

그래픽스 강의노트 04 - 색상 모델 실습

강영민

동명대학교

2021년 2학기

RGB - 광원, 재질 조정 후 관찰색 확인

Lines 1-25 / 150

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

import sys

from PyQt5.QtWidgets import QOpenGLWidget, QApplication, QMainWindow, QLabel,
    QLineEdit, QVBoxLayout, QWidget
from PyQt5.QtWidgets import QGroupBox, QGridLayout, QSlider
from PyQt5.QtCore import *

import numpy as np

class MyGLWidget(QOpenGLWidget):

    def __init__(self, parent=None):
        super(MyGLWidget, self).__init__(parent)
        self.light = np.array([0.0, 0.0, 0.0])
        self.mat = np.array([0.0, 0.0, 0.0])

    def initializeGL(self):
        # OpenGL 그리기를 수행하기 전에 각종 상태값을 초기화
        glClearColor(0.8, 0.8, 0.6, 1.0)

    def resizeGL(self, width, height):
        # 카메라의 투영 특성을 여기서 설정
        glMatrixMode(GL_PROJECTION)
```

RGB - 광원, 재질 조정 후 관찰색 확인

Lines 26-50 / 150

```
glLoadIdentity()

def paintGL(self):
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()

    # 색과 프리미티브를 이용한 객체 그리기
    # 광원의 색
    glColor3fv(self.light)
    glBegin(GL_QUADS)
    glVertex3fv([-1.0, 1.0, 0.0])
    glVertex3fv([-1.0, 0.0, 0.0])
    glVertex3fv([ 0.0, 0.0, 0.0])
    glVertex3fv([ 0.0, 1.0, 0.0])
    glEnd()

    # 색과 프리미티브를 이용한 객체 그리기
    # 물체의 재질 색상
    glColor3fv(self.mat)
    glBegin(GL_QUADS)
    glVertex3fv([0.0, 1.0, 0.0])
    glVertex3fv([0.0, 0.0, 0.0])
    glVertex3fv([1.0, 0.0, 0.0])
    glVertex3fv([1.0, 1.0, 0.0])
```

RGB - 광원, 재질 조정 후 관찰색 확인

Lines 51-75 / 150

```
glEnd()
```

```
# 색과 프리미티브를 이용한 객체 그리기
```

```
# 눈에 보이는 색
```

```
glColor3fv(self.light * self.mat)
```

```
glBegin(GL_QUADS)
```

```
glVertex3fv([-0.5, -0.2, 0.0])
```

```
glVertex3fv([-0.5, -0.8, 0.0])
```

```
glVertex3fv([ 0.5, -0.8, 0.0])
```

```
glVertex3fv([ 0.5, -0.2, 0.0])
```

```
glEnd()
```

```
# 그려진 프레임버퍼를 화면으로 송출
```

```
glFlush()
```

```
def setL(self, channel, val):  
    self.light[channel] = val/99  
    self.update()
```

```
def setM(self, channel, val):  
    self.mat[channel] = val/99  
    self.update()
```

```
class MyWindow(QMainWindow):
```

```
    def __init__(self, title = ''):
```

RGB - 광원, 재질 조정 후 관찰색 확인

Lines 76-100 / 150

```
QMainWindow.__init__(self)      # call the init for the parent class
self.setWindowTitle(title)

# GUI 설정

central_widget = QWidget()
self.setCentralWidget(central_widget)

gui_layout = QVBoxLayout()      # CentralWidget에 사용될 수직 나열 레이아웃
                                # 배치될 것들 - GL Window + Control
central_widget.setLayout(gui_layout)

self.glWidget = MyGLWidget()    # OpenGL Widget
gui_layout.addWidget(self.glWidget)

self.controlGroup = QGroupBox('Control')
gui_layout.addWidget(self.controlGroup)

control_layout = QGridLayout()
self.controlGroup.setLayout(control_layout)

lightR = QSlider(Qt.Horizontal)
lightR.valueChanged.connect(lambda val: self.glWidget.setL(0, val))

lightG = QSlider(Qt.Horizontal)
```

RGB - 광원, 재질 조정 후 관찰색 확인

Lines 101-125 / 150

```
lightG.valueChanged.connect(lambda val: self.glWidget.setL(1, val))

lightB = QSlider(Qt.Horizontal)
lightB.valueChanged.connect(lambda val: self.glWidget.setL(2, val))

matR = QSlider(Qt.Horizontal)
matR.valueChanged.connect(lambda val: self.glWidget.setM(0, val))

matG = QSlider(Qt.Horizontal)
matG.valueChanged.connect(lambda val: self.glWidget.setM(1, val))

matB = QSlider(Qt.Horizontal)
matB.valueChanged.connect(lambda val: self.glWidget.setM(2, val))

control_layout.addWidget(lightR, 1, 1)
control_layout.addWidget(lightG, 2, 1)
control_layout.addWidget(lightB, 3, 1)
control_layout.addWidget(matR, 1, 2)
control_layout.addWidget(matG, 2, 2)
control_layout.addWidget(matB, 3, 2)

def main(argv = []):
    app = QApplication(argv)
    window = MyWindow('Colors: Light - Material - Observation')
    window.setFixedSize(1200, 800)
```

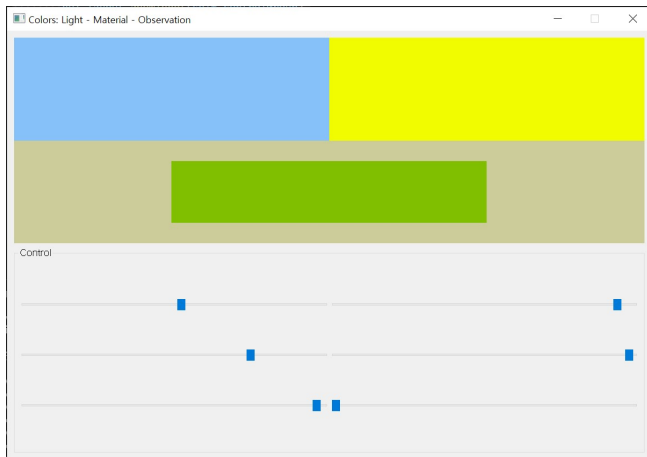
RGB - 광원, 재질 조정 후 관찰색 확인

Lines 126-150 / 150

```
        window.show()
        sys.exit(app.exec_())

if __name__ == '__main__':
    main(sys.argv)
{"mode": "full", "isActive": false}
```

간단한 OpenGL 프로그램의 실행 결과



RGB - 광원, 재질 조정 후 관찰색 확인

Lines 1-25 / 100

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

import sys

from PyQt5.QtWidgets import QOpenGLWidget, QApplication, QMainWindow, QLabel,
    QLineEdit, QVBoxLayout, QWidget
from PyQt5.QtWidgets import QGroupBox, QGridLayout, QSlider
from PyQt5.QtCore import *

import numpy as np

class MyGLWidget(QOpenGLWidget):

    def __init__(self, parent=None):
        super(MyGLWidget, self).__init__(parent)
        self.colors = []
        self.colors.append(np.array([0.0, 0.0, 0.0]))
        self.colors.append(np.array([0.0, 0.0, 0.0]))
        self.colors.append(np.array([0.0, 0.0, 0.0]))

    def initializeGL(self):
        # OpenGL 그리기를 수행하기 전에 각종 상태값을 초기화
        glClearColor(0.8, 0.8, 0.6, 1.0)

    def resizeGL(self, width, height):
```

RGB - 광원, 재질 조정 후 관찰색 확인

Lines 26-50 / 100

```
# 카메라의 투영 특성을 여기서 설정
glMatrixMode(GL_PROJECTION)
glLoadIdentity()

def paintGL(self):
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()

# 색과 프리미티브를 이용한 객체 그리기
glBegin(GL_TRIANGLES)
glColor3fv(self.colors[0])
glVertex3fv([ 0.0, 1.0, 0.0])
glColor3fv(self.colors[1])
glVertex3fv([-1.0, 0.0, 0.0])
glColor3fv(self.colors[2])
glVertex3fv([ 1.0, 0.0, 0.0])
glEnd()

# 그려진 프레임버퍼를 화면으로 송출
glFlush()

def setColor(self, point, channel, val):
    self.colors[point][channel] = val/99
    self.update()
```

RGB - 광원, 재질 조정 후 관찰색 확인

Lines 51-75 / 100

```
class MyWindow(QMainWindow) :  
  
    def __init__(self, title = '') :  
        QMainWindow.__init__(self)      # call the init for the parent class  
        self.setWindowTitle(title)  
  
        ### GUI 설정  
  
        central_widget = QWidget()  
        self.setCentralWidget(central_widget)  
  
        gui_layout = QVBoxLayout()      # CentralWidget에 사용될 수직 나열 레이아웃  
                                         # 배치될 것들 - GL Window, Control  
        central_widget.setLayout(gui_layout)  
  
        self.glWidget = MyGLWidget()    # OpenGL Widget  
        gui_layout.addWidget(self.glWidget)  
  
        self.controlGroup = QGroupBox('Control')  
        gui_layout.addWidget(self.controlGroup)  
  
        control_layout = QGridLayout()  
        self.controlGroup.setLayout(control_layout)
```

RGB - 광원, 재질 조정 후 관찰색 확인

Lines 76-100 / 100

```
for i in range(3):      # 3 points
    for j in range(3): # RGB
        slider = QSlider(Qt.Horizontal)
        name = '{} {}'.format(i, j)
        slider.setObjectName(name)
        slider.valueChanged.connect(self.changeColor)
        control_layout.addWidget(slider, j, i)

def changeColor(self, value):
    i, j = self.sender().objectName().split()
    self.glWidget.setColor(int(i), int(j), value)

def main(argv = []):
    app = QApplication(argv)
    window = MyWindow('Color Interpolation')
    window.setFixedSize(400, 800)
    window.show()
    sys.exit(app.exec_())

if __name__ == '__main__':
    main(sys.argv)
```

간단한 OpenGL 프로그램의 실행 결과

