

Lab05. Illumination with Normal Vector

2 Spot light

• 무대조명 : spotlight 예

```
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *

import math

Ld = [1,1,1,1]
Ls = [1,1,1,1]
Lp = [1,1,5,1]
Md = [1,0,0,1]
Ms = [1,1,1,1]
shininess = [127.0]

t = 0

def LightSet():
    glLightfv(GL_LIGHT0, GL_DIFFUSE, Ld)
    glLightfv(GL_LIGHT0, GL_SPECULAR, Ls)
    #spot light setting
    glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 30.0)
    glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, [0,0,-1])
    glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, 3.0)
    #

    glMaterialfv(GL_FRONT, GL_DIFFUSE, Md)
    glMaterialfv(GL_FRONT, GL_SPECULAR, Ms)
    glMaterialfv(GL_FRONT, GL_SHININESS, shininess)

def LightPosition(t):
    Lp[0] = 3.0*math.sin(t)
    Lp[1] = 3.0*math.cos(t)
    glLightfv(GL_LIGHT0, GL_POSITION, Lp)

def glInit() :
    glClearColor(0,0,0.5,1)
    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
    glEnable(GL_DEPTH_TEST)
    LightSet()
```

```
def disp() :
    global t
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    # CAMERA setting
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    #gluPerspective(60, 1.0, 0.1, 1000)
    glOrtho(-6,6,-6,6,-100,100)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
    gluLookAt(0,0,10, 0,0,0, 0,1,0)
    #gluLookAt(0,10,0, 0,0,0, 0,0,1)

    t+= 0.01
    LightPosition(t)
    # OBJECTS
    for x in range(-10, 11) :
        for y in range(-10, 11) :
            glPushMatrix()
            glTranslatef(x/2.0, y/2.0, 0)
            #glutSolidSphere(0.5, 20,20)
            glutSolidSphere(0.2, 20,20)
            glPopMatrix()

    glFlush()

def main():
    # windowing
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE | GLUT_DEPTH | GLUT_RGB)
    glutInitWindowSize(512,512)
    glutInitWindowPosition(50,50)
    glutCreateWindow(b'Spot Lights')

    # initialization
    glInit()

    # register callbacks
    glutDisplayFunc(disp)
    glutIdleFunc(disp)

    # enter main loop
    glutMainLoop()

if __name__ == '__main__' :
    main()
```

구생성

3 실습문제1

- 조명의 색깔을 변경해 보자.
- 조명의 반경을 줄여보자.
- 무대조명을 2개 더 설치해서 각각 움직여보자.
- 일반 조명 하나를 설치하여 0.1초 정도 켜졌다 꺼지는 것을 반복하는 장면을 구현해 보자.

4 법선벡터와 조명

- 두 개의 삼각형에 조명의 위치를 변화시켜 법선벡터와의 관계에 따라 shading이 어떻게 변하는지 보여준다.

```
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
import sys
import math
import numpy as np

# Light Properties
Ld = [1, 1, 1, 1]
Ls = [1, 1, 1, 1]
# Material Properties
Md = [1, 1, 0, 1]
Ms = [1, 1, 1, 1]
shininess = [120]

t = 0

# initialization
def GLinit():

    glClearColor(0, 0, 1, 0)
    glEnable(GL_COLOR_MATERIAL)
    glEnable(GL_DEPTH_TEST)
    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
    glLightfv(GL_LIGHT0, GL_DIFFUSE, Ld)
    glLightfv(GL_LIGHT0, GL_SPECULAR, Ls)
    glMaterialfv(GL_FRONT, GL_DIFFUSE, Md)
    glMaterialfv(GL_FRONT, GL_SPECULAR, Ms)
    glMaterialfv(GL_FRONT, GL_SHININESS, shininess)
```

배경색의 설정, 깊이버퍼에 의한 검사를 사용하고 확산광, 경면광을 설정.
객체의 확산 및 경면 반사에 대한 색상을 설정.

5 법선벡터와 조명(2)

두개의 삼각형을 그리는데 주어진 점으로 벡터를 구하고 그 벡터로 외적($\text{np.cross}()$)을 구함. 이 외적의 값을 크기로 나누어 ($\text{normal} = \text{normal} / (\text{np.linalg.norm}(\text{normal}))$) 법선벡터를 구함.

```
# display callback
def display() :
    global t

    glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT)

    # CAMERA SETTING
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(60.0, 1.0, 0.1, 1000)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
    gluLookAt(0,2,4, 0,0,0, 0,1,0)

    t += 0.01
    glLightfv(GL_LIGHT0, GL_POSITION,
[math.sin(t), 1, 0, 1])
```

조명의 위치를 $\sin()$ 함수로 x 의 값을 변화

```
sqrt2 = math.sqrt(2.0)

glBegin(GL_TRIANGLES)

# triangle 1
glColor3fv([1, 0, 0])
p1 = np.array([0, 1, 0])
p2 = np.array([0, 1, 1])
p3 = np.array([1, 0, 0])
u = p2 - p1 # vector from p1 to p2
v = p3 - p1 # vector from p1 to p3
normal = np.cross(u, v)
normal = normal / (np.linalg.norm(normal))
glNormal3fv(normal)
glVertex3fv(p1)
glVertex3fv(p2)
glVertex3fv(p3)

# triangle 2
glColor3fv([0, 1, 1])
p1 = np.array([0, 1, 0])
p2 = np.array([-1, 0, 0])
p3 = np.array([0, 1, 1])
u = p2 - p1 # vector from p1 to p2
v = p3 - p1 # vector from p1 to p3
normal = np.cross(u, v)
normal = normal / (np.linalg.norm(normal))
glNormal3fv(normal)
glVertex3fv(p1)
glVertex3fv(p2)
glVertex3fv(p3)
glEnd()

glFlush()
```

6 법선벡터와 조명(3)

`glutInitDisplayMode(GLUT_SINGLE | GLUT_DEPTH | GLUT_RGBA)`에서 깊이버퍼를 사용하여 앞뒤관계를 표현한다.

```
def main():
    # windowing
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE | GLUT_DEPTH | GLUT_RGBA)
    glutInitWindowSize(600, 600)
    glutInitWindowPosition(10, 10)
    glutCreateWindow(b"Normal & Light on Triangles")

    GLinit()

    # register callbacks
    glutDisplayFunc(display)
    glutIdleFunc(display)

    # enter main-loop
    glutMainLoop()

if __name__ == '__main__':
    main()
```


7 메쉬파일 불러오기

- 파일에 저장된 정점과 면의 정보를 읽어서 이를 표현

```
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *

import math
import numpy as np

def loadMesh(filename):
    print(filename)
    with open(filename, "rt") as mesh :
        nV = int(next(mesh))
        verts = [[0,0,0] for idx in range(nV)]
        for i in range(0, nV) :
            verts[i][0], verts[i][1], verts[i][2] = [float(x) for x in next(mesh).split()]
        nF = int(next(mesh))
        faces = [[0,0,0] for idx in range(nF)]
        for i in range(0, nF) :
            faces[i][0], faces[i][1], faces[i][2] = [int(x) for x in next(mesh).split()]
    return verts, faces

V, F = loadMesh("testMesh.txt")

print(V)
print(F)
```

8 메쉬파일 불러오기(2)

```
# Light Properties
Ld = [1,1,1,1]
Ls = [1,1,1,1]
# Material Properties
Md = [1,1,0,1]
Ms = [1,1,1,1]
shininess = [120]

t = 0

# initialization
def GLinit() :
    #색상,z-buffer, 조명 설정
    glClearColor(0,0,1,0)
    glEnable(GL_COLOR_MATERIAL)
    glEnable(GL_DEPTH_TEST)
    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
    glLightfv(GL_LIGHT0, GL_DIFFUSE, Ld)
    glLightfv(GL_LIGHT0, GL_SPECULAR, Ls)
    glMaterialfv(GL_FRONT, GL_DIFFUSE, Md)
    glMaterialfv(GL_FRONT, GL_SPECULAR, Ms)
    glMaterialfv(GL_FRONT, GL_SHININESS, shininess)
```

```
def computeNormal(p1, p2, p3) :
    u = np.array([p2[i] - p1[i] for i in range(0, 3)])
    v = np.array([p3[i] - p1[i] for i in range(0, 3)])
    N = np.cross(u,v)
    N = N / np.linalg.norm(N)
    return N

def drawVerts(v, f):
    glBegin(GL_TRIANGLES)
    for i in range(len(f)) :
        p1, p2, p3 = f[i][0], f[i][1], f[i][2]
        N = computeNormal(v[p1], v[p2], v[p3])
        glNormal3fv(N)
        glVertex3fv(v[p1])
        glVertex3fv(v[p2])
        glVertex3fv(v[p3])
    glEnd()
```


9 메쉬파일 불러오기(3)

```
# display callback
def display() :
    global t, V, F

    glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT)
    # CAMERA SETTING
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(60.0, 1.0, 0.1, 1000)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
    gluLookAt(0,2,4, 0,0,0, 0,1,0)

    t += 0.01
    glLightfv(GL_LIGHT0, GL_POSITION,
[math.sin(t),1,0,1])
    glRotatef(t*10, 0,1,0)

    # draw vertices
    drawVerts(V,F)

    glFlush()
```

```
def main():
    # windowing
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE | GLUT_DEPTH
|GLUT_RGBA)
    glutInitWindowSize(600,600)
    glutInitWindowPosition(10,10)
    glutCreateWindow(b"Light on Mesh in file ")

    GLinit()

    # register callbacks
    glutDisplayFunc(display)
    glutIdleFunc(display)

    # enter main-loop
    glutMainLoop()

if __name__ == '__main__' :
    main()
```

10 메쉬파일 불러오기(4)

- 메쉬파일 정보는 다음과 같다.

```
4
0 1 0
-1 0 0
0 0 1
1 0 0
4
0 1 2
0 2 3
0 3 1
2 3 1
```

- 정점과 면을 읽어들이는 함수 loadMesh() 이해
- 면을 지정하는 순서가 법선벡터와 밀접한 관계가 있음. (순서를 바꾸어 보자)

II 실습문제

1. 실습문제1
2. Lab05-3을 이용하여 육면체를 그려보자. 육면체파일 testCube.txt을 직접 작성하라.
3. 1과 2을 종합하는 프로그램을 작성하라. 즉 육면체를 lab05-1과 같이 여러 개를 배치하고 직접조명, 무대조명을 부가하라.(최종 제출물)

I2 실습문제 제출

이번 실습문제는

- 모든 프로그램은 압출하여 하나의 파일로
- PLMS Lab xx 해당 번호폴더
- 기한은 일주일(11/16)

제출합니다.