



Computação Gráfica

Aula 6 OpenGL – Realismo e Iluminação

Prof. João Marcelo Ribeiro

Realismo

- Podemos definir Realismo Visual como as técnicas de tratamento computacional aplicadas aos objetos sintéticos gerados (por modelagem de sólidos, partículas, fractais ou qualquer técnica de geração), com objetivo de criar-lhes uma imagem sintética, o mais próximo da realidade que se teria, se eles fossem construídos e filmados.
- Podemos considerar o realismo em duas etapas:
 - Estática: Onde são tratados os objetos e cenas estáticas é trabalhado o chamado realismo fotográfico.
 - Dinâmica: A parte dinâmica esta relacionada ao movimento da cena e seus personagens, chamada de animação.

Realismo Visual (rendering)

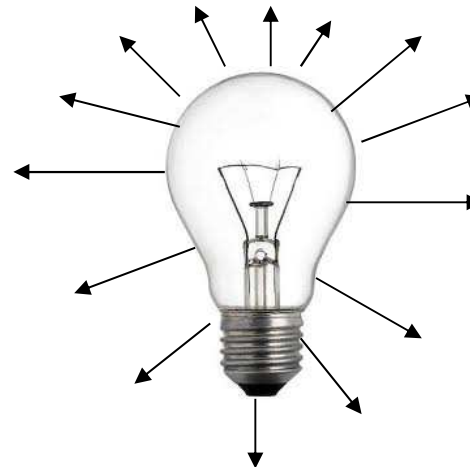
- Sintetizar uma imagem (uma cena ou um objeto) é criá-la em termos da definição dos dados dos objetos que a compõem.
- Isso se faz, a partir da geometria da cena, das informações sobre os materiais de que são feitos os objetos (suas cores e suas texturas), das condições de iluminação ambiente e da posição de observação da cena.
- Nesse processo de criação sintética, é denominado “rendering” a fase de introdução nas cenas, do realismo fotográfico.

Iluminação

- A iluminação é uma técnica importante em objetos gráficos criados através do computador. Sem iluminação os objetos tendem a não apresentar características realistas.
- A iluminação é fundamental para dar a noção de profundidade. Sem iluminação, uma esfera e um disco possuem a mesma representação.
- O princípio de iluminação consiste em simular como os objetos refletem as luzes.
- A luz pode ser emitida ou refletida de objetos. Os emissores de luz são as fontes de luz (lâmpadas, velas, fogo, sol, estrelas) e os refletores de luz são normalmente os objetos que serão coloridos de maneira realística (renderizados).
- As fontes de luz são caracterizadas por suas intensidades e frequências (ou comprimento de onda) enquanto os refletores são caracterizados pelas propriedades de suas superfícies como cor, material, etc.
- A escolha do tipo a ser usado para representar uma fonte ou um emissor de luz depende diretamente do ambiente da cena.

Fontes de Luz

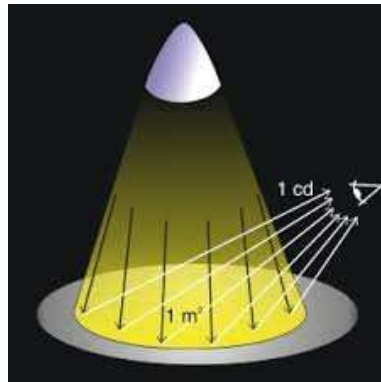
- A natureza de luz diz respeito à fonte de luz utilizada. Geralmente, três tipos diferentes de fonte de luz podem ser incluídos em uma cena 3D: luzes fotométricas (pontual), direcional e refletora (*spot*).
- Na fonte de luz fotométrica, a fonte de energia pode ser direcionada para um ponto, então, a forma pontual (uma única lâmpada) é aquela cujos raios de luz emanam uniformemente em todas as direções a partir de um único ponto. Neste caso, a iluminação de um objeto varia de uma parte para outra, dependendo da direção e da distância da fonte de luz.



- Uma fonte de luz direcional é aquela cujos raios de luz vêm sempre da mesma direção. Portanto, o seu efeito é percebido dependendo da orientação da superfície. Por exemplo, uma face é plenamente iluminada se estiver perpendicular aos raios de luz incidentes. Quanto mais oblíqua uma face estiver em relação aos raios de luz, menor será sua iluminação.



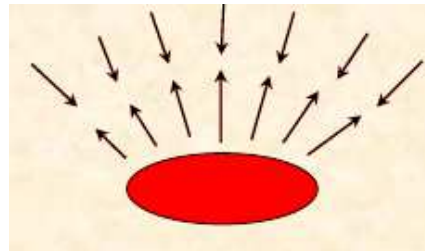
- Uma fonte de luz spot é semelhante à luz pontual, porém os raios de luz são emitidos na forma de um cone, apontado para determinada direção. Isso faz com que a intensidade da luz diminua conforme o raio de luz se desvia dessa direção.



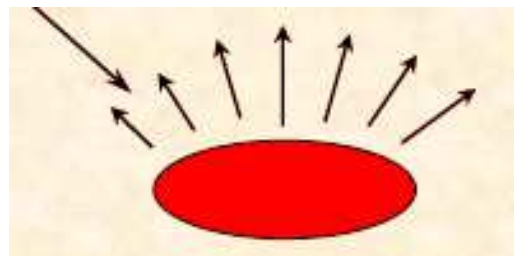
Modos de Reflexão

- O modelo de reflexão descreve interação dos raios de luz com uma superfície, considerando as propriedades da superfície e a natureza da fonte de luz incidente.
- O seu principal objetivo é exibir os objetos tridimensionalmente na tela bidimensional de maneira que os mesmos se aproximem da realidade.
- Além disso, por meio de um modelo de reflexão é possível fazer com que objetos do tipo espelho apresentem em sua superfície a imagem de outros objetos do universo.
- Diversas informações são necessárias para que um modelo de reflexão processe a intensidade da cor de cada ponto a ser exibido, tais como: cor do objeto, cor da fonte de luz, posição da fonte de luz, posição da cena 3D do ponto a ser exibido e posição do observador virtual.
- Podemos citar como modelos de reflexão os tipos: Ambiente, Difusa, Especular e Emissiva.

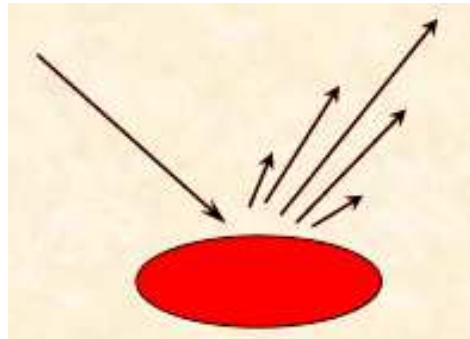
- O modelo de reflexão ambiente origina-se da interação e da reflexão dos raios de luz com todas as superfícies da cena e é incidente em uma superfície a partir de todas as direções. Outra forma de interpretar a luz ambiente é como a luz que está presente no ambiente, mas cuja origem não pode ser precisamente determinada. Portanto, esse tipo de reflexão permite a visibilidade de superfícies que não estejam recebendo diretamente raios de luz.



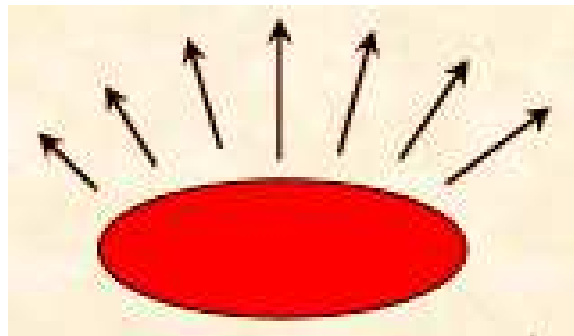
- O modelo de reflexão difusa (difundida) ocorre na superfície da maioria dos objetos que não emitem luz. Todo objeto absorve a luz do sol ou a luz emitida de uma fonte artificial e reflete parte desta luz. Esse tipo de reflexão depende da cor do objeto e da posição da fonte de luz: a quantidade de luz refletida percebida pelo observador não depende da sua posição. A reflexão difusa cria o efeito de *degradé* nos objetos, se consideramos que os objetos apresentam uma superfície total ou parcialmente opaca. – Luz proveniente de uma direção bem definida e refletiva igualmente segundo todas as direções.



- O modelo de reflexão especular é a responsável pela geração do ponto de brilho dos objetos. Essa reflexão é processada de acordo com a cor do objeto, a posição da luz e a posição do observador – e a cor do brilho em geral é a mesma da fonte de luz. A posição do observador é importante, pois a luz refletida de uma superfície reflexiva deixa a superfície com o mesmo ângulo que o raio de luz incidente forma com o vetor normal da superfície.



- O modelo de reflexão emissiva simula a luz que se origina de um objeto; a cor emissiva de uma superfície adiciona intensidade ao objeto, mas não é afetada por qualquer fonte de luz; ela também não introduz luz adicional da cena. – A luz emitida pela própria superfície do objeto, não afetada por qualquer fonte de luz.



Modelos de Tonalização - Shading

- Os modelos de tonalização são usados para determinar a cor de cada ponto que compõe a superfície de um objeto.
- O realismo de uma imagem depende diretamente do modelo de tonalização aplicado.
- Duas informações são fundamentais para os modelos de tonalização: quais são as propriedades da superfície e quais são as propriedades da iluminação que estão sendo aplicadas sobre ela.
- As principais propriedades da superfície são reflexão e transparência. Com relação à iluminação, deve-se saber qual o modelo que está sendo considerado.
- Os três modelos de tonalização mais utilizados são: *Flat (Flat Shading)*, *Gouraud* e *Phong*.

- O modelo Flat, também conhecido como constante ou facetado, é o modelo mais simples, no qual se determina um único valor de intensidade para cada face, é utilizado para colorizar a face inteira. Como esta técnica não produz variações de tonalização, pode ocorrer um problema de descontinuidade da intensidade.
- O modelo Gouraud é o modelo que busca obter suavidade na exibição de objetos com superfícies curvas quando representados por faces, eliminando parcialmente o problema da descontinuidade. A melhora no resultado da exibição do objeto deve-se à interpolação de intensidade ao longo de cada face.
- O modelo Phong é a luz em qualquer ponto é composta por três componentes: **luz difusa**, **luz especular** e **luz ambiente**. Essas três componentes são aditivas e determinam o aspecto final da iluminação e da cor de um determinado ponto na cena ou da superfície de um determinado polígono plano contido nela.

Comandos OpenGL

- O OpenGL divide iluminação em 3 partes:
 - Propriedades de materiais
 - Propriedades de luzes
 - Parâmetros globais de iluminação
- A iluminação está disponível em ambos os modos, RGBA e modo de índice de cores.
- Basicamente existem dois comandos que serão usados na criação de um modelo de iluminação:
 - Light
 - Define as características da fonte de luz
 - Material
 - Define as propriedades do material do objeto

- Cor de uma fonte de luz
 - é caracterizada pela quantidade de vermelho, verde e azul que ela emite
- Material
 - É caracterizado pela porcentagem dos componentes R, G e B que chegam e são refletidos em várias direções
- No modelo de iluminação a luz em uma cena vem de várias fontes de luz que podem ser “ligadas” ou “desligadas” individualmente.
- A fonte de luz tem efeito somente quando existem superfícies que absorvem e refletem luz.
- O material pode:
 - Emitir luz
 - Refletir parte da luz incidente em todas as direções ou numa única direção

- No OpenGL os comandos necessários para trabalhar com a remoção das faces escondidas e permitir uma correta visualização dos objetos em 3D são:
- Inicialmente, deve-se indicar que se quer utilizar o buffer de profundidade.
 - `glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);`
- Habilitar o depth-buffering
 - `glEnable(GL_DEPTH_TEST);`
- Função usada para definir quais buffers devem ser inicializados.
 - `glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);`
 - Limpa a janela de visualização com a cor de fundo definida previamente

- No OpenGL, as funções utilizadas para especificação das fontes de luzes são:
- **glLightfv(GLenum light, GLenum pname, TYPE *param);**
 - Função usada para especificar as propriedades de uma fonte de luz. O primeiro parâmetro *light* indica a fonte de luz desejada, de GL_LIGHT0 a GL_LIGHT7. que permite especifica as características da projeção perspectiva para visualização de cenas 3D.
 - O parâmetro *light* indica a fonte de luz desejada, de GL_LIGHT0 a GL_LIGHT7.
 - O parâmetro *pname* define as características da luz e pode receber uma das constantes abaixo:
 - **TABELA**
 - O parâmetro *param* consiste em um vetor do tipo GLfloat que determina o valor para o qual *pname* é especificado.

TABELA LIGHTFV

Nome do parâmetro	Valor Padrão	Significado
GL_AMBIENT	(0.0, 0.0, 0.0, 1.0)	Intensidade RGBA da luz ambiente
GL_DIFFUSE	(1.0, 1.0, 1.0, 1.0)	Intensidade RGBA da luz difusa
GL_SPECULAR	(1.0, 1.0, 1.0, 1.0)	Intensidade RGBA da luz especular
GL_POSITION	(0.0, 0.0, 1.0, 0.0)	Posição da luz (x,y,z,w)
GL_SPOT_DIRECTION	(0.0, 0.0, -1.0)	Direção da luz spotlight (x,y,z)
GL_SPOT_EXPONENT	0.0	Expoente spotlight
GL_SPOT_CUTOFF	180.0	Ângulo do spotlight
GL_CONSTANT_ATTENUATION	1.0	Fator de constante de atenuação
GL_LINEAR_ATTENUATION	1.0	Fator de atenuação linear
GL_QUADRATIC_ATTENUATION	0.0	Fator de atenuação quadrática

- Função usada para especificar as propriedades do modelo de iluminação.
- **glLightModelfv(GLenum pname, TYPE *param);**
 - Função usada para especificar as propriedades do modelo de iluminação.
 - O parâmetro *pname* é usado para definir a característica do modelo de iluminação e pode assumir um dos valores listados abaixo:
 - ***TABELA***
 - O parâmetro *param* consiste nos valores que devem ser usados com o modelo selecionado.

TABELA LIGHTMODEL

Nome do Parâmetro	Valor Padrão	Significado
GL_LIGHT_MODEL_AMBIENT	(0.2, 0.2,0.2,1.0)	Intensidade RGBA ambiente de toda a cena
GL_LIGHT_MODEL_LOCAL_VIEWER	0.0 OU GL_FALSE	Como o ângulo de reflexão especular é calculado
GL_LIGHT_MODEL_TWO_SIDE	0.0 OU GL_FALSE	Define entre a iluminação de um ou dois lados

Exemplo

- `GLfloat luz_ambiente[4] = {0.0, 0.0, 0.0, 1.0}; // valores RGBA para a intensidade da luz ambiente`
- `GLfloat luz_difusa[4] = {1.0, 1.0, 1.0, 1.0}; // valores RGBA para a intensidade da luz difusa`
- `GLfloat luz_especular[4] = {1.0, 1.0, 1.0, 1.0}; // valores RGBA para a intensidade da luz especular`
- `GLfloat luz_posicao[4] = {1.0, 1.0, 1.0, 0.0};`
- `// {x,y,z,w} onde: se w=0, fonte de luz direcional: posição infinita e direção dada por (x,y,z). Se w=1, fonte de luz posicional, com posição dada por (x,y,z) em coordenada de objeto.`
- `glLightfv(GL_LIGHT0, GL_AMBIENT, luz_ambiente);`
- `glLightfv(GL_LIGHT0, GL_DIFFUSE, luz_difusa);`
- `glLightfv(GL_LIGHT0, GL_SPECULAR, luz_especular);`
- `glLightfv(GL_LIGHT0, GL_POSITION, luz_posicao);`

- **// Define a refletância do material**
- `glMaterialfv(GL_FRONT, GL_SPECULAR, especificidade);`

Nome do Parâmetro	Valor Padrão	Significado
GL_AMBIENT	(0.2, 0.2, 0.2, 1.0)	Cor ambiente do material
GL_DIFFUSE	(0.8, 0.8, 0.8, 1.0)	Cor difusa do material
GL_AMBIENT_AND_DIFFUSE		Cores ambiente e difusa do material
GL_SPECULAR	(0.0, 0.0, 0.0, 1.0)	Cor especular do material
GL_SHININESS	0.0	Expoente especular
GL_EMISSION	(0.0, 0.0, 0.0, 1.0)	Cor emissiva do material
GL_COLOR_INDEXES	(0,1,1)	Índice de cores, ambiente, difusa e especular

- **Onde:**
- Podem ser fixadas propriedades materiais separadamente para cada face especificando qualquer uma destas: GL_FRONT ou GL_BACK, ou para ambas as faces simultaneamente por GL_FRONT_AND_BACK.
- GL_DIFFUSE : cor básica de objeto – percentagens RGBA para a luz difusa refletida
- GL_SPECULAR : cor de destaque do objeto - percentagens RGBA para a luz especular refletida
- GL_AMBIENT : cor do objeto quando não diretamente iluminado – percentagens RGBA para a luz ambiente refletida
- GL_EMISSION : cor emitida pelo objeto – valores RGBA para a intensidade da luz emitida pelo próprio material (não influencia a iluminação de outros objetos, ou seja, não atua como fonte de luz).
- GL_SHININESS : concentração de brilhos em objetos. Valores variam de 0 (superfícies muito áspera – nenhum destaque) até 128 (muito brilhante).

Exemplo Completo

```
int main(void)
{
    // Inicialmente, deve-se indicar que se quer utilizar o buffer de profundidade.
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);

    glutInitWindowPosition(5,5);

    glutInitWindowSize(450,450);

    glutCreateWindow("Desenho de uma esfera 3D com iluminação");

    glutDisplayFunc(Desenha);

    glutReshapeFunc(AlteraTamanhoJanela);

    glutKeyboardFunc (Teclado);

    Inicializa();

    glutMainLoop();

    return 0;
}
```

```
void Inicializa (void)
{
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);

    glEnable(GL_COLOR_MATERIAL);

    glEnable(GL_LIGHTING); //Habilita o uso de iluminação

    glEnable(GL_LIGHT0); // Habilita a luz de número 0

    glEnable(GL_DEPTH_TEST); // Habilita o depth-buffering

    glShadeModel(GL_SMOOTH); // Habilita o modelo de tonalização de Gouraud
}
```



```
void AlteraTamanhoJanela(GLsizei w, GLsizei h)
{
    if ( h == 0 ) h = 1;
    glViewport(0, 0, w, h);
    fAspect = (GLfloat)w/(GLfloat)h;

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // Especifica a projeção perspectiva(angulo,aspecto,zMin,zMax)
    gluPerspective(45,fAspect,0.5,500);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    Definelluminacao(); // DEFINIR ILUMINAÇÃO

    // Posiciona e orienta o observador
    glTranslatef(0,0,-150);
}
```

```
void Teclado (unsigned char tecla, int x, int y)
{
    if (tecla == 27)                // Tecla ESC
        exit(0);
}
```

```
void Desenha(void)
{
    //Função usada para definir quais buffers devem ser inicializados.
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    // Chama a função que especifica os parâmetros de iluminação
    Definelluminacao();

    glColor3f(0.0f, 0.0f, 1.0f);

    glutSolidTorus(20.0, 35.0, 20, 40);

    glFlush();
}
```

// Função responsável pela especificação dos parâmetros de iluminação

void Definelluminacao (void)

{

GLfloat luzAmbiente[4]={0.0,0.0,0.0,1.0};

GLfloat luzDifusa[4]={0.7,0.7,0.7,1.0}; // COR

GLfloat luzEspecular[4]={1.0, 1.0, 1.0, 1.0}; // BRILHO

GLfloat posicaoLuz[4]={0.0, 0.0,100.0, 1.0};

// Capacidade de brilho do material

GLfloat especularidade[4]={1.0,1.0,1.0,1.0};

// Define a refletância do material

glMaterialfv(GL_FRONT, GL_SPECULAR, especularidade);

GLfloat especMaterial[1]={60};

// Define a concentração do brilho

glMaterialfv(GL_FRONT, GL_SHININESS, especMaterial);

// Ativa o uso da luz ambiente

glLightModelfv(GL_LIGHT_MODEL_AMBIENT, luzAmbiente);

// 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);

}