

# 컴퓨터그래픽스

Lab #5 - myLighting -

제 출 일	2017년 11월 13일
분 반	00반
담당교수	남병규
학 과	컴퓨터공학과
학 번	201302423
이 름	신종욱

## 1. Purpose of the lab

Light할 때 쓰이는 함수들 직접 구현해보기

## 2. Source code

```
struct lightmaterial
{
    double ambient[4];
    double diffuse[4];
    double specular[4];
    double diffuselight;
    double specularlight;
    double ambientlight;
}; //물체의 속성을 가진 구조체
lightmaterial identity;
//기본 속성을 전역으로 선언

struct RGBL
{
    double RGBLvalue[4];
}; //RGBL의 구조체
enum PROP {AMB,DIFF,SPEC}; //확산광 주변광 경면광을 나타내기 위한 enum이다.
enum light{ GL_LIGHT0, GL_LIGHT1, GL_LIGHT2, GL_LIGHT3, GL_LIGHT4, GL_LIGHT5, GL_LIGHT6, GL_LIGHT7, GL_LIGHTING};
//light의 세기에 대해서 나타내기 위한 enum이다.
```

구현에 필요한 구조체 선언과 enum클래스 선언부이다.

enum클래스에서 GL\_LIGHTING은 GLENABLE 사용때 하기 위한 것이다.

```
void mySetLight(light ID, PROP attri, struct RGBL coef) {
    if (attri == 0) {
        myAmbient(ID, coef);
    }
    else if (attri == 1) {
        myDiffuse(ID, coef);
    }
    else if (attri == 2) {
        mySpecular(ID, coef);
    }
    //들어온 PROP의 값에 따라 실행되는 함수가 선택된다.
}
```

mySetLight는 들어온 PROP에 따라 확산,주변,경면 광의 값을 변경해주는 메소드이다.

```

void myAmbient(light ID, struct RGBL coef) {
    if (identity.ambientlight == 12880) { //ENABLE일 경우에만 실행
        identity.ambient[0] = coef.RGBLvalue[0];
        identity.ambient[1] = coef.RGBLvalue[1];
        identity.ambient[2] = coef.RGBLvalue[2];
        identity.ambient[3] = coef.RGBLvalue[3];
        identity.ambientlight = 16385 + ID;
    }
    else printf("LightENABLE이 안되었습니다.");
}

void myDiffuse(light ID, struct RGBL coef) {
    if (identity.diffuselight == 12880){
        identity.diffuse[0] = coef.RGBLvalue[0];
        identity.diffuse[1] = coef.RGBLvalue[1];
        identity.diffuse[2] = coef.RGBLvalue[2];
        identity.diffuse[3] = coef.RGBLvalue[3];
        identity.diffuselight = 16385 + ID;
    }
    else printf("LightENABLE이 안되었습니다.");
}

void mySpecular(light ID, struct RGBL coef) {
    if (identity.specularlight == 12880){
        identity.specular[0] = coef.RGBLvalue[0];
        identity.specular[1] = coef.RGBLvalue[1];
        identity.specular[2] = coef.RGBLvalue[2];
        identity.specular[3] = coef.RGBLvalue[3];
        identity.specularlight = 16385 + ID;
    }
    else printf("LightENABLE이 안되었습니다.");
}

```

확산,주변,경면광들의 RGBL값들을 바꾸는 메소드이다. 만약 LightEnable을 안 했을 경우에는 에러메세지 출력으로 나타내었고 GL.h파일을보니 light0은 0x4000 이고 light1는 0x4001이라서 0x4000인 16385에 enum은 순서대로 0,1,2,3이니 ID값과 0x4000 더한값을 light로 저장하였다.

```

void myAttenuation(double A, double B, double C, double Dist ) {
    double Attenuation = 1.0 / (A + B*Dist + C*Dist*Dist);
    identity.diffuselight *= Attenuation;
    identity.specularlight *= Attenuation;
    //거리에 따른 light 조절
}

void myLighting(light ID) {
    if (ID == 8) {
        identity.diffuselight = 12880;
        identity.specularlight = 12880;
        identity.ambientlight = 12880;
    }
} //GLEnable

```

거리에따른 밝기의 감소를 나타내고싶을땐 myAttenuation을 써서 확산광과 경면광의 값들을 나눠준다. myLighting함수는 GLEnable과 같은 효과로 조명사용하겠다는 표시이다. 12880값은 GL.h에서 있는 GLlighting값이 0x0B50이라서 저렇게 설정하였고 ID==8은 인자가 GLlight가 들어왔을 때 실행한다고 구현하려고 추가한 조건이다.

```

//////////////////////////////////////////////////
void reset() {

    identity.diffuselight=0;
    identity.specularlight=0;
    identity.ambientlight=0;

    identity.diffuse[0] = { 1.0 };
    identity.diffuse[1] = { 1.0 };
    identity.diffuse[2] = { 1.0 };
    identity.diffuse[3] = { 1.0 };

    identity.ambient[0] = { 1.0 };
    identity.ambient[1] = { 1.0 };
    identity.ambient[2] = { 1.0 };
    identity.ambient[3] = { 1.0 };

    identity.specular[0] = { 1.0 };
    identity.specular[1] = { 1.0 };
    identity.specular[2] = { 1.0 };
    identity.specular[3] = { 1.0 };

}

void main(int argc, char **argv)
{
    reset();
    myLighting(GL_LIGHTING);
    struct RGBL red;
    red.RGBLvalue[0] = 1;
    red.RGBLvalue[1] = 0;
    red.RGBLvalue[2] = 0;
    red.RGBLvalue[3] = 1;

    struct RGBL redbule;
    redbule.RGBLvalue[0] = 0.5;
    redbule.RGBLvalue[1] = 0;
    redbule.RGBLvalue[2] = 0.5;
    redbule.RGBLvalue[3] = 1;

    struct RGBL redgreen;
    redgreen.RGBLvalue[0] = 0.5;
    redgreen.RGBLvalue[1] = 0.5;
    redgreen.RGBLvalue[2] = 0;
    redgreen.RGBLvalue[3] = 1;
}

```

reset함수는 전역변수인 물체속성을 초기화 하기위한 함수이다.

```

////////////////////////////////////
mySetLight(GL_LIGHT0, AMB, red);
mySetLight(GL_LIGHT1, DIFF, redbule);
mySetLight(GL_LIGHT2, SPEC, redgreen);
printlight();

myAttenuation(3, 5, 1, 10);
printlight();

////////////////////////////////////
reset();
myLighting(GL_LIGHTING);

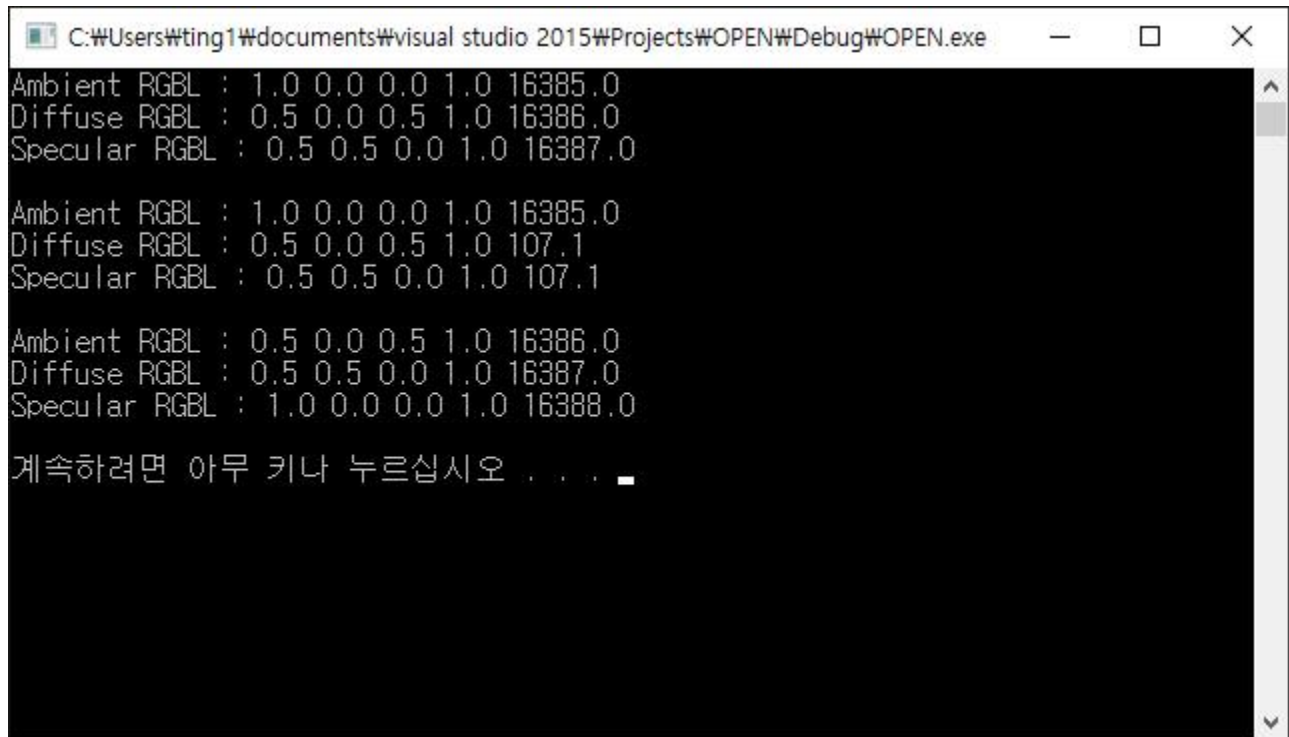
mySetLight(GL_LIGHT1, AMB, redbule);
mySetLight(GL_LIGHT2, DIFF, redgreen);
mySetLight(GL_LIGHT3, SPEC, red);
printlight();

system("pause");
}

```

main함수 코드로 RGBL값을 지정해주고 값을넣고 Dist전후로 출력해보고  
reset후 확산,주변,경면광들의 RGBL값을 바꾸고 출력해보았다.

### 3. Results



The screenshot shows a Windows command prompt window titled "C:\Users\Wting1\documents\visual studio 2015\Projects\OPEN\Debug\OPEN.exe". The window contains the following text:

```
Ambient RGBL : 1.0 0.0 0.0 1.0 16385.0  
Diffuse RGBL : 0.5 0.0 0.5 1.0 16386.0  
Specular RGBL : 0.5 0.5 0.0 1.0 16387.0  
  
Ambient RGBL : 1.0 0.0 0.0 1.0 16385.0  
Diffuse RGBL : 0.5 0.0 0.5 1.0 107.1  
Specular RGBL : 0.5 0.5 0.0 1.0 107.1  
  
Ambient RGBL : 0.5 0.0 0.5 1.0 16386.0  
Diffuse RGBL : 0.5 0.5 0.0 1.0 16387.0  
Specular RGBL : 1.0 0.0 0.0 1.0 16388.0  
  
계속하려면 아무 키나 누르십시오 . . .
```

2번째를보면 거리에 따른 빛의 감소를 나타내 주었더니 light값이 줄어든걸 볼 수 있다.

### 4. Discussions

일단 최대한 비슷하게 OpenGL을 따라 해봤는데 의문인게 LIGHT값들은 다들 큰값으로 선언 되어있는데 특별한 의미가 있는지 궁금하다 그리고 이 LIGHT값의 차에 따라 나타나는 차이가 큰지도 궁금하다.