

# Exercício V – Parte II (2012/I)

Disciplina: Computação Gráfica

Professora: Soraia R. Musse

## Aula prática em OpenGL – Iluminação e Sombreamento

O **objetivo desta aula** é trabalhar com as diferentes técnicas de iluminação e sombreado de objetos usando a biblioteca **OpenGL**, a fim de proporcionar mais realismo à nossa cena. Você pode obter os códigos fontes que serão utilizados na aula de hoje na página da disciplina. Este programa possui implementações do exercício anterior.

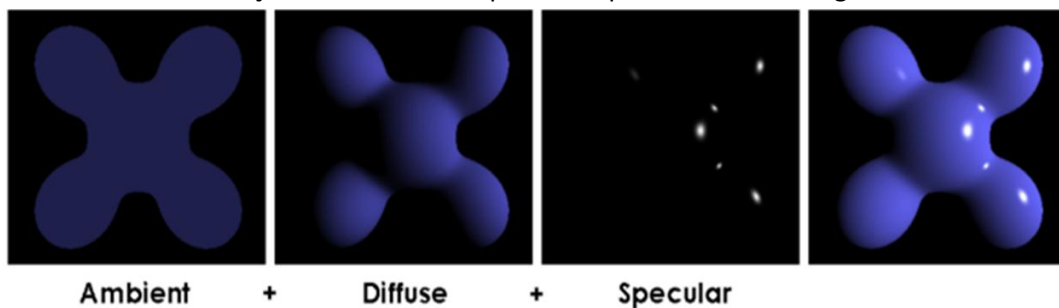
Inicialmente, verifique o conteúdo do programa fonte fornecido. Compile e execute o programa e verifique que aparecerá uma janela com um tabuleiro representando o chão, um objeto *teapot* no centro da cena, e uma esfera emissora de luz.

No mundo real, para que seja possível enxergar objetos em um ambiente, é fundamental que exista pelo menos uma fonte de luz. De forma simplificada, objetos são visíveis porque refletem e absorvem raios de luz. Como na Computação Gráfica procura-se simular o que acontece na realidade, é necessário incluir fontes de luz no ambiente virtual 3D e descrever como será a interação dessas fontes com os demais objetos da cena.

A iluminação da cena é criada a partir da combinação de componentes que podem variar de forma independente, como foi visto na aula anterior. As principais componentes são:

- Luz Ambiente: luz sem direção específica, resultante das reflexões entre as muitas superfícies presentes no ambiente. Ilumina todos os objetos da cena.
- Reflexão Difusa: A iluminação do objeto varia de acordo com a direção e a distância desta fonte de luz. O brilho dependerá somente do ângulo formado entre a direção da fonte da luz e a normal da superfície, independente da direção do observador.
- Reflexão Especular: Círculos brilhantes (*highlight*) que aparecem em superfícies. O brilho varia de acordo com a posição do observador.

O resultado da combinação destes três componentes pode ser visto na figura abaixo.



Voltando para o código OpenGL, observe o conteúdo do método `Inicializa()`. Ele possui uma chamada para o método `Definelluminacao()`, que configura os parâmetros de iluminação da nossa cena. Além disso, possui mais 3 linhas importantes, que permitem habilitar as funcionalidades de iluminação na cena:

- `glEnable(GL_COLOR_MATERIAL)`: Habilita a definição da cor do material a partir da cor corrente;

- glEnable(GL\_LIGHTING): Habilita o uso de iluminação;
- glEnable(GL\_LIGHT0): Habilita a luz de número 0.

Dentro do método Definelluminacao(), podemos selecionar o nosso modelo de tonalização. Existem diferentes modelos de tonalização (*shading*) na literatura, como vimos aula passada. Os três mais utilizados são *flat*, *Gouraud* e *Phong*. A OpenGL suporta diretamente apenas *flat* e *Gouraud*.

**A primeira tarefa é visualizar as diferentes técnicas de tonalização. Para isso, altere a linha de código glShadeModel(GL\_SMOOTH), para glShadeModel(GL\_FLAT). Veja a diferença entre o modelo de Gourad (GL\_SMOOTH) para o de Flat (GL\_FLAT).**

A seguir, são definidas as propriedades do material para o nosso *teapot*, através da função void glMaterialfv(GLenum face, GLenum pname, TYPE \*param):

```
// Define a refletância do material
glMaterialfv(GL_FRONT, GL_SPECULAR, especularidade);
// Define a concentração do brilho
glMateriali(GL_FRONT, GL_SHININESS, especMaterial);
```

A função recebe 3 parâmetros. O primeiro, GL\_FRONT, quer dizer que apenas as faces frontais do(s) objeto(s) possuirão as propriedades que estão sendo definidas, ou seja, somente as faces que estarão visíveis para o observador. O parâmetro *pname* indica a propriedade do material que está sendo especificada. A constante GL\_SPECULAR é usada quando se deseja que o objeto tenha brilho, nesse caso o terceiro parâmetro irá definir a cor e a intensidade do brilho. Quando a especular é utilizada, para especificar a intensidade e o tamanho de concentração do brilho, a constante GL\_SHININESS deve ser passada como segundo parâmetro em uma nova chamada da função glMateriali. Nesse caso, o último parâmetro deve ser apenas um número (*int* ou *float*) entre 0 e 128 (coeficiente especular).

A seguir, são especificadas as propriedades do modelo de iluminação, através da função glLightModelfv (GLenum pname, TYPE \*param):

```
// Ativa o uso da luz ambiente
glLightModelfv(GL_LIGHT_MODEL_AMBIENT, luzAmbiente);
```

O primeiro parâmetro é usado para definir a característica do modelo de iluminação. No caso, definimos que a intensidade ambiente RGBA será a mesma para toda cena, através da constante GL\_LIGHT\_MODEL\_AMBIENT. Os valores que devem ser usados com o modelo selecionado são definidos por meio do segundo parâmetro, que corresponde a um vetor de float.

Depois, são definidos os parâmetros de uma fonte de luz que estará presente na cena, usando a função glLightfv(GLenum light, GLenum pname, TYPE \*param):

```
// Define os parâmetros da luz de número 0
glLightfv(GL_LIGHT0, GL_AMBIENT, luzAmbiente);
glLightfv(GL_LIGHT0, GL_DIFFUSE, luzDifusa );
glLightfv(GL_LIGHT0, GL_SPECULAR, luzEspecular );
glLightfv(GL_LIGHT0, GL_POSITION, posicaoLuz );
```

O primeiro parâmetro indica a fonte de luz desejada, de GL\_LIGHT0 a GL\_LIGHT7. O segundo define as características da luz, onde selecionamos as seguintes constantes:

- GL\_AMBIENT: Cor ambiente da luz (r,g,b,a).
- GL\_DIFFUSE: Cor difusa da luz (r,g,b,a).
- GL\_SPECULAR: Cor especular da luz (r,g,b,a).
- GL\_POSITION: Posição da luz. (x,y,z,w). O parâmetro w especifica se a luz é puntual (possui uma posição exata na cena) ou direcional (se encontra no infinito ou a uma distância muito grande, como o sol). W = 0.0 significa que a fonte de luz é direcional.

O último parâmetro consiste em um vetor do tipo GLfloat que determina o valor para o qual *pname* é especificado.

**Agora vamos adicionar alguns controles à nossa cena para que possamos interagir com os componentes da iluminação. Na função GerenciaTecladoEspecial(...) adicione o seguinte trecho de código.**

```
switch (key)
{
    case GLUT_KEY_LEFT:
        posicaoLuz[0] -= 2;
        break;
    case GLUT_KEY_RIGHT:
        posicaoLuz[0] += 2;
        break;
    case GLUT_KEY_UP:
        posicaoLuz[1] += 2;
        break;
    case GLUT_KEY_DOWN:
        posicaoLuz[1] -= 2;
        break;
    case GLUT_KEY_PAGE_UP:
        posicaoLuz[2] -= 2;
        break;
    case GLUT_KEY_PAGE_DOWN:
        posicaoLuz[2] += 2;
        break;
}
```

Ao pressionar as teclas definidas pelo código acima, mudar a posição da fonte de luz na cena. Experimente reposicionar a fonte de luz para ver o efeito causado no objeto.

Agora vamos alterar as propriedades do modelo de iluminação, a fim de visualizar melhor os efeitos de cada uma delas. Para isso adicione o seguinte trecho de código no método GerenciaTeclado(...):

```
switch (key)
{
    case 'q':
        luzAmbiente[0] -= 0.1;
        luzAmbiente[1] -= 0.1;
        luzAmbiente[2] -= 0.1;
        break;
    case 'w':
        luzAmbiente[0] += 0.1;
        luzAmbiente[1] += 0.1;
        luzAmbiente[2] += 0.1;
        break;
    case 'a':
        luzDifusa[0] -= 0.1;
```

```

        luzDifusa[1] -= 0.1;
        luzDifusa[2] -= 0.1;
        break;
case 's':
    luzDifusa[0] += 0.1;
    luzDifusa[1] += 0.1;
    luzDifusa[2] += 0.1;
    break;

case 'z':
    luzEspecular[0] -= 0.1;
    luzEspecular[1] -= 0.1;
    luzEspecular[2] -= 0.1;
    break;
case 'x':
    luzEspecular[0] += 0.1;
    luzEspecular[1] += 0.1;
    luzEspecular[2] += 0.1;
    break;
}

```

Veja como cada componente de iluminação pode ser alterado de forma independente, causando o efeito desejado na cena.

**Exercício:** Tente adicionar mais uma fonte de luz à nossa cena, configurando-a com a cor vermelha. Veja o efeito que as múltiplas fontes de luz causam no objeto.