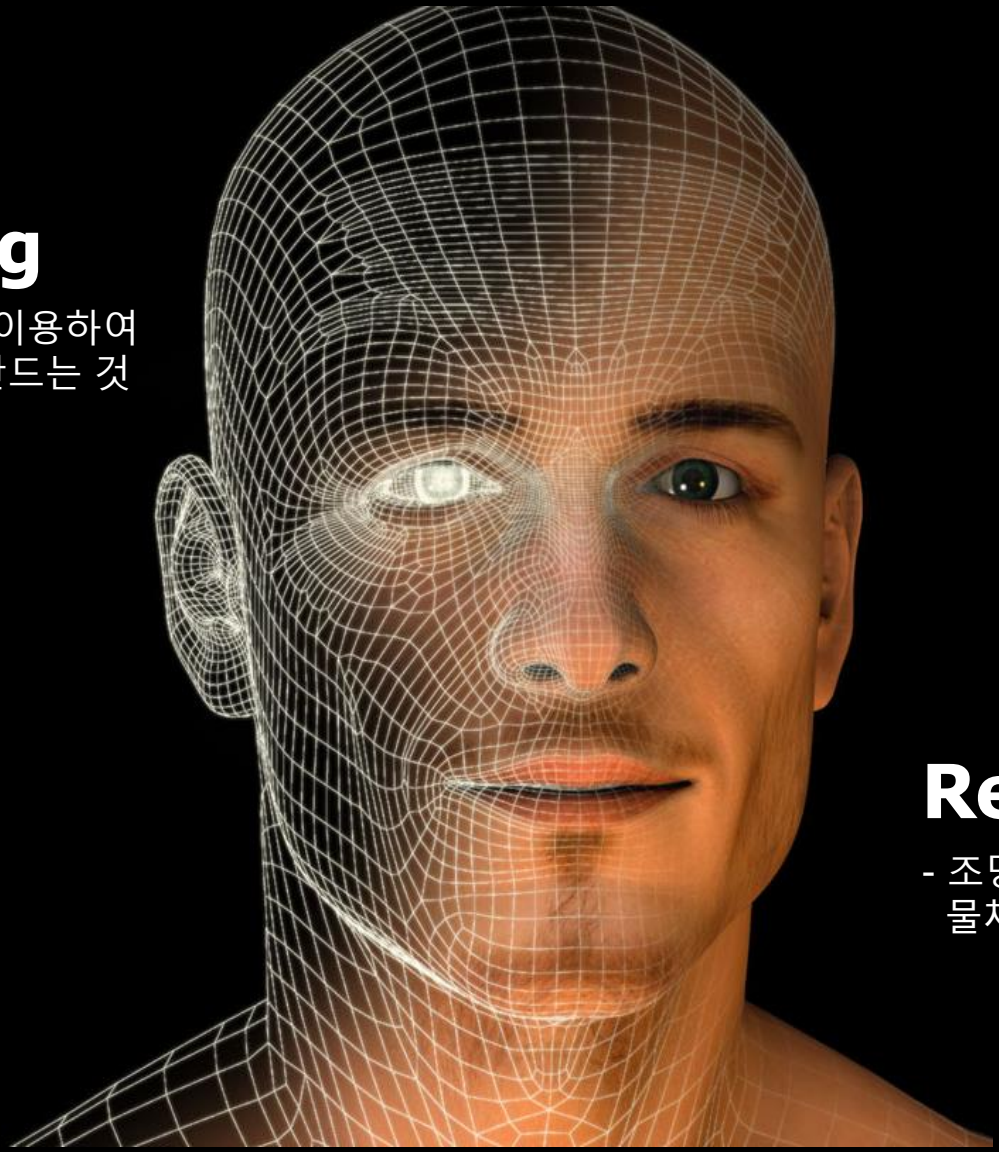
## Illumination model



# Two parts of Computer Graphics

## Modeling

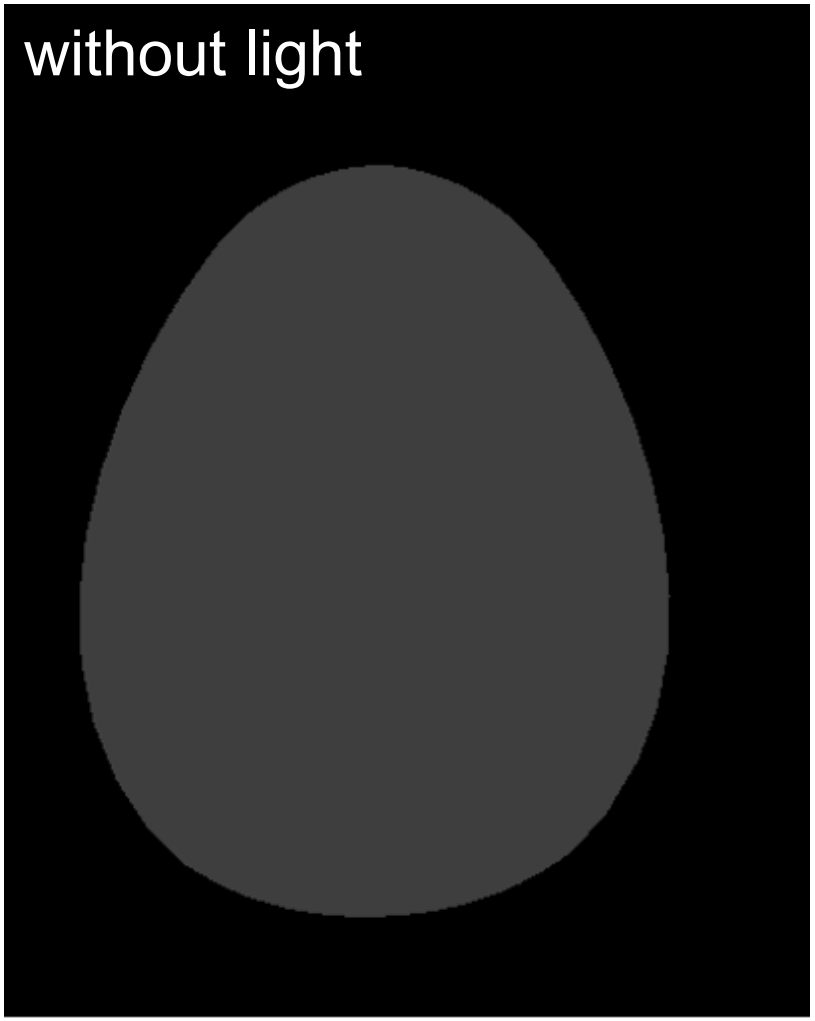
- Vertex / Face를 이용하여
  사물의 형태를 만드는 것

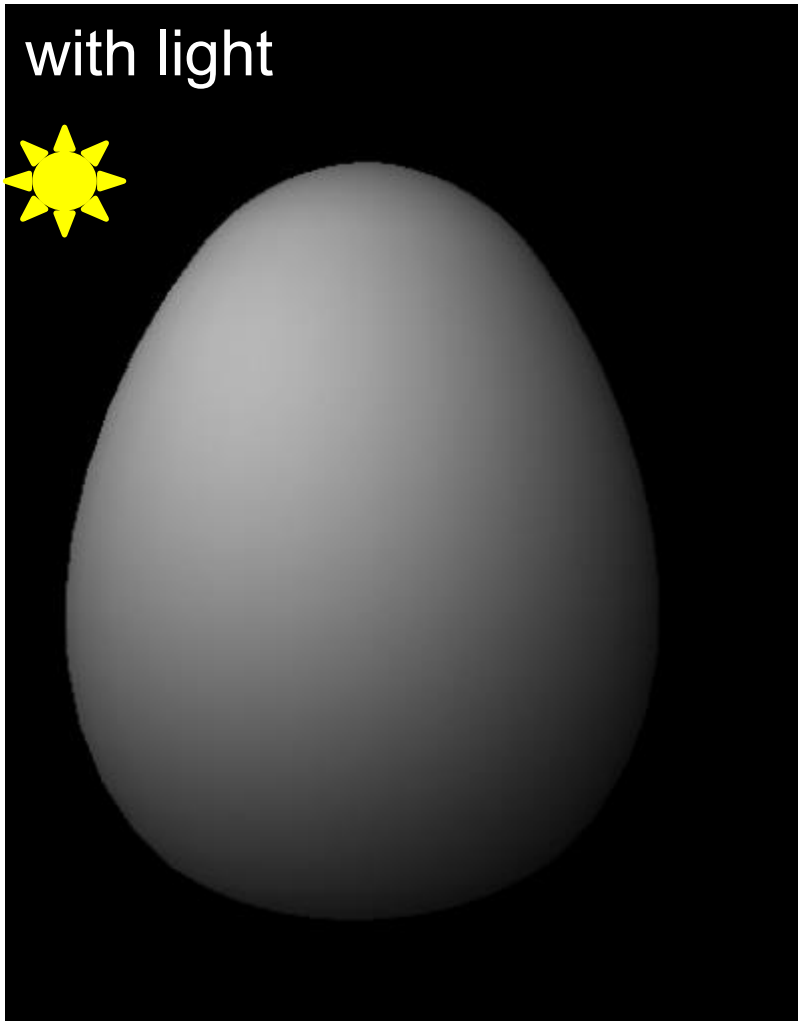


## Rendering

- 조명과 음영을 이용해서
  물체의 색을 결정하는 것

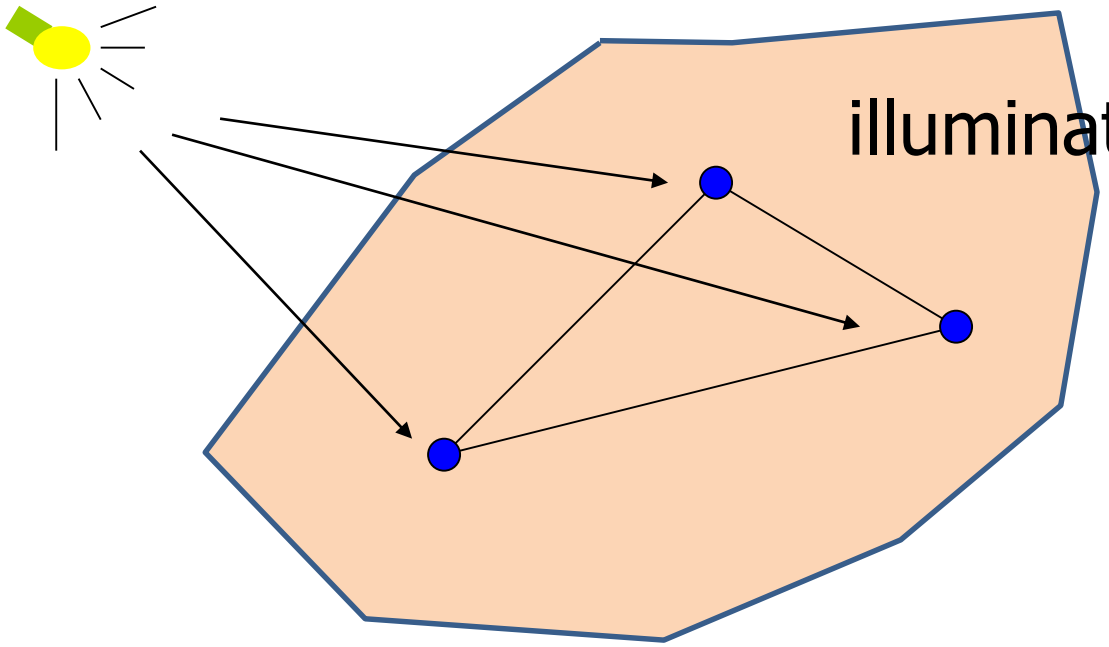
# A scene without & with a light



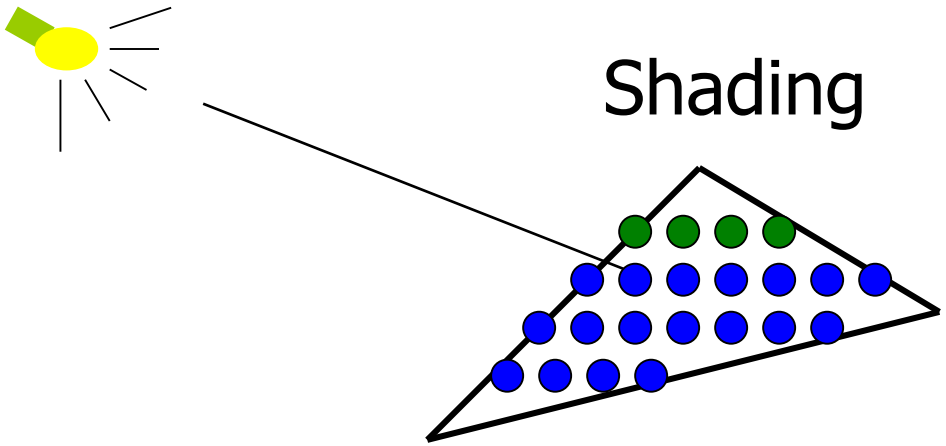
# Lighting (illumination)

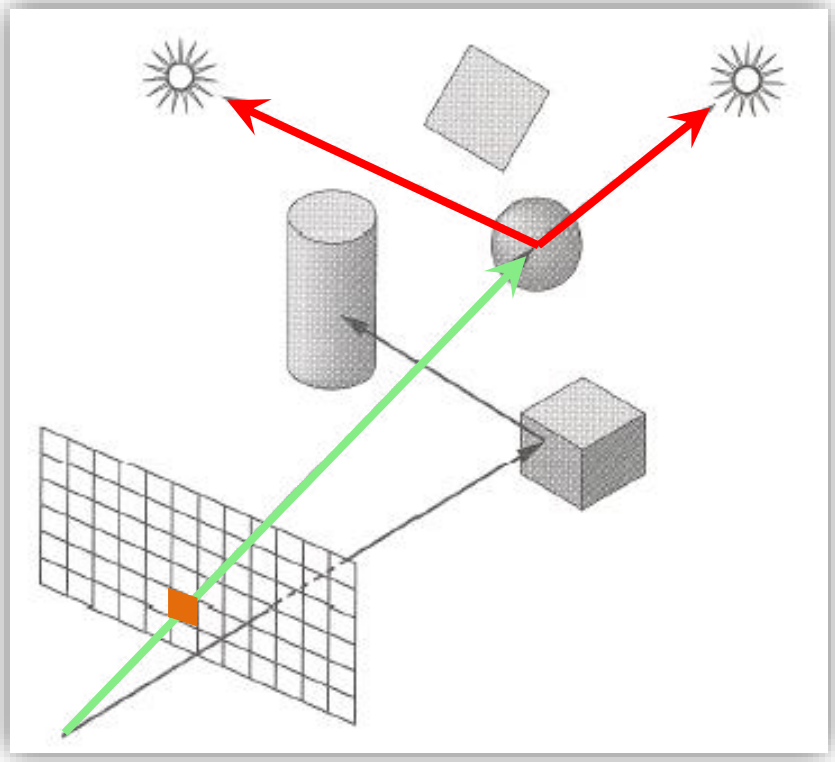
- **Model the interaction of light with surface points** to determine their final color and brightness




# Shading

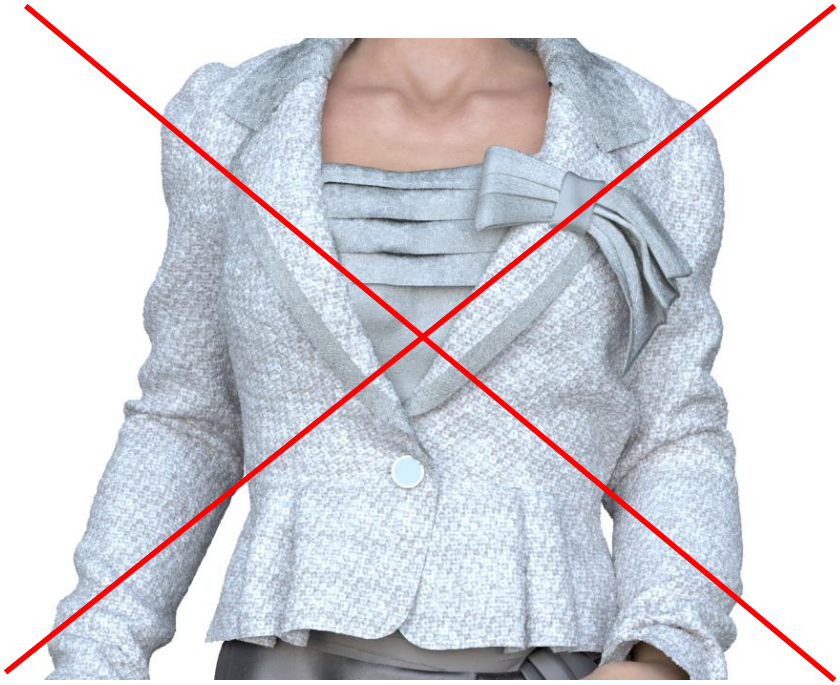
- Apply the lighting model at a set of points across the entire surface
  - Determine where the lights come from
  - Figure out which pixels to fill
  - Paint the color for each pixel



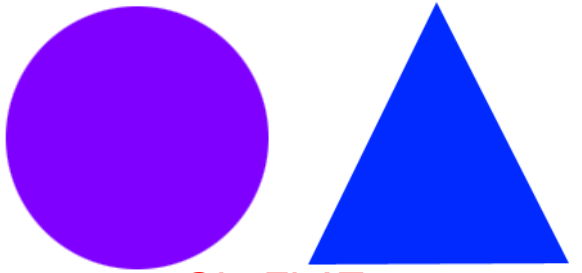



# Shading with OpenGL

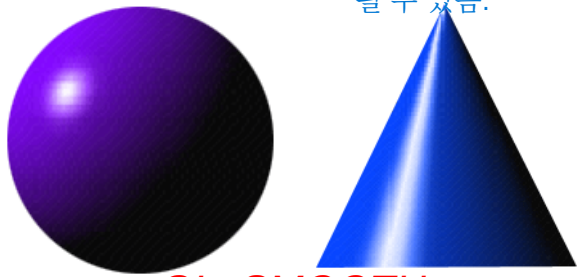
- OpenGL is focused on real-time applications.
- OpenGL provides very basic shading models.



Polygon을 1개의 색상으로 shading



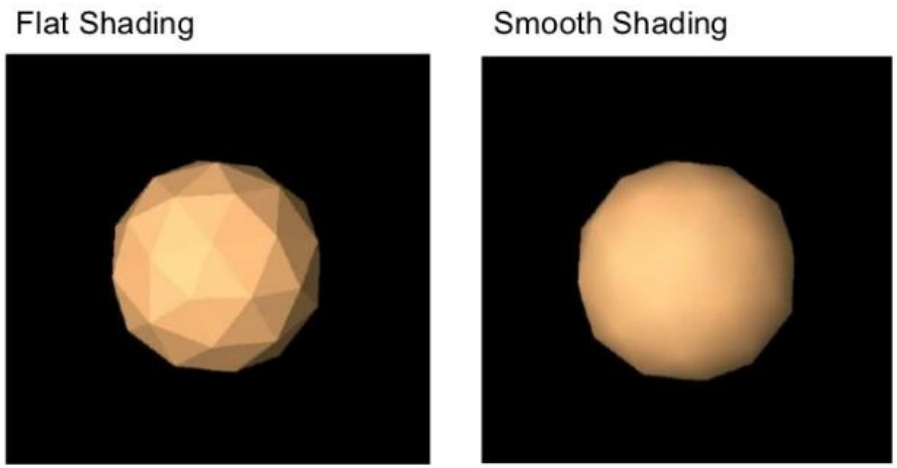
*GL_FLAT* 다각형 하나 같은 색으로. 물체 구성하는 작은 단위가 될 수 있음.

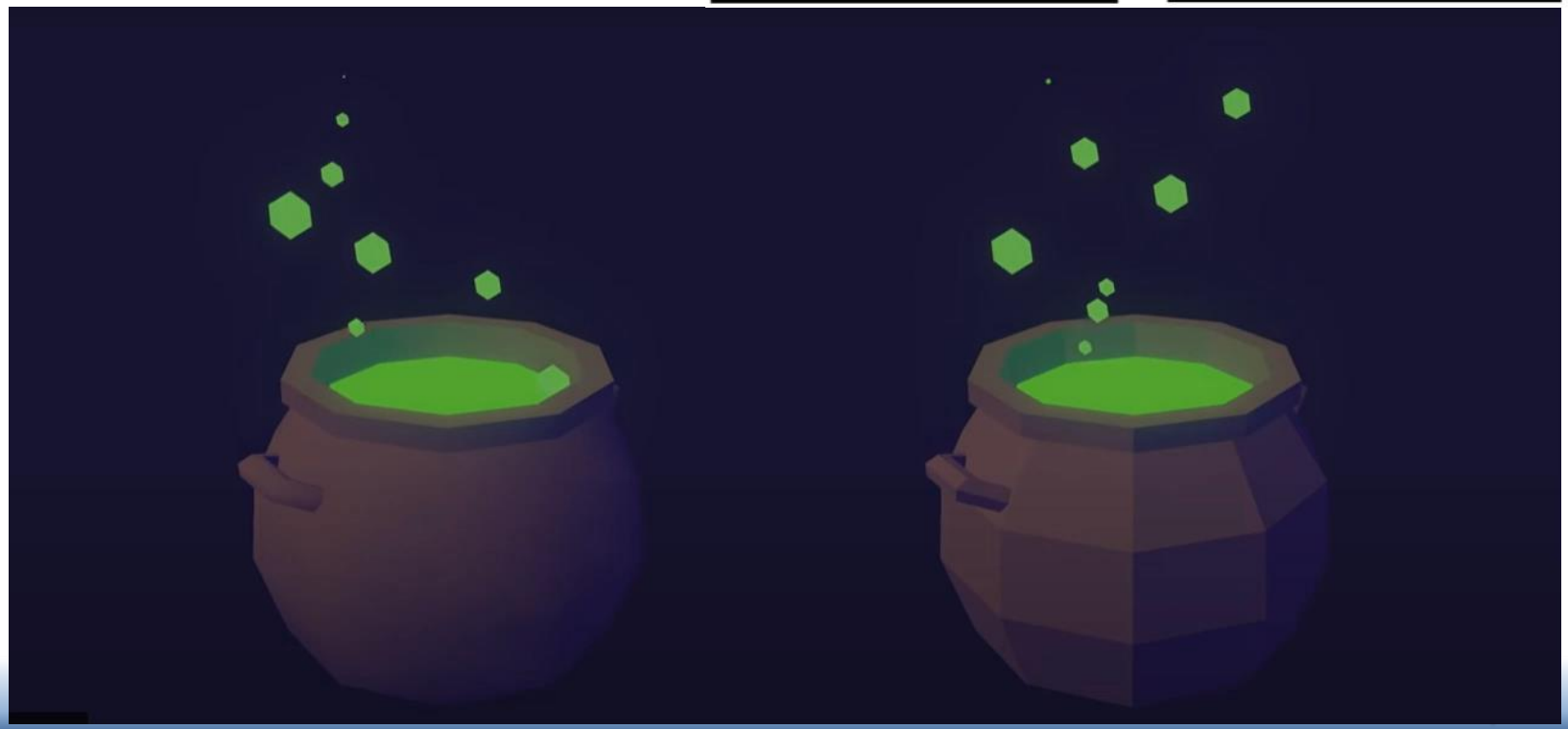


*GL_SMOOTH* 그라데이션. 도형 모양이 잘 안 보이게 됨.

# Flat Shading vs. Smooth Shading

- **GL_FLAT** : Flat shading
  - One color per polygon
- **GL_SMOOTH** : Smooth
  - smoothing : color interpolation

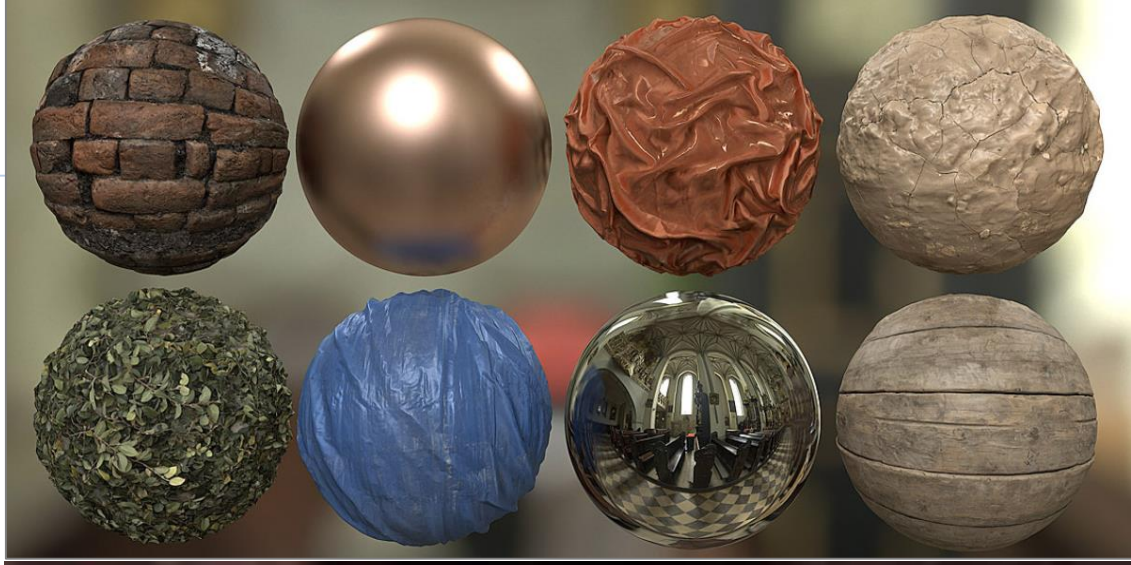


# Shading

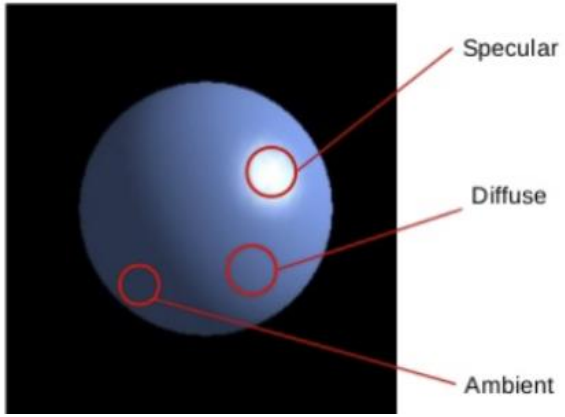
빛과 오브젝트 재질에 따라 다른 섀이딩을 해야함.

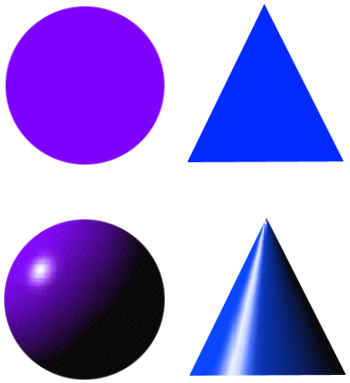
- Light Source
- Object Material
- Shading Model
  - Flat shading
  - Smooth shading

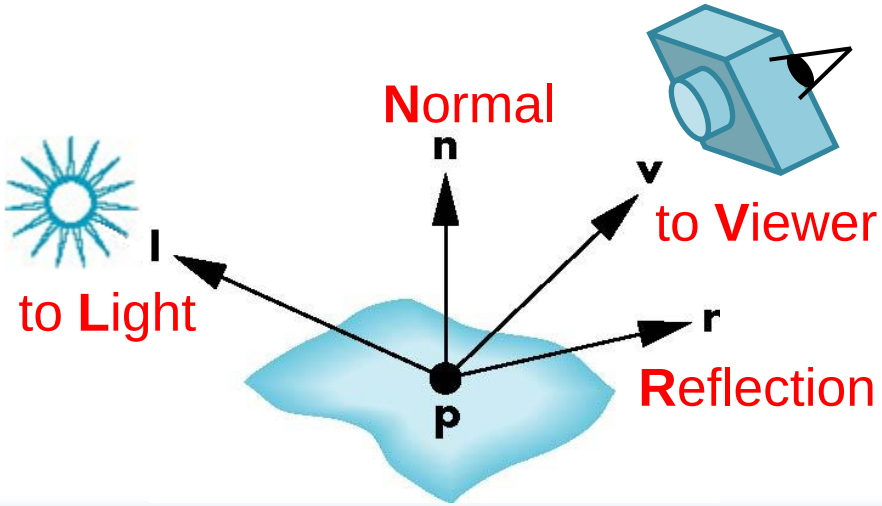


# How to Shade in OpenGL?

1. **Set the light sources**
   - Position
   - Type: directional light, point light, spotlight
   - Other Properties: ambient, diffuse, specular
2. **Set the materials**
   - Properties: ambient, diffuse, specular, emission, shininess
3. **Set the shading method**
   - Flat & Smoothing





# How to Shade in OpenGL?

1. light
2. material

위치, 종류

1. **Light Source : Position & Type**
2. Illumination Model : Light & Material

   Ambient, Diffuse, Specular

# Light Position

# Light *position* in 3차원

- Sets the values of individual light source(LIGHT0) parameters
  - **glLightfv( GL_LIGHT0, GL_POSITION, GLfloat pos[] );**

몇 번째 라이트의 (속성 – 값) 설정하는 함수

float

vector

어떤 라이트의 성질과 특성값을 매개변수로 제공.

glLightfv(light, property, value)

f: 파라미터가 실수값
v: 값을 벡터로 주어야한다. (배열)

라이트 번호
GL_LIGHT0~7

OpenGL Light 8개까지 제공

gl의 상수constant로 선언되어 있음.

**특성값**
여기서는 라이트 위치값
(x, y, z)

**라이트의 어떤 특성을 설정할 것인지?**
여기서는 라이트 위치
GL_POSITION

포지션 성질을 선언. / position말고도 다른 성질 올 수 있음.
이것도 상수constant

# Light *position*

glLightfv(..)
glEnable(GL_LIGHTING)
glEnable(GL_LIGHT0)


Before

```
int main()
{
        glViewport(0, 0, w, h);
배경 색 glClearColor(0.0, 0.0, 0.0, 0.0);
        glEnable(GL_DEPTH_TEST);
}
```

```
int main()
{
        glViewport(0, 0, w, h);
        glClearColor(0.0, 0.0, 0.0, 0.0);
        glEnable(GL_DEPTH_TEST);

                                  x    y    z
        GLfloat position[] = { 0.0, 0.0, 2.0, 0.0 };
        glLightfv(GL_LIGHT0, GL_POSITION, position);
        glEnable(GL_LIGHTING);
        glEnable(GL_LIGHT0);
}
```

나중에 사용하려고 남긴 값.

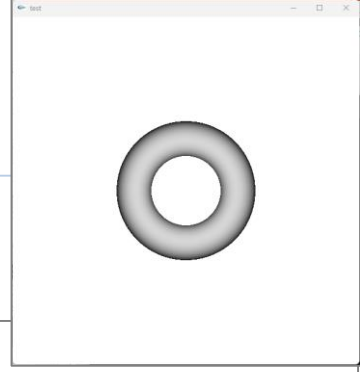
전체 라이트 꺼지는 것

라이트구현

0번째라이트만 관여

# Light *position*

- 조명 위치 설정 : glLightfv(GL_LIGHT0, GL_POSITION, position)



```
#include <iostream>
#include <gl/glut.h>
using namespace std;

#define WIDTH 600
#define HEIGHT 600

void display()
{
    GLfloat position[] = { 0, 0, 2, 0 };

    glClear(GL_COLOR_BUFFER_BIT
        | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-1, 1, -1, 1, 1, 30);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0, 0, 2, 0, 0, 0, 0, 1, 0);
    // Light Position
    glLightfv(GL_LIGHT0, GL_POSITION, position);

    glutSolidTorus(0.1, 0.3, 100, 100);
    glFlush();
}
```

0,0,2 위치에 라이트

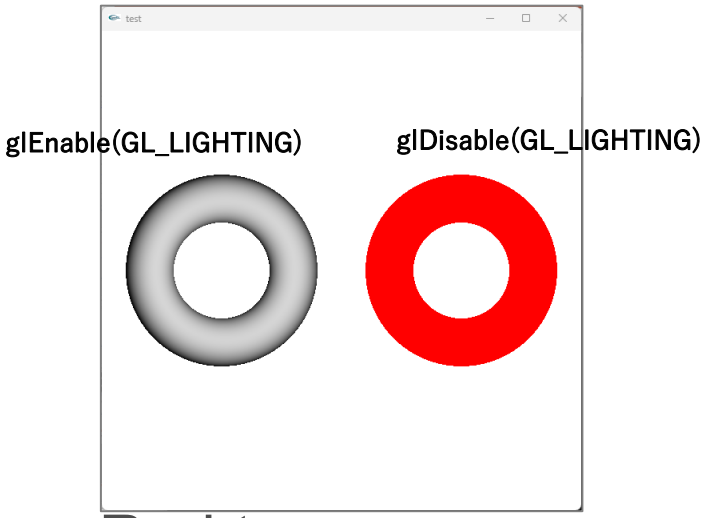
```
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGBA
                        | GLUT_DEPTH
                        | GLUT_SINGLE);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(WIDTH, HEIGHT);
    glutCreateWindow("test");

    glClearColor(1, 1, 1, 0);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```

개별라이트 활성화와
전체 라이트 활성화는
둘다 있어야 제 기능을 함.

XYZ 기준축을 그리고,
조명, 카메라, 토러스를 그려보세요.

# Light *position*

- 조명 위치 설정
  - glLightfv(GL_LIGHT0, GL_POSITION, position)
- 조명 활성화/비활성화
  - glEnable(GL_LIGHTING)
  - glDisable(GL_LIGHTING)




Red torus
White light (default)

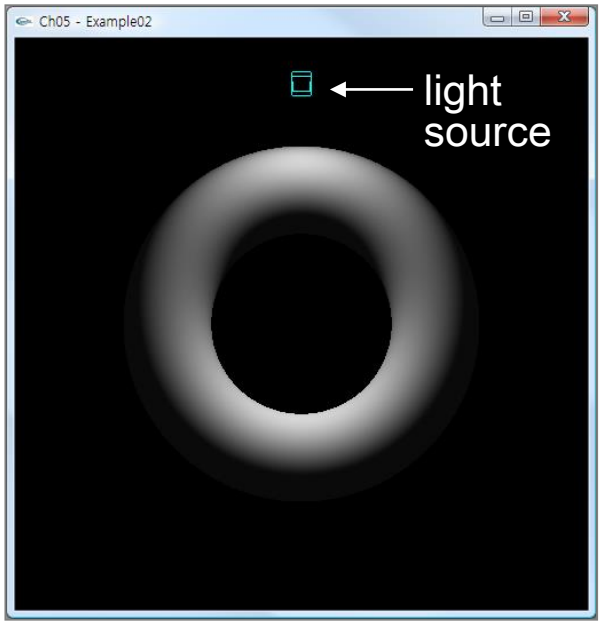
조명받는 왼쪽 토러스
조명없는 오른쪽 토러스

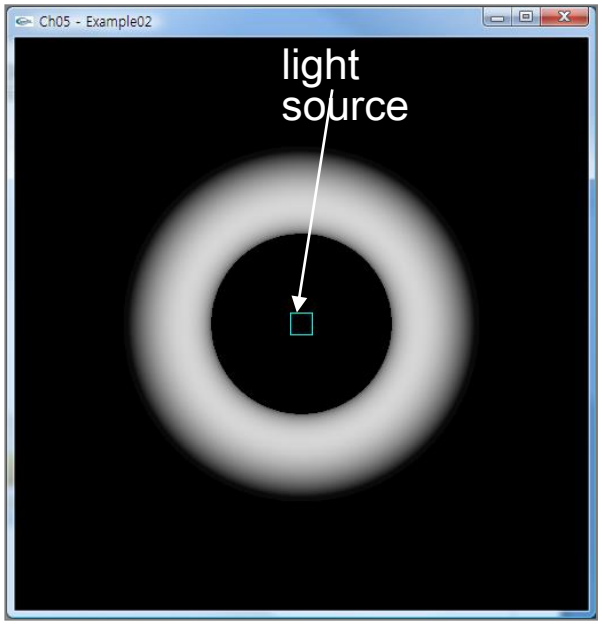
```
#include <iostream>
#include <gl/glut.h>
using namespace std;

#define WIDTH 600
#define HEIGHT 600

void display()
{
    GLfloat position[] = { 0, 0, 2, 0 };

    glClear(GL_COLOR_BUFFER_BIT| GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-1, 1, -1, 1, 1, 30);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0, 0, 2, 0, 0, 0, 0, 1, 0); // camera
    glLightfv(GL_LIGHT0, GL_POSITION, position);

    glTranslatef(-0.5, 0, 0);
    glColor3f(1, 0, 0);
    glutSolidTorus(0.1, 0.3, 100, 100);      // Left Torus

    glDisable(GL_LIGHTING);
    glTranslatef(1.0, 0, 0);
    glColor3f(1, 0, 0);
    glutSolidTorus(0.1, 0.3, 100, 100);      // Right Torus
    glEnable(GL_LIGHTING);

    glFlush();
}
```

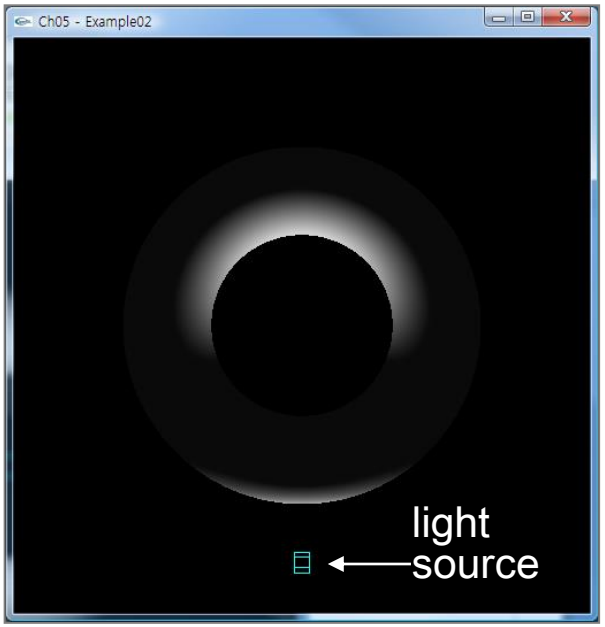
enable할때까지 라이트 영향 안받음.
enable 전까지 조명 영향 없음.
인에이블 후로는 영향 O

# Light *position*

- Light source의 position을 바꾸면 어떤 현상이 일어날까 ?
- Solid torus 주위에서 Light source를 회전시키면서 solid torus가 어떻게 shading 되는지 관찰해 보자.
  - *Light source* **의 다양한 위치** *(position)* **테스트**

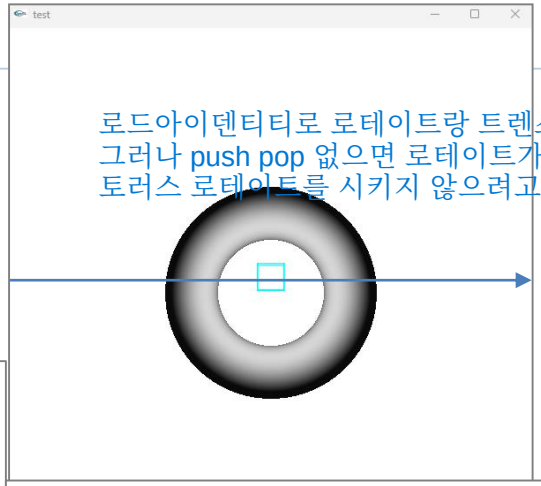


# Light *position : rotation*

XYZ 기준축을 그리고,
조명, 카메라, 토러스를 그려보세요.

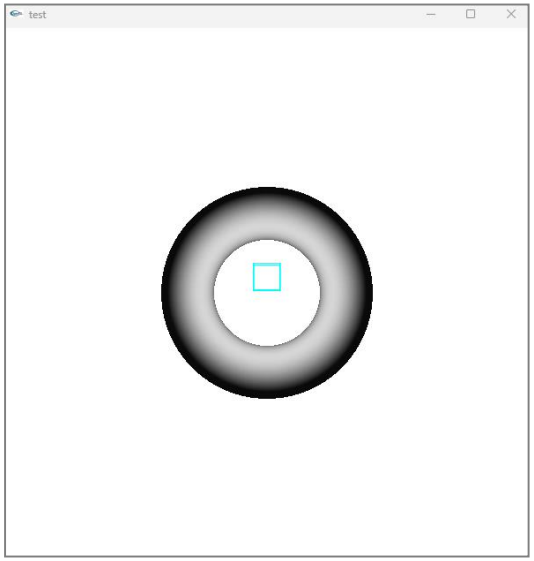



로드아이덴티티로 로테이트랑 트렌스레이트는 누적 X
그러나 push pop 없으면 로테이트가 적용됨.
토러스 로테이트를 시키지 않으려고 조명 부분에만 push pop 한것.

```
#include <iostream>
#include <gl/glut.h>
using namespace std;

#define WIDTH 600
#define HEIGHT 600

float spin = 0;

void display()
{
    GLfloat position[] = { 0, 0, 1, 0 };

    // Orthographic View Volume
    glClear(GL_COLOR_BUFFER_BIT
            | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity(); 누적 방지.
    glOrtho(-1, 1, -1, 1, 1, 30);

    // Camera Position, Direction, Up vector
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0, 0, 3, 0, 0, 0, 0, 1, 0); //camera

    // LIGHT Rotation
    glPushMatrix();
    glRotatef(spin, 1.0, 0.0, 0.0);
    spin += 0.3;        // 회전 속도 조절
    glLightfv(GL_LIGHT0, GL_POSITION, position);
                위치는 시키지만 눈에 보이지 않음. 라이트 ㅇㅇ
    // Skyblue LIGHT Cube Display
    glTranslatef(position[0], position[1], position[2]);
                                                0,0,1
    glDisable(GL_LIGHTING); 라이트 영향 off
    glColor3f(0, 1, 1); 조명과 같은 위치에 skyblue
    glutWireCube(0.1); 큐브를 그려서 조명의 위치 확인함.
    glEnable(GL_LIGHTING); 라이트 영향 on
    glPopMatrix();

    // Draw Solid Torus
    glColor3f(1, 0, 1);
    glutSolidTorus(0.1, 0.3, 100, 100);

    glFlush();
}
```

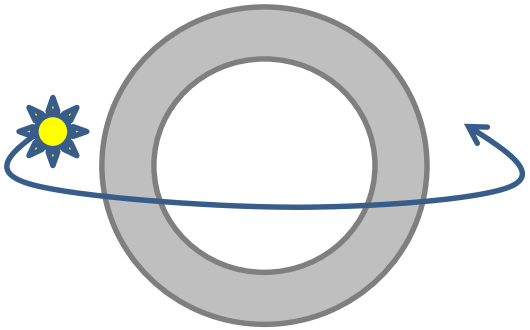
# Light *position : rotation*

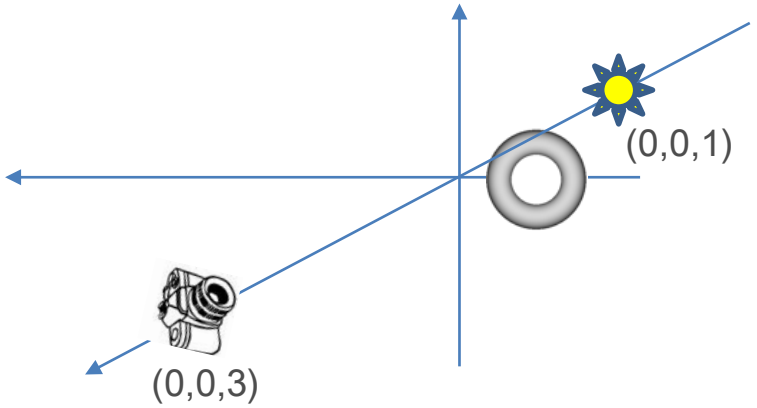
```
int main(int argc, char** argv)
{
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_RGBA | GLUT_DEPTH
		| GLUT_SINGLE);
	glutInitWindowPosition(100, 100);
	glutInitWindowSize(WIDTH, HEIGHT);
	glutCreateWindow("test");

	glClearColor(1, 1, 1, 0);
	glEnable(GL_DEPTH_TEST);
	glEnable(GL_LIGHTING);
	glEnable(GL_LIGHT0);

	glutIdleFunc(display);
	glutDisplayFunc(display);
	glutMainLoop();
	return 0;
}
```



# Quiz1. Light position

- 조명이 다음과 같이 가로로 회전하는 코드 실습
- 물체와 조명의 상대적인 위치 변경하기
  - 물체 : (0.3, 0, 0)
  - 조명 : (0, 0, -1)

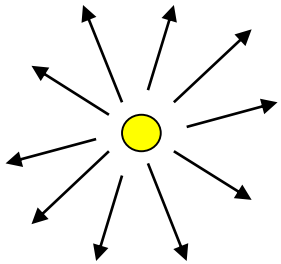




# Light Type

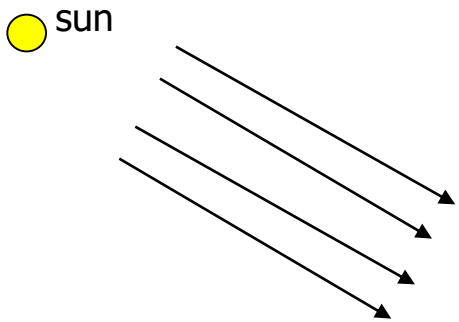
# Light Source Model

- Simple mathematical models
  - Point light (점 광원) : 실내 조명
  - Directional light (방향성 광원) : sun
  - Spotlight (집중광, 스포트라이트)

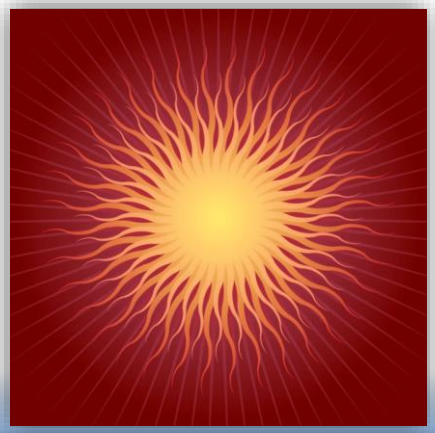


**Point light**

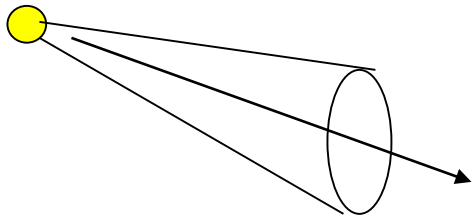
사방팔방 비추는
실내등. 위치 중요




**Directional light**



태양.
너무멀리 있어서
위치는 무의미. 방향만

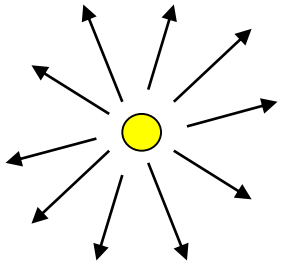


**Spotlight**

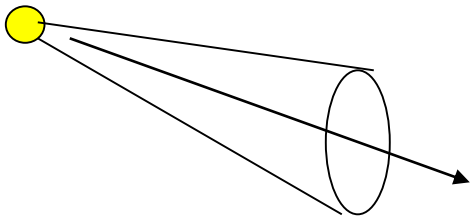
어떤위치
어떤방향 중요

# Light Source Model

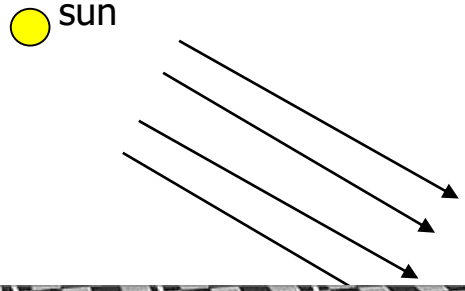
**Point light**
점 광원
실내 조명

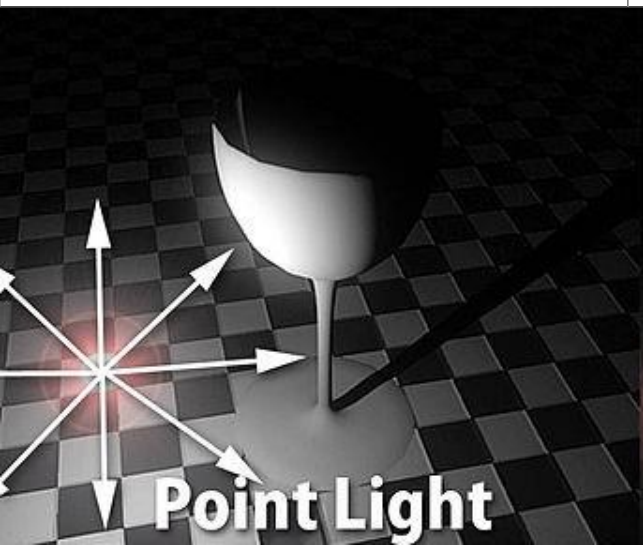


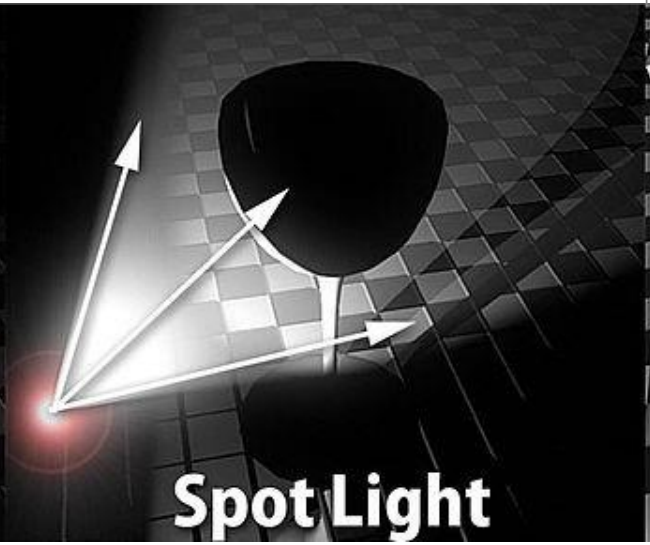
**Spotlight**
집중광

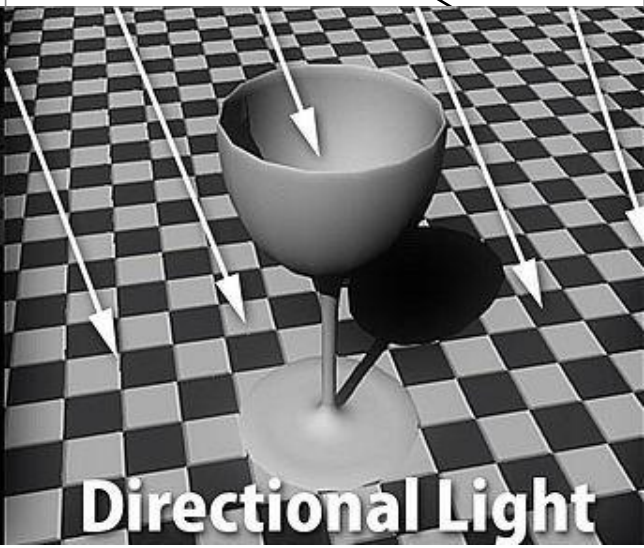


**Directional light**
방향성 광원
sun



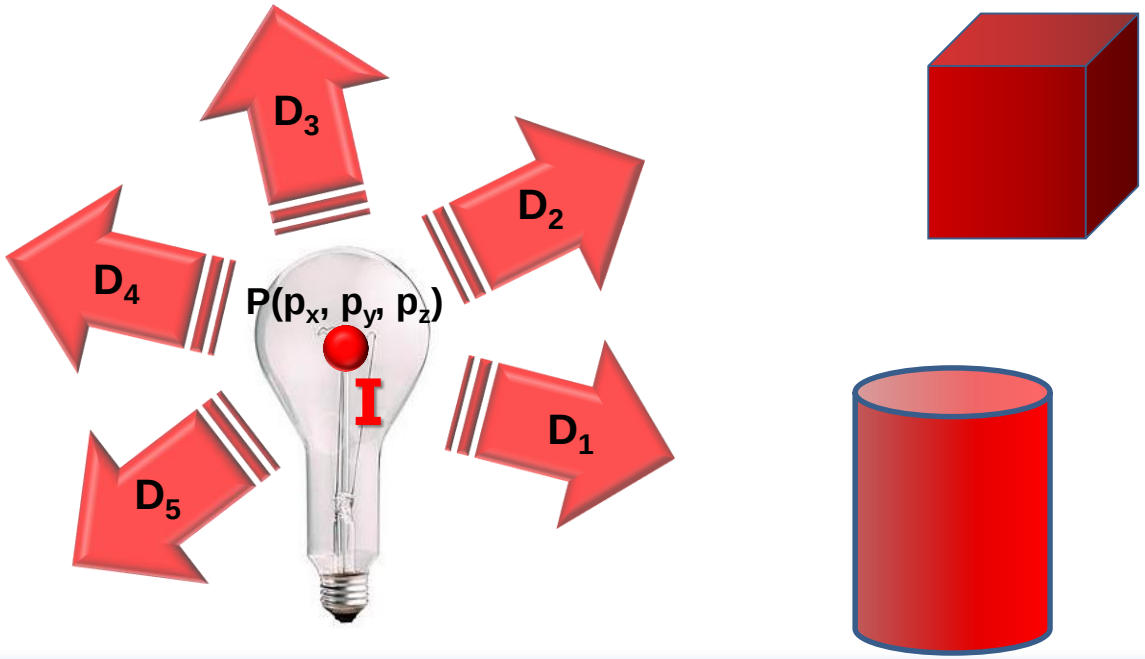
# Point Light Source(점 광원)

- Omnidirectional point source
  - Intensity I : 조명이 얼마나 밝은가? 속성 2개
  - Position P($p_x$, $p_y$, $p_z$) : 조명의 위치

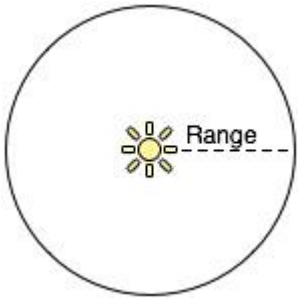
```
GLfloat light_position[] = {0.0, 0.0, 1.0, 1.0};
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```

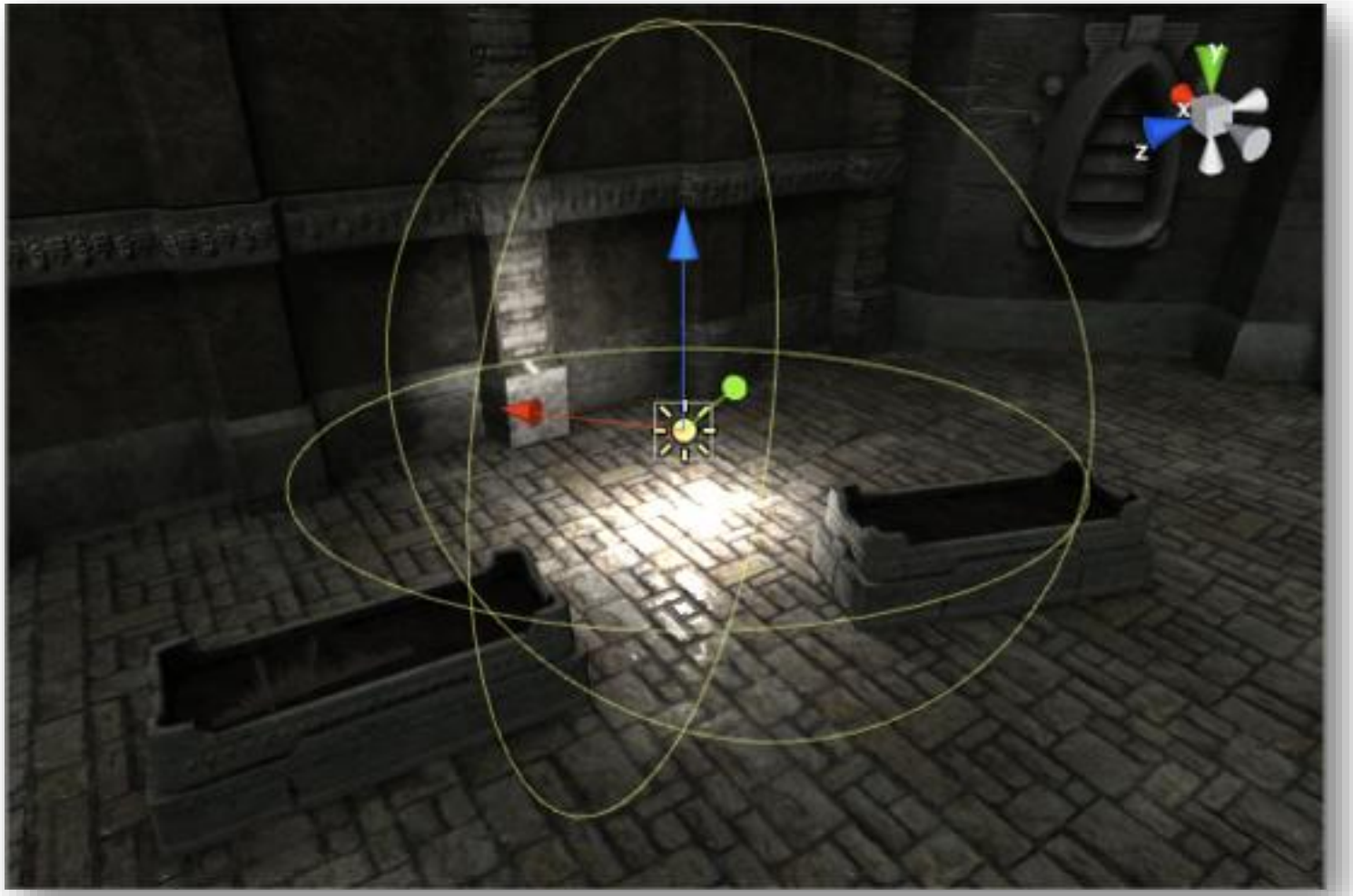
여기 맨끝이 1이면 위치야! 라는걸 의미
이자리가 0이면 방향

이 함수로 조명위치 설정 가능.



# Point Light Example



# Directional Light Source(방향성 광원)

- Directional Light Source
  - Point light source at infinity

속성 2개

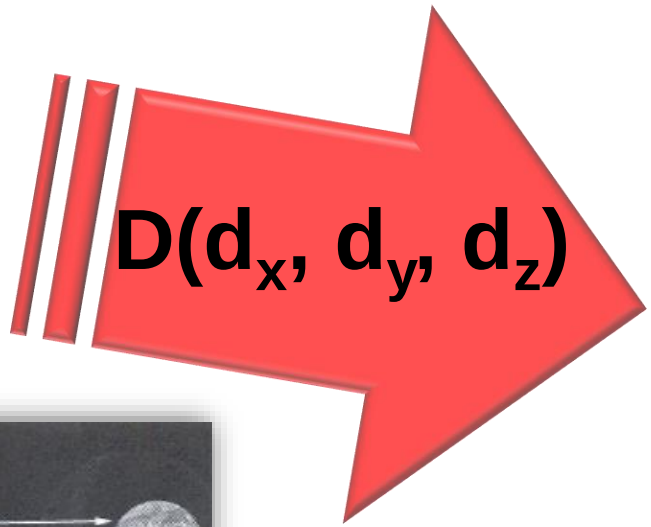
  - Intensity I : 조명이 얼마나 밝은가
  - Direction D($d_x$, $d_y$, $d_z$) : 조명의 방향

cf) 조명 위치는 의미없음. 너무 멀어서

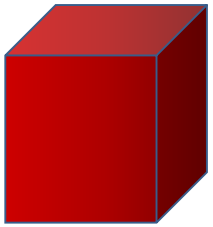
0,0,1의 방향에서 태양이 비춰오는 것.

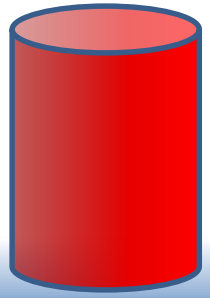
```
GLfloat light_position[] = {0.0, 0.0, 1.0, 0.0};
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```

I

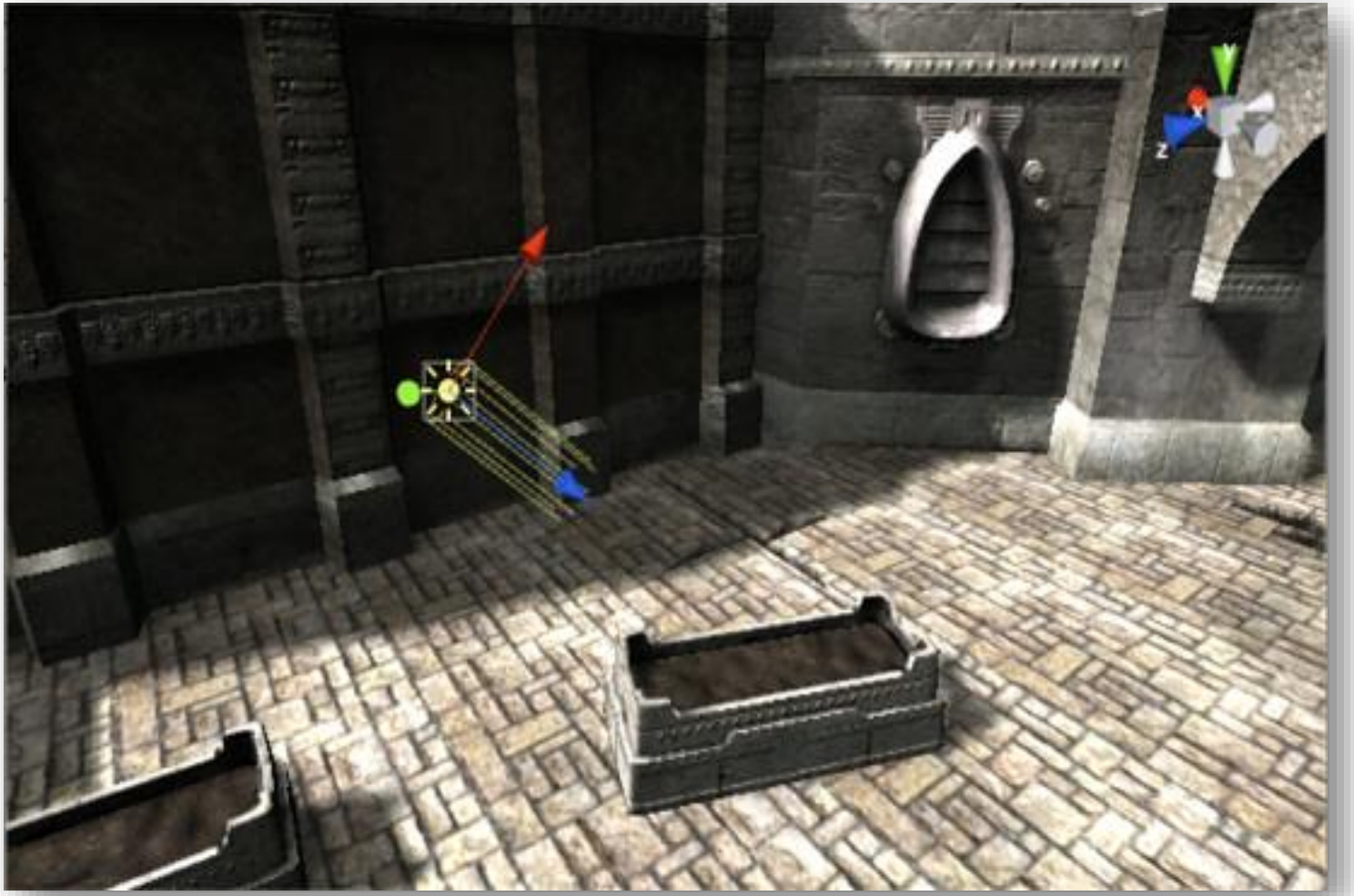


D($d_x$, $d_y$, $d_z$)





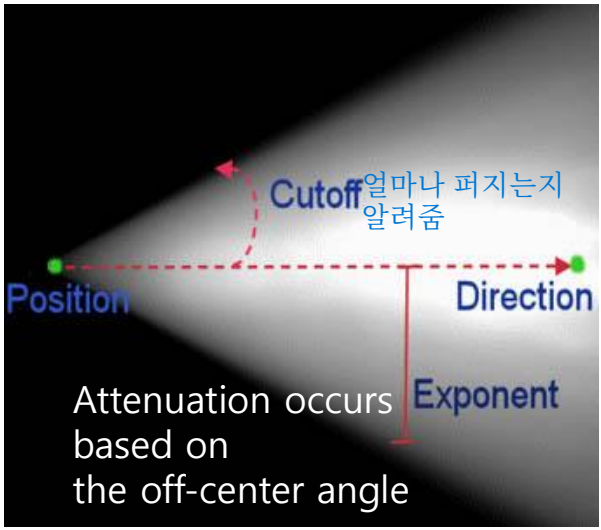
# Directional Light Example

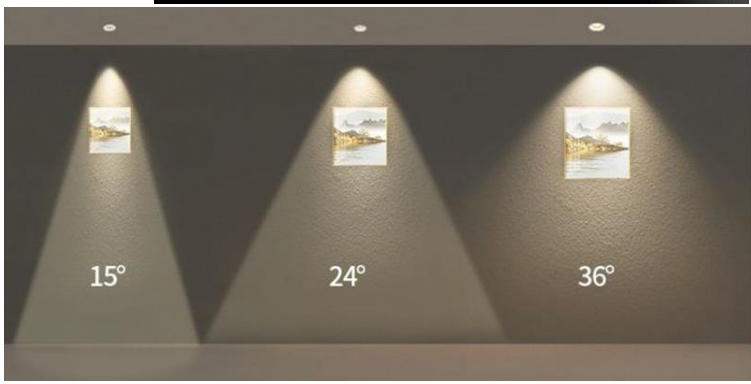


# Spotlight Source(집중광)

- Point light source with direction
  - Intensity I : 조명의 밝기
  - Position P($p_x$, $p_y$, $p_z$) : 조명의 위치
  - Direction D($d_x$, $d_y$, $d_z$) : 조명의 방향
  - cutoff : spotlight 범위 (angle) 센터에서 끝까지의 각도.(반각)
  - exponent : 조명 감쇠 효과 <0이면 범위 내의 빛의 세기 모두 동일
    (값이 클수록 감쇠효과 커짐)

속성 5개



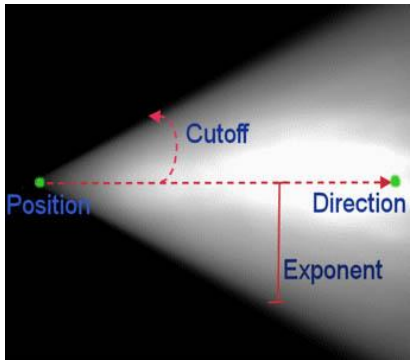




cutoff angle

# OpenGL Light Sources: *types*

몇 번째 라이트의 (속성–값) 설정하는 함수

glLightfv(light, property, value)




## Spotlight (집중 광원)

- **GL_POSITION** lightfv
  - GLfloat light_position[] = {0.0, 0.0, 1.0, **1.0**};
  - glLightfv(GL_LIGHT0, GL_POSITION, light_position);

위치 or 방향

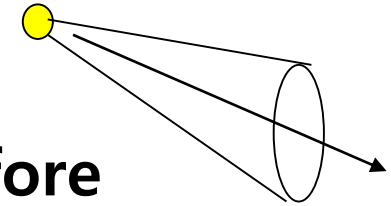
- **GL_SPOT_DIRECTION**
  - By default, the negative z axis (0,0,-1)

- **GL_SPOT_CUTOFF** 값 하나. 배열 X/ ligjtf
  - the angle between the cone's axis and the cone's edge.
  - It ranges from 0 to 90 degrees.

128이면 감쇠효과 굉장히 큼.

- **GL_SPOT_EXPONENT** ranges from 0 to 128. 값 하나. 배열 X
  - By default, its value is 0, which means there is no light attenuation as the spotlight spreads toward the side edges of the cone.
  - In other words, the cone of light is evenly distributed.

# OpenGL Light Sources: *types*



## Spotlight (집중 광원)

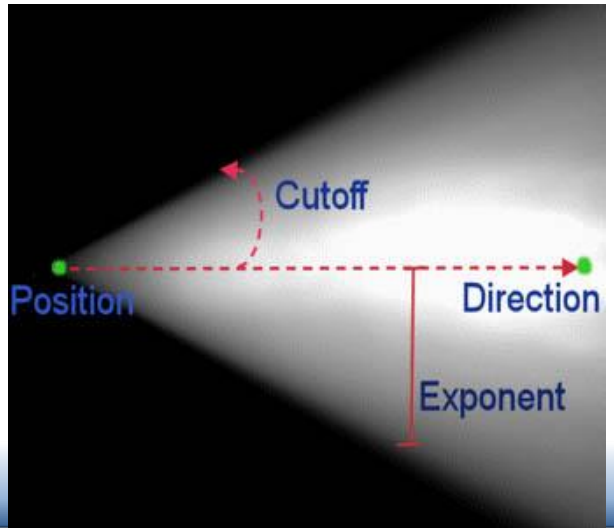
- **Create point source as before**

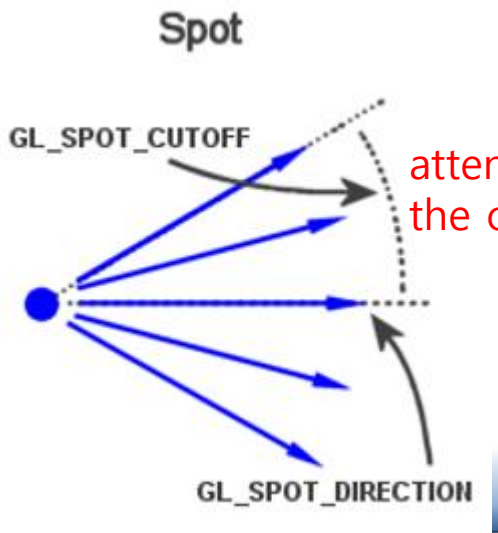
```
GLfloat light_position[] = {0.0, 0.0, 1.0, 1.0};
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```

- **Spot light source**

```
GLfloat sd[] = {-1.0, -1.0, 0.0};
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, sd);
glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 45.0);
glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, 2.0);
```

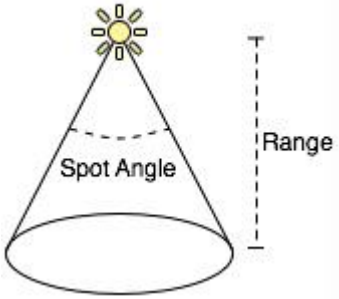
v 없음에 유의

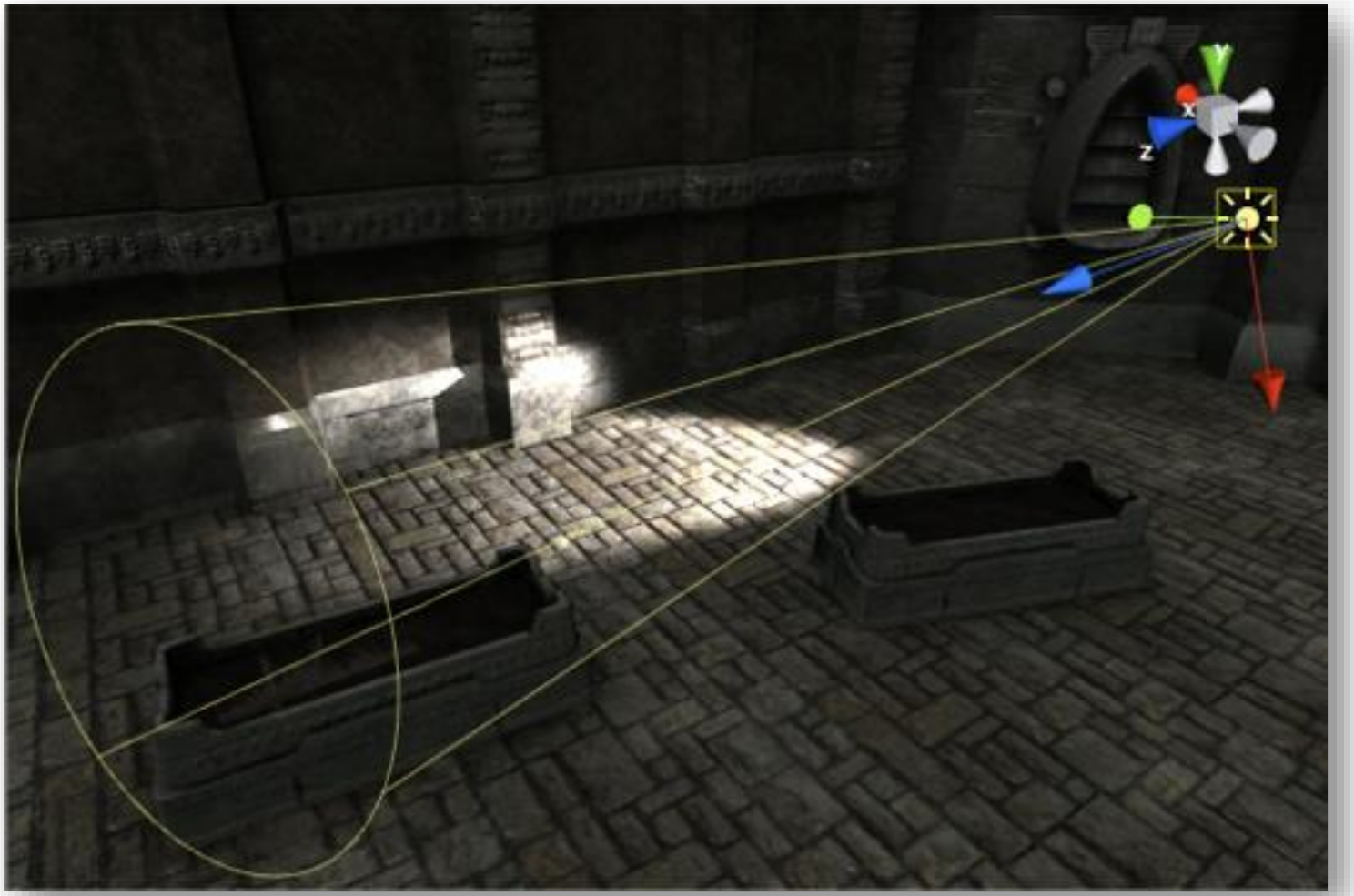



attenuation occurs based on the off-center angle

# Spotlight Example



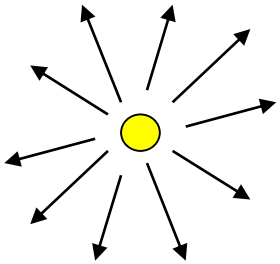
# Spotlight vs. Point Light

**Point light**

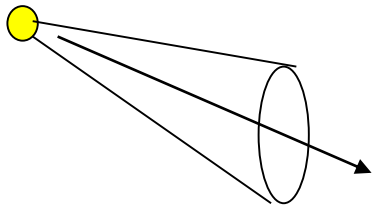
**= Special version of spotlight**

- light position (o)
- light direction (x)
- cutoff : spotlight 범위 (angle >= 90)
- exponent : 조명 감쇠 효과(exponent = 0)

GLfloat light_position[] = {0.0, 0.0, 1.0, ____1____**?** };



**Point Light**



**Spotlight**

# Quiz2. Light Types

- Point Light, Directional Light, Spotlight 중에 방향성 광원을 모두 찾아서 쓰세요.

  directional light, spotlight

- Point Light에서 중요한 2가지 속성은?

  intensity, position

- Directional Light에서 중요한 2가지 속성은?

  intensity, direction

- Spotlight에서 중요한 5가지 속성은?

  intensity, position, direction, cutoff, exponent

# Quiz3. OpenGL Light Types

- Point Light의 위치를 설정하는 코드를 작성하세요.

```
GLfloat_light_position[] = {0,.0, 0.0, 1.0, 1.0};
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```

- Directional Light의 방향을 설정하는 코드를 작성하세요.

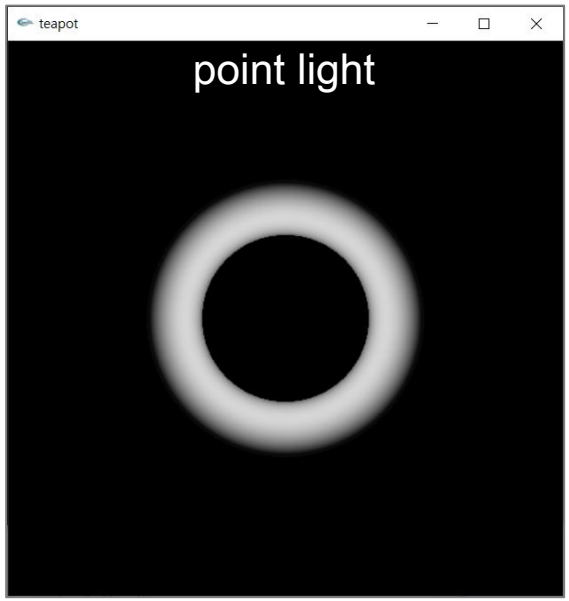
```
GLfloat light_position[] = {0.0, 0.0, 1.0, 0.0};
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```

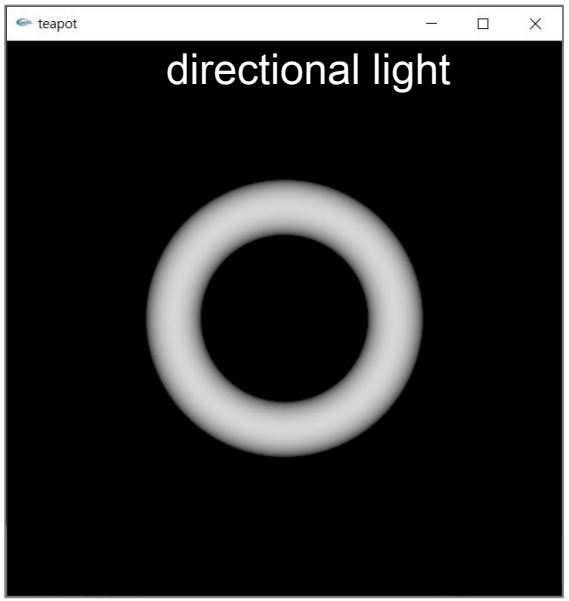
- Spotlight에서 중요한 속성 5가지를 설정하는 코드를 작성하세요.

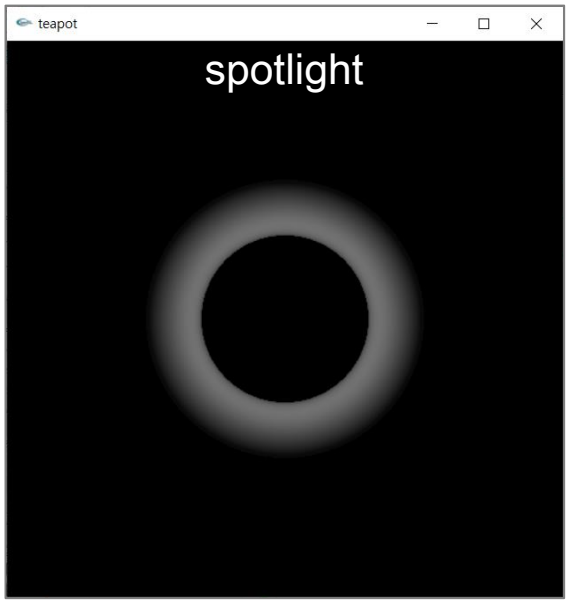
```
GLfloat light_position[] = {0.0, 0.0, 1.0, 1.0};
glLightfv(GL_LIGHT0, GL_POSITION, light_position);

GLfloat sd[] = {-1.0, -1.0, 0.0};
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, sd);
glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 각도float값_예: 45.0);
glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, 감쇠효과를위한 float값_예: 2.0);

    +
```

# OpenGL Light Types: *practice*

- 키보드 1,2,3을 누를 때 다음의 각 light source가 solid torus에 어떤 영향을 미치는지 확인해 보자.
  - *Point light source*
  - *Directional light source*
  - *Spotlight*

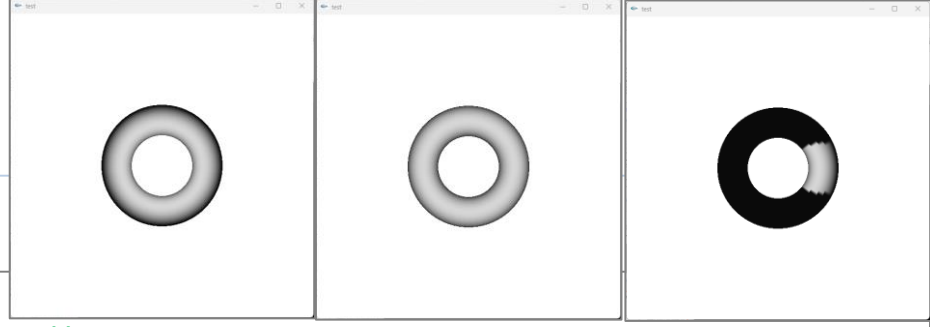


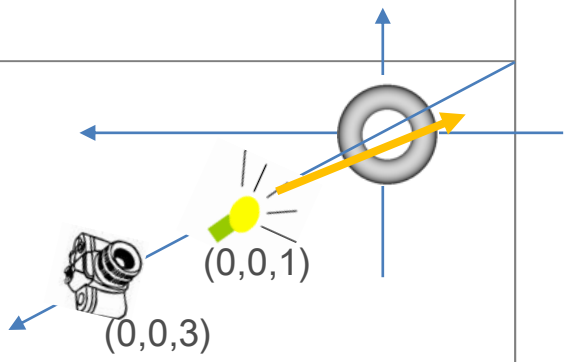



# OpenGL Light Types

- Three light types
  - Point Light
  - Directional Light
  - Spotlight






```
#include <iostream>
#include <gl/glut.h>
using namespace std;

#define WIDTH 600
#define HEIGHT 600
float spin = 0;
int ltype = 1;

void display()
{
	GLfloat position[] = { 0, 0, 1, 0 };

	// Orthographic View Volume
	glClear(GL_COLOR_BUFFER_BIT
		| GL_DEPTH_BUFFER_BIT);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	glOrtho(-1, 1, -1, 1, 1, 30);

	// Camera Position, Direction, Up vector
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
	gluLookAt(0, 0, 3, 0, 0, 0, 0, 1, 0);

	// LIGHT
	switch (ltype) {
	case 1: // Point Light   맨 왼쪽거
		position[3] = 1;
		glLightfv(GL_LIGHT0, GL_POSITION, position);
		glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 90);
		glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, 0.0);
		break;
	case 2: // Directional Light   가운데
		position[3] = 0;
		glLightfv(GL_LIGHT0, GL_POSITION, position);
		break;
	case 3: // Spotlight   오른쪽
		position[3] = 1;
		glLightfv(GL_LIGHT0, GL_POSITION, position);
		GLfloat sd[] = { 0.3, 0.0, -1.0 };
		glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, sd);
		glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 10);
		glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, 0);
	break;
	}

	// Draw Solid Torus
	glColor3f(1, 0, 1);
	glutSolidTorus(0.1, 0.3, 100, 100);

	glFlush();
}
```

# OpenGL Light Types

- Three light types
  - Point Light
  - Directional Light
  - Spotlight
- Keyboard callback function
  - 1: Point light
  - 2: Directional light
  - 3: Spotlight

  * glutPostRedisplay() 함수 작성 잊지말 것!!

```
void keyboard(unsigned char key, int x, int y)
{
	switch (key) {
	case '1': ltype = 1; break;
	case '2': ltype = 2;  break;
	case '3': ltype = 3;  break;
	default: break;
	}
	glutPostRedisplay();
}

int main(int argc, char** argv)
{
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_RGBA | GLUT_DEPTH
		| GLUT_SINGLE);
	glutInitWindowPosition(100, 100);
	glutInitWindowSize(WIDTH, HEIGHT);
	glutCreateWindow("test");

	glClearColor(1, 1, 1, 0);
	glEnable(GL_DEPTH_TEST);
	glEnable(GL_LIGHTING);
	glEnable(GL_LIGHT0);

	glutKeyboardFunc(keyboard);
	glutDisplayFunc(display);
	glutMainLoop();
	return 0;
}
```

# 과제

강의 슬라이드를 참조하여 Solid Torus를 (0,0,0) 위치에 그리고 Light의 위치와 종류를 설정하는 코드를 구현하세요. (강의 슬라이드에 있는 대로 거의 그대로 구현하면 됩니다.)
그리고, 다음 기능을 가지도록 확장하세요.

**(1) Light의 위치 조절**
- key '+'를 누를 때마다 Light의 z 위치를 + 방향으로 일정 정도 증가시킨다.
- key '-'를 누를 때마다 Light의 z 위치를 - 방향으로 일정 정도 증가시킨다.

**(2) Light의 Y축 기준 회전**
- key 's'를 누르면 회전을 멈추고, 다시 's'를 누르면 회전을 하도록 구현한다. 회전 방향은 자유롭게.
- 회전을 멈출 때는 조명이 기본 위치 (0,0,1)에 위치하도록 초기화한다.
- 회전하고 있는 조명을 'red cube'로 표시한다.

**(3) Light의 Type 변경**
- key '1'은 point light source, key '2'는 directional light source, key '3'은 'spot light'로 설정한다.
- key 'a'와 'b'로 spotlight angle을 증가/감소시킨다.
- key 'c'와 'd'로 spotlight의 attenuation을 증가/감소시킨다.

**흥미로운 숙제 재미있게 해보아요 !**
**Good Luck~**

Lighting & Shading

**Thank you~**