Módulo 10

Sistemas Gráficos e Interação

Instituto Superior de Engenharia do Porto

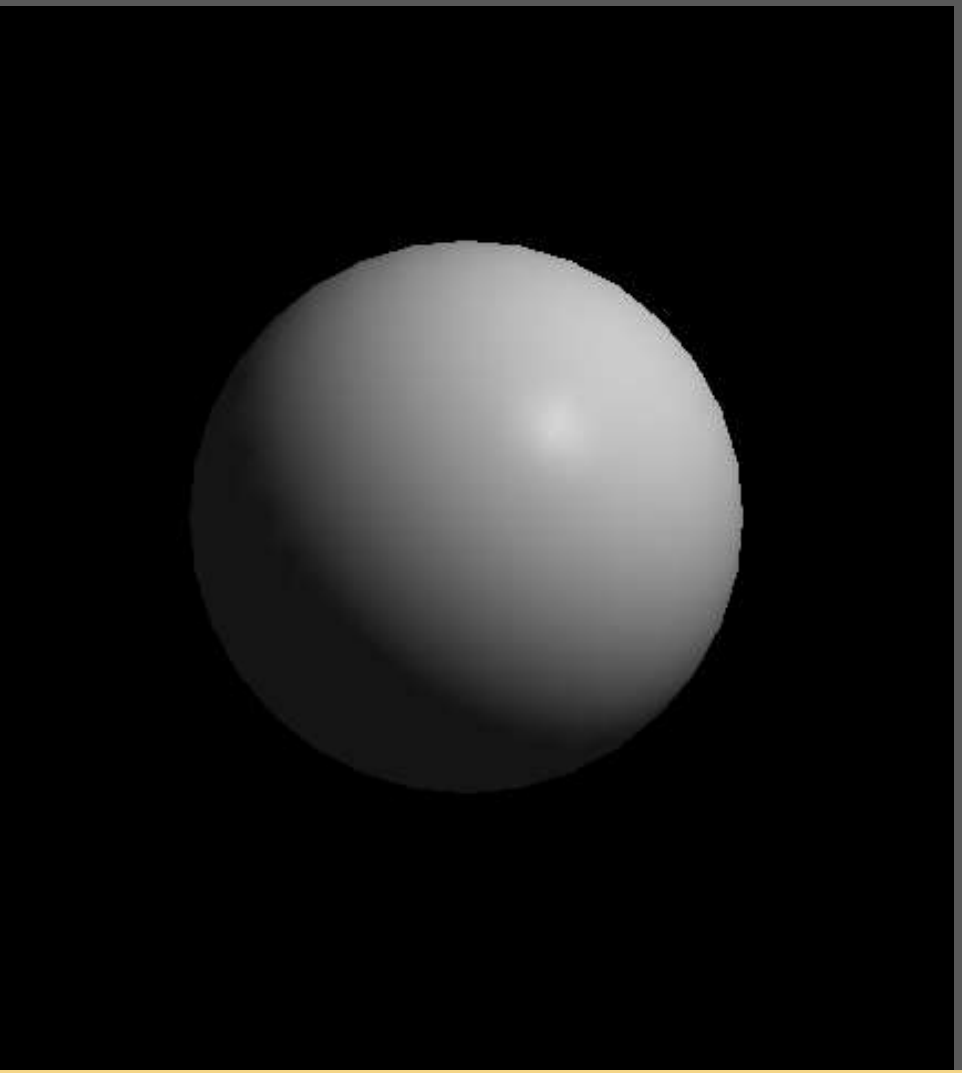
Filipe Pacheco

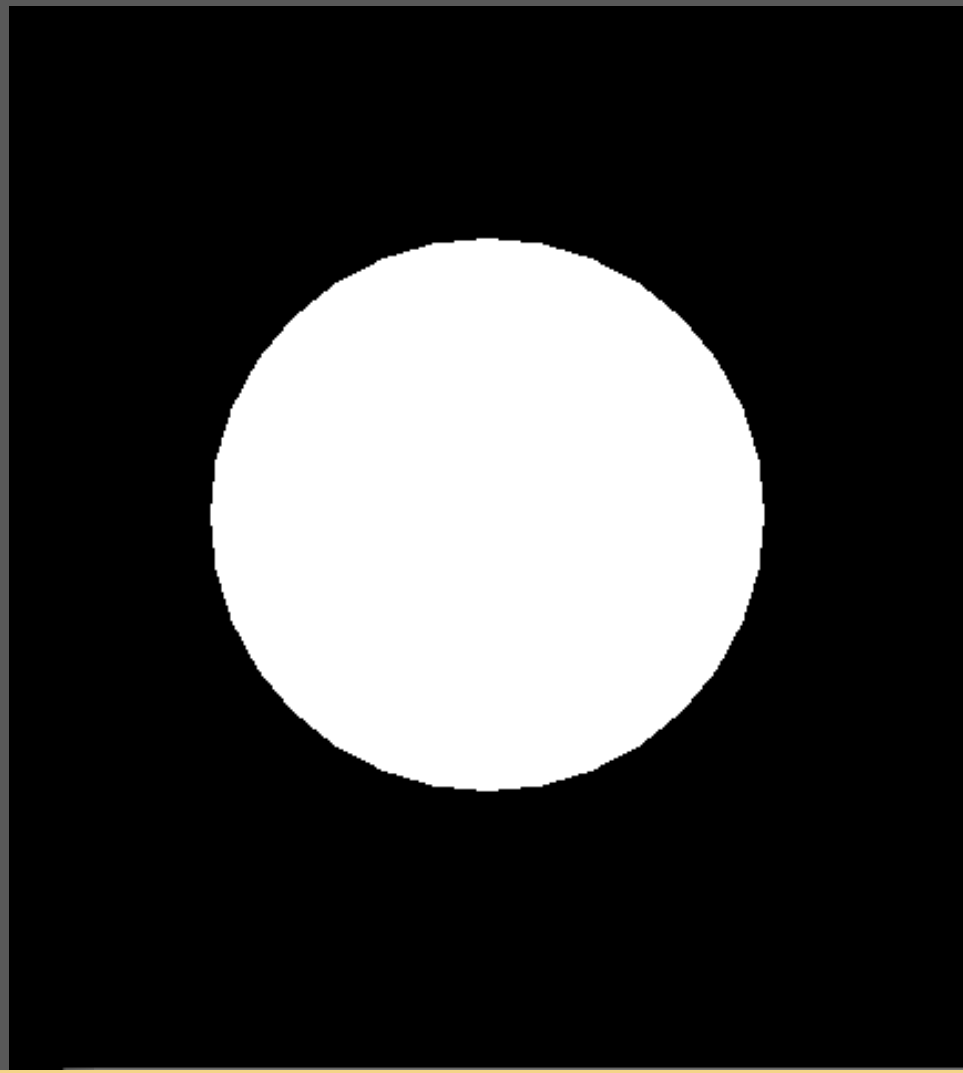
ffp@isep.ipp.pt

Iluminação

- Tipos de iluminação
- Fontes de Luz
- Modelos de iluminação
- Materiais

Esfera iluminada ou não



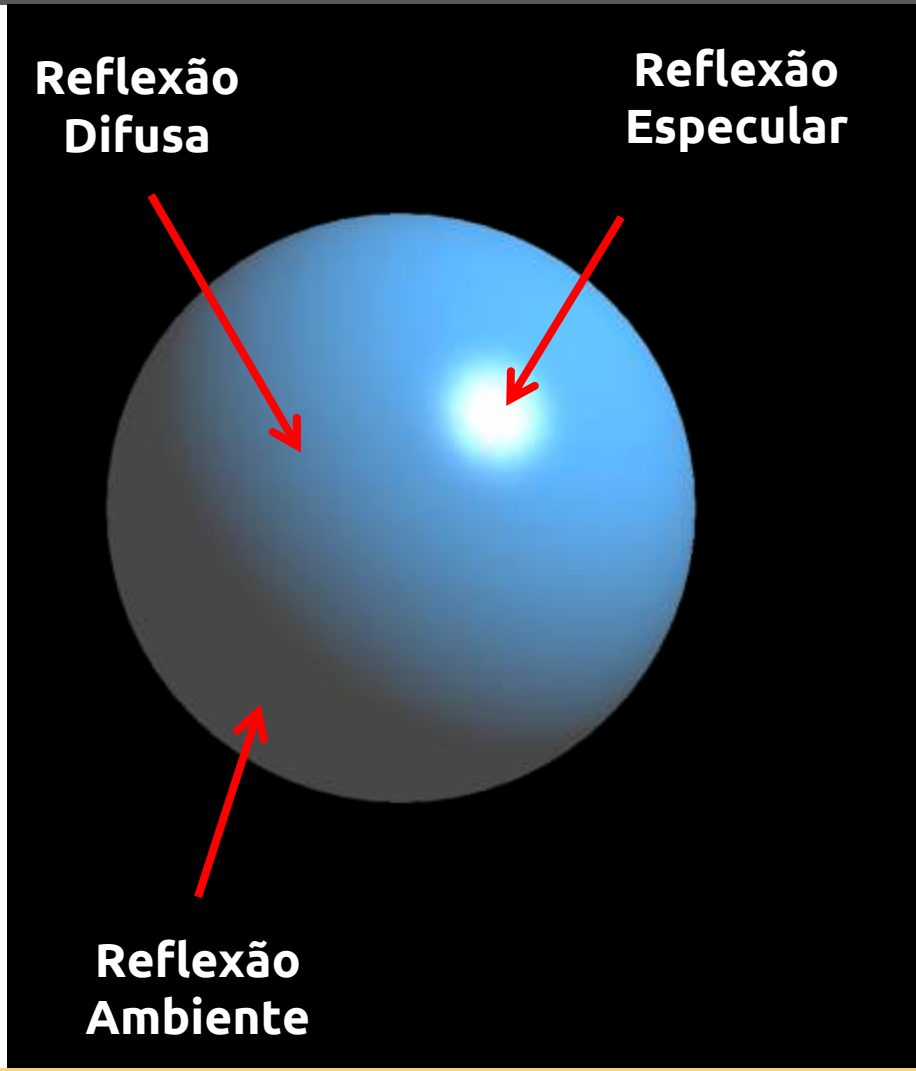


Iluminação

- OpenGL aproxima luz do mundo real em componentes RGB
  - **Fontes de luz** — emitem luz
  - **Objetos (materiais)** — refletem luz
    - Espalhando-a genericamente
    - Espalhando-a numa direção preferencial
- Luz de uma cena é proveniente de várias fontes de luz
  - Posicionais, direcionais ou ambiente

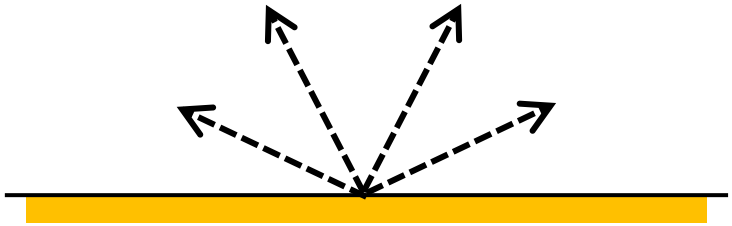
Tipos de reflexão

- **Ambiente**
  Luz espalhada uniformemente em todas as direções; resulta da luz a bater e ser refletida em superfícies
- **Difusa**
  Luz vinda de uma determinada direção; ao bater numa superfície a luz é espalhada uniformemente
- **Especular**
  Luz vinda de uma determinada direção; ao bater numa superfície a luz é refletida numa direção especifica

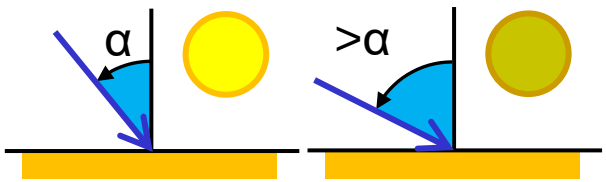


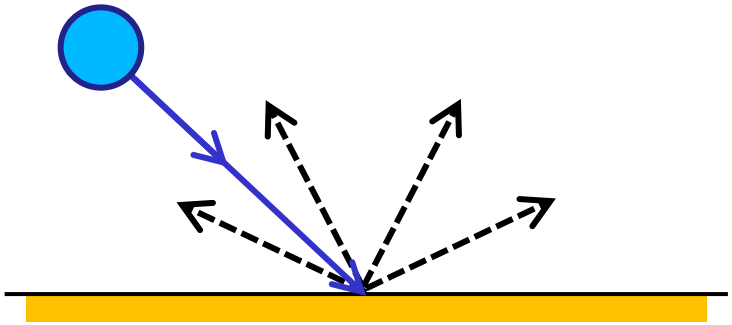
Tipos de reflexão

- **Ambiente**

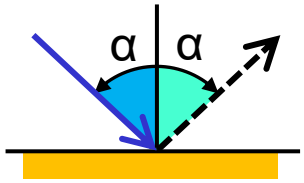


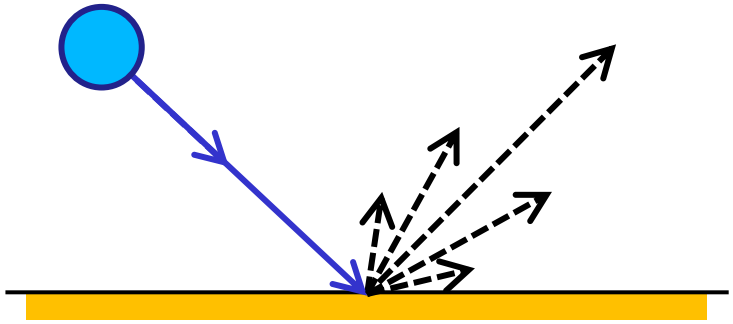
- **Difusa**





- **Especular**





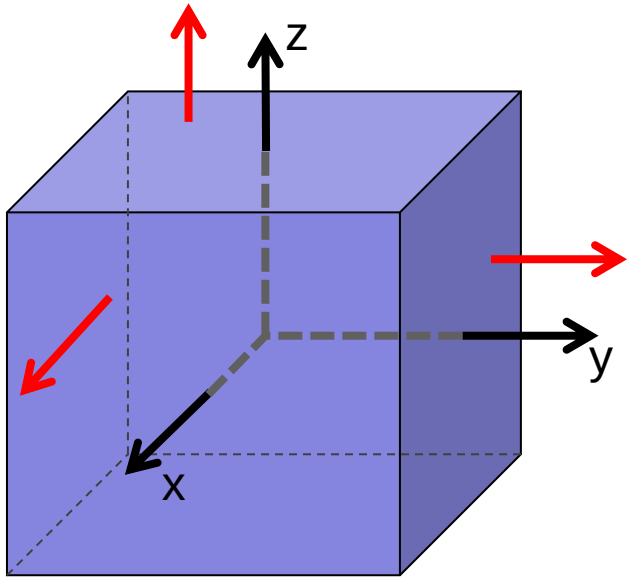
Iluminação em OpenGL

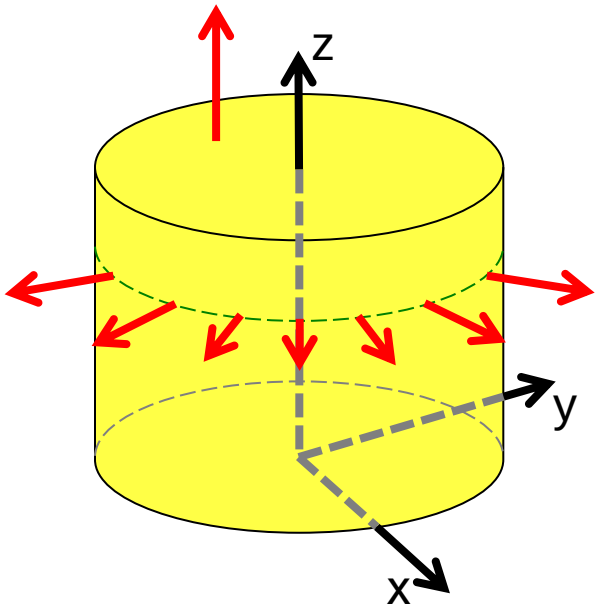
- Passos necessários:
  1. Configurar e posicionar uma ou mais fontes de luz
  2. Configurar e escolher um modelo de iluminação (nível de luz ambiente e posicionamento do ponto de vista).
  3. Definir propriedades dos materiais que compõem os objetos da cena
  4. Definir normais para cada vértice (irão determinar a orientação do objeto em relação às fontes de luz)

Vetor normal

- **`glNormal(x, y, z)`**
  - Utilizado para calcular a maneira como a luz reflete na superfície do objeto
  - Define um vetor perpendicular à superfície/vértice
    - Invocado dentro de *glBegin/glEnd* antes de *glVertex*
  - Deve ter comprimento unitário
    - Dividir cada componente x, y, z pelo comprimento da normal $\sqrt{x^2 + y^2 + z^2}$
    - **`glEnable(GL_NORMALIZE)`**

Normais

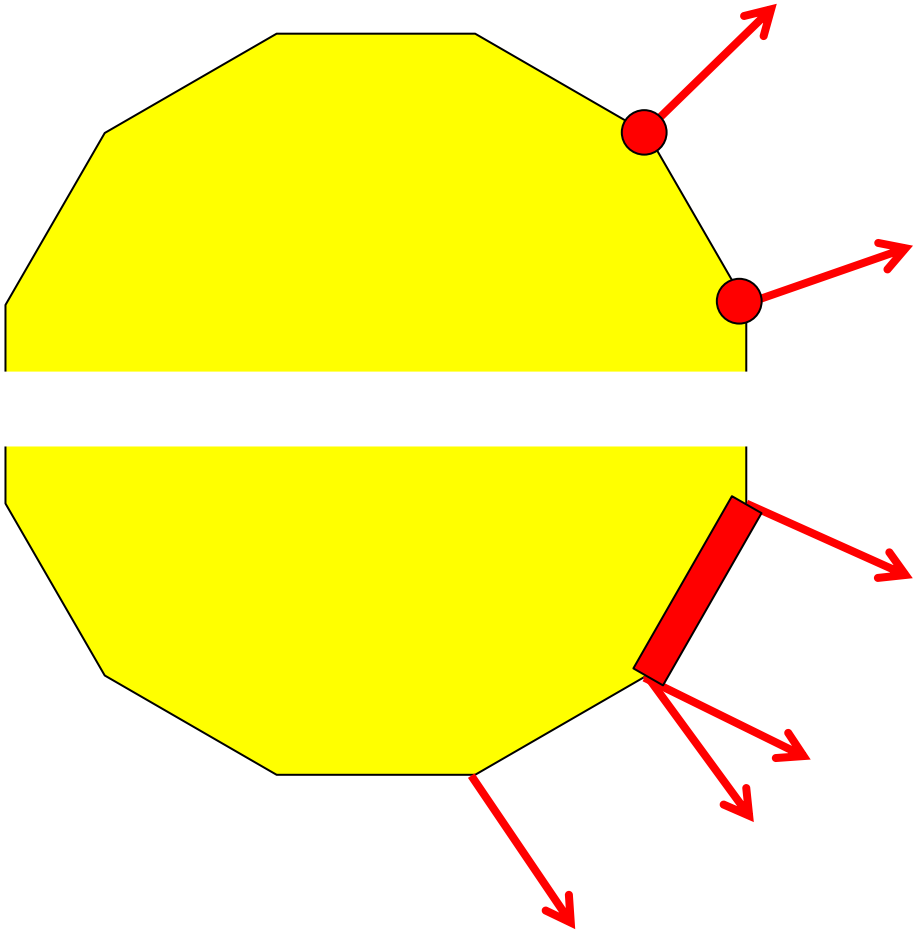




Exemplo

```
void display()
{
    ...
    glBegin(GL_POLYGON)
        glNormal3f(1.0, 0.0, 0.0);
        glVertex3f(0, 1, 0);
        glVertex3f(0, 0, 0);
        glVertex3f(0, 0, -1);
        glVertex3f(0, 1, -1);
    glEnd();
    ...
}
```

Normais



Luzes

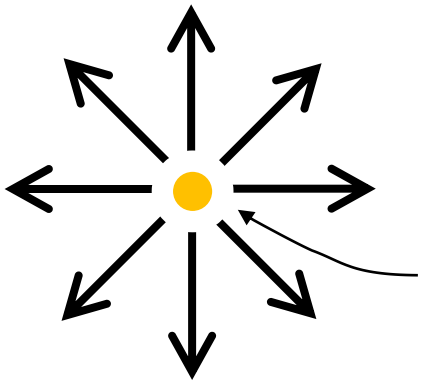
- glLightfv (luz, parâmetro, valor)
  - Luz
    - GL_LIGHT0 .. GL_LIGHT7
  - Parâmetro
    - GL_AMBIENT
    - GL_DIFFUSE
    - GL_SPECULAR
    - GL_POSITION
    - ...

Parâmetro de glLight

Parameter Name	Default Value	Meaning
GL_AMBIENT	(0.0, 0.0, 0.0, 1.0)	ambient RGBA intensity of light
GL_DIFFUSE	(1.0, 1.0, 1.0, 1.0) – Luz 0 **(0.0, 0.0, 0.0, 1.0) – Luz 1 .. 7**	diffuse RGBA intensity of light
GL_SPECULAR	(1.0, 1.0, 1.0, 1.0) – Luz 0 **(0.0, 0.0, 0.0, 1.0) – Luz 1 .. 7**	specular RGBA intensity of light
GL_POSITION	(0.0, 0.0, 1.0, 0.0)	(x, y, z, w) position of light
GL_SPOT_DIRECTION	(0.0, 0.0, -1.0)	(x, y, z) direction of spotlight
GL_SPOT_EXPONENT	0.0	spotlight exponent
GL_SPOT_CUTOFF	180.0	spotlight cutoff angle
GL_LINEAR_ATTENUATION	0.0	linear attenuation factor
GL_QUADRATIC_ATTENUATION	0.0	quadratic attenuation factor

GL_POSITION

- Luz posicional
  - (x, y, z, w)
  - w ≠ 0 ➔ posicional
    - *Guardada em Eye coordinates (movendo a câmara a luz "vai junto")*
    - *Atenuação ativada.*

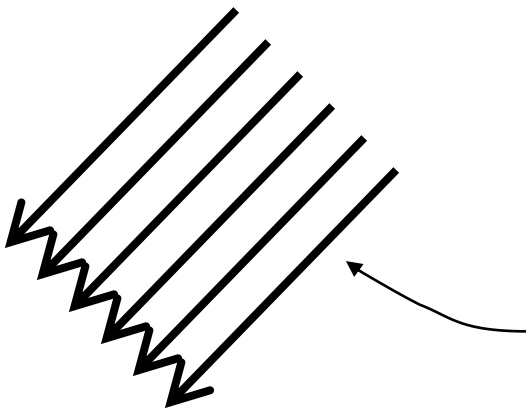


Posição: (x, y, z)

(x, y, z, 1.0)

GL_POSITION

- Luz direcional
  - (x, y, z, w)
  - w = 0 ➔ direcional
    - *Apenas indicamos a direção da luz, posição é irrelevante*
    - *Atenuação desativada*

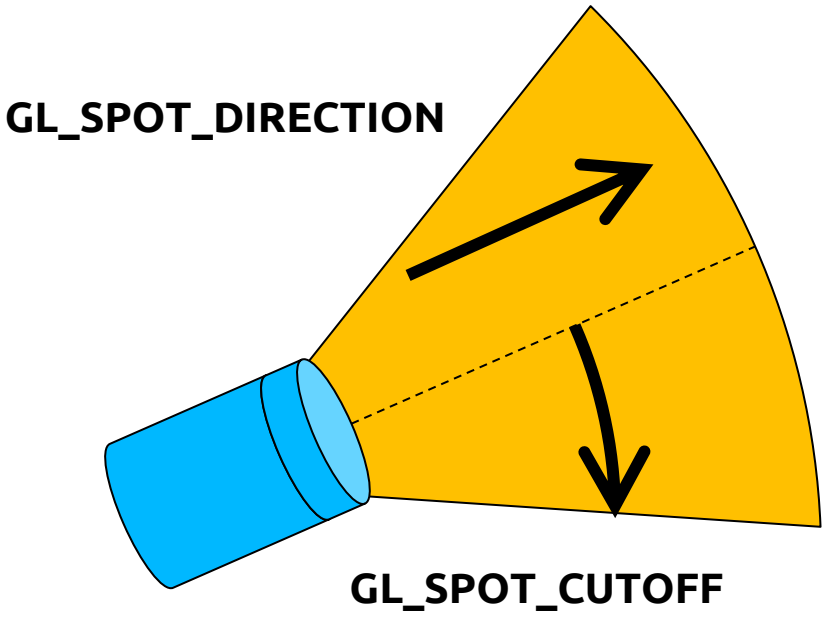


(x, y, z, 0.0)

Direção: (x, y, z)

Focos

- Por omissão uma luz emite em todas as direções
- É possível definir um foco indicando a direção e a abertura



- É possível definir a "concentração" de luz no cone usando **GL_SPOT_EXPONENT**

Exemplo

```
void init()
{
    GLfloat light_ambient[] = { 0.0, 0.0, 0.0, 1.0 };
    GLfloat light_diffuse[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat light_specular[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat light_position[] = { 1.0, 1.0, 1.0, 0.0 };

    ...
    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);
    ...
}
```

Ligar/desligar iluminação

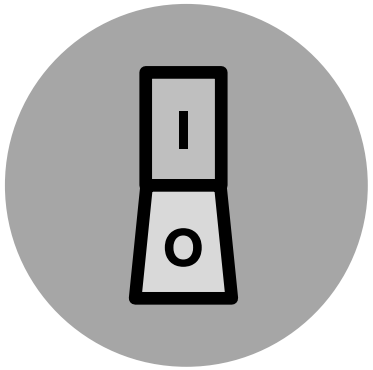
- glEnable(GL_LIGHTING)
  glDisable(GL_LIGHTING)

  Ativa/Desativa a iluminação, é semelhante a um "interruptor geral" de uma casa

- glEnable(GL_LIGHT*n*)
  glDisable(GL_LIGHT*n*)

  Liga e desliga uma fonte de luz, é semelhante ao interruptor de uma lâmpada ou de um candeeiro

  n é um inteiro entre 0 e 7



Modelo de iluminação

- glLightModel(parâmetro, valor)

  Parâmetro:
  - GL_LIGHT_MODEL_AMBIENT
    - Luz ambiente existente na cena
      Default: (0.2, 0.2, 0.2, 1.0)
  - GL_LIGHT_MODEL_LOCAL_VIEWER
    - Controla o cálculo das reflexões
      Default: GL_FALSE
  - GL_LIGHT_MODEL_TWO_SIDE
    - Controla o tratamento da iluminação para as duas faces dos polígonos
      Default: GL_FALSE

Materiais

- glMaterial(face, parâmetro, valor)
  - Face
    - GL_FRONT
    - GL_BACK
    - GL_FRONT_AND_BACK
  - Parâmetro
    - GL_AMBIENT
    - GL_DIFFUSE
    - GL_SPECULAR
    - ...

Parâmetro de glMaterial

Parameter Name	Default Value	Meaning
GL_AMBIENT	(0.2, 0.2, 0.2, 1.0)	ambient color of material
GL_DIFFUSE	(0.8, 0.8, 0.8, 1.0)	diffuse color of material
GL_AMBIENT_AND_DIFFUSE		ambient and diffuse color of material
GL_SPECULAR	(0.0, 0.0, 0.0, 1.0)	specular color of material
GL_SHININESS	0.0	specular exponent
GL_EMISSION	(0.0, 0.0, 0.0, 1.0)	emissive color of material
GL_COLOR_INDEXES	(0,1,1)	ambient, diffuse, and specular color indices

Materiais

- Possuem componente ambiente, difusa, especular e emissora
  - Ambiente e difusa definem a cor do objeto e são normalmente iguais
  - Especular é normalmente branco para garantir a cor das fontes de luz
  - Emissora simula uma fonte de luz dentro do próprio objeto

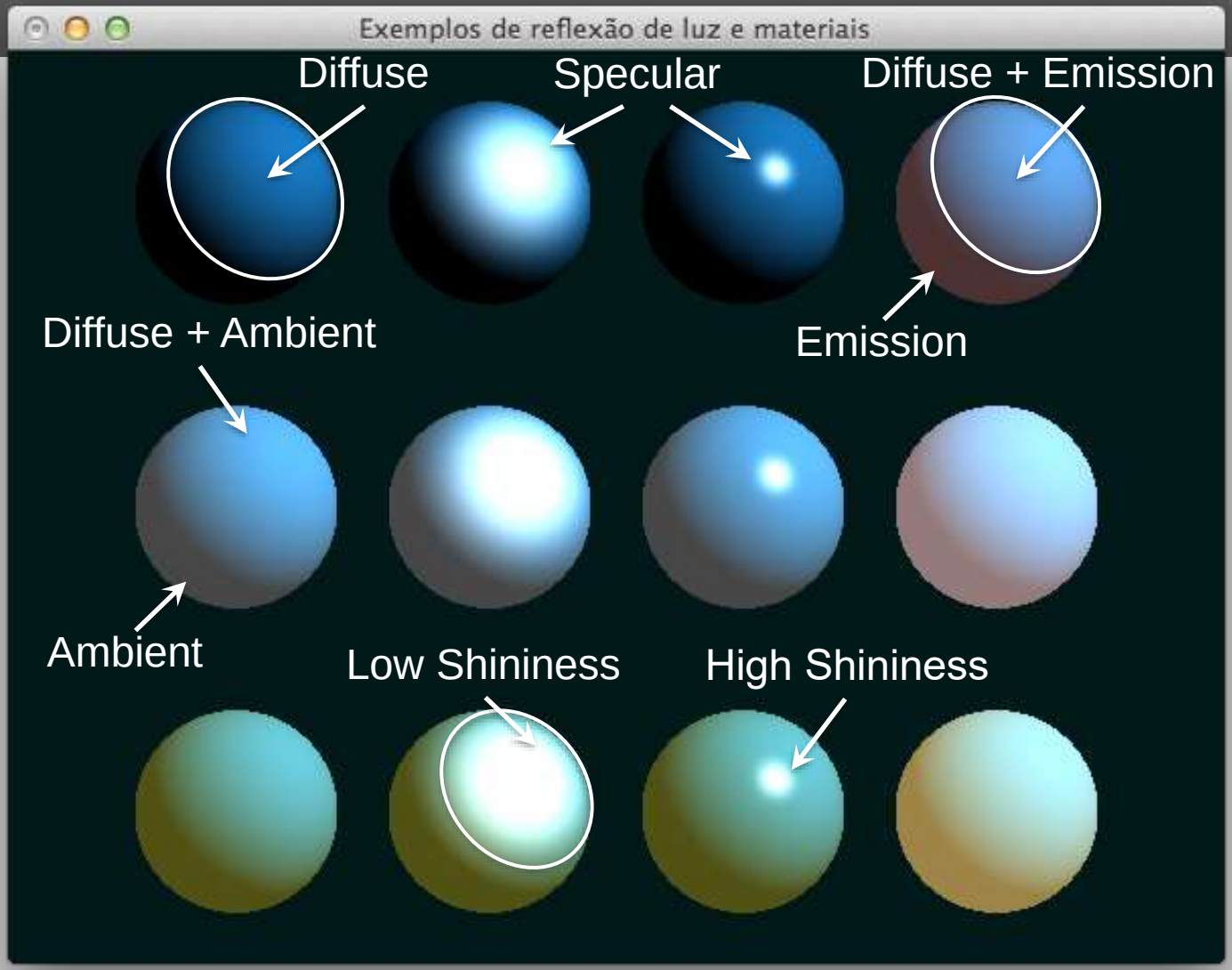
Materiais

- Percentagem de reflexão de cada componente de cor RGB
  - Exemplo:
    - R = 1.0, G= 0.5, B = 0.0
    - Reflete **todo** o vermelho
    - Reflete 50% do verde
    - **Absorve todo** o azul
  - Ou seja:
    - Fonte de Luz emite (LR, LG, LB)
    - Material reflete (MR, MG, MB)
    - “Cor” visível = (LR*MR, LG*MG, LB*MB)

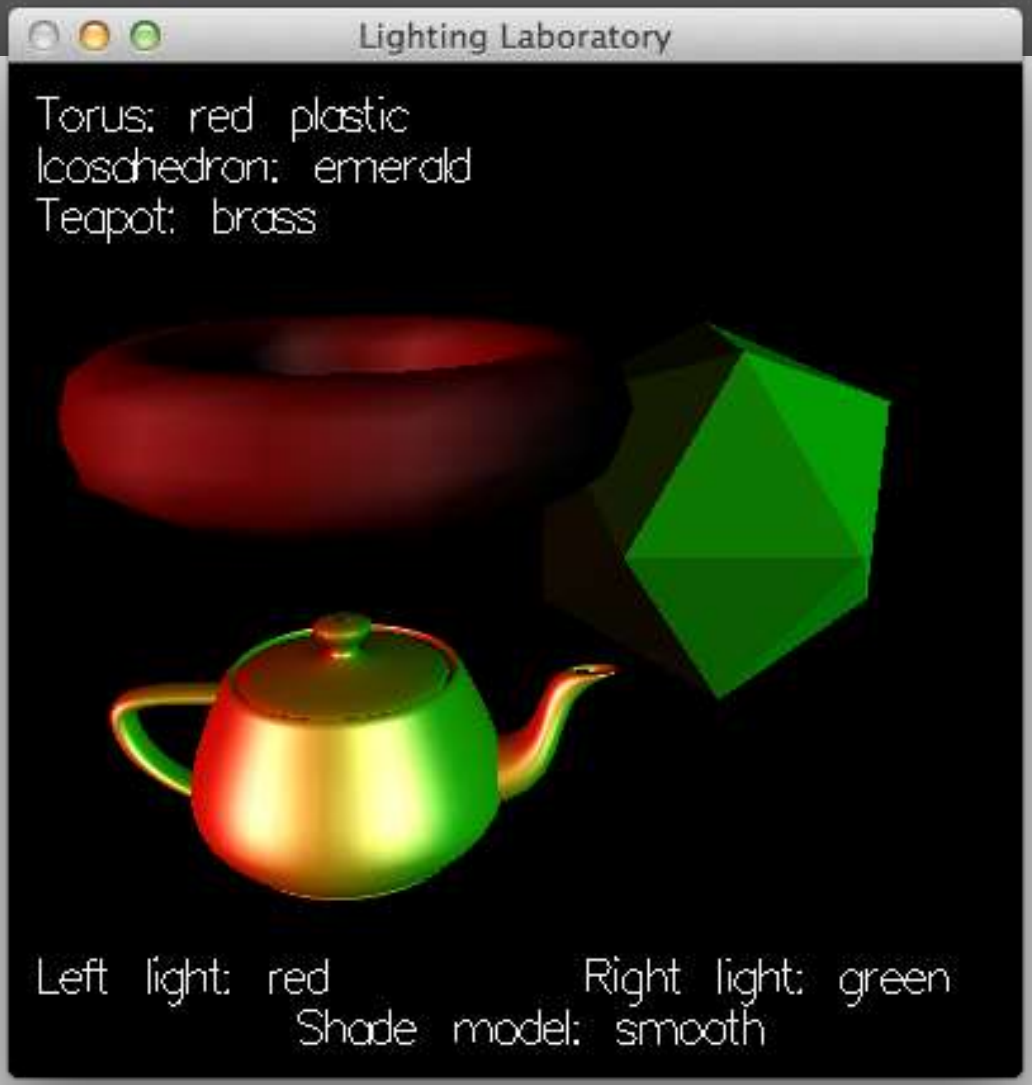
Materiais: exemplo

- Objeto vermelho {1.0, 0.0, 0.0}
  - Com fonte de Luz vermelha
    {1.0, 0.0, 0.0}
    ➔ objeto vermelho {1.0, 0.0, 0.0}
  - Com fonte de Luz branca
    {1.0, 1.0, 1.0}
    ➔ objeto vermelho {1.0, 0.0, 0.0}
    - Só a componente vermelha da luz branca é refletida
  - Com fonte de Luz verde
    {0.0,1.0,0.0}
    ➔ objeto preto {0.0, 0.0, 0.0}
    - Todo o verde é absorvido

Demo


Exemplos de reflexão de luz e materiais
Diffuse
Specular
Diffuse + Emission
Diffuse + Ambient
Emission
Ambient
Low Shininess
High Shininess

Light lab


Lighting Laboratory
Torus: red plastic
Icosahedron: emerald
Teapot: brass
Left light: red
Right light: green
Shade model: smooth

Posicionar luzes na cena

- Em OpenGL a posição de uma fonte de luz (posicional) é tratada como uma primitiva, sendo por isso transformada pela matriz de modelo/vista
- Luz fixa (mundo)
  - Definir posição da luz após as transformações de vista
- Luz móvel (mundo)
  - Aplicar transformação antes de definir posição da luz
- Luz que acompanha o ponto de vista (câmara)
  - Definir posição da luz **antes** de qualquer transformação (i.e., a seguir a ***glLoadIdentity***)
  - Modificar o ponto de vista usando ***gluLookAt***

Luz fixa

```
void display(void)
{
    GLfloat light_position() = {0.0, 0.0, 0.0, 1.0};

    glClear(GL_COLOR_BUFFER_MASK | GL_DEPTH_BUFFER_MASK);
    glLoadIdentity();

    gluLookAt (ex, ey, ez, 0.0, 0.0, 0.0, upx, upy, upz);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);

    glutSolidTorus (0.275, 0.85, 8, 15);
    glFlush();
}
```

Luz móvel

```
static GLdouble spin;

void display(void)
{
  GLfloat light_position[] = { 0.0, 0.0, 1.5, 1.0 };
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glPushMatrix();
    // viewing transformation
  gluLookAt(0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

    // light position
  glPushMatrix();
      glRotated(spin, 1.0, 0.0, 0.0);
      glLightfv(GL_LIGHT0, GL_POSITION, light_position);
  glPopMatrix();

  // draw world

  ...
}
```

Demo


/Users/matos/Documents/cadeiras/SGRAI/2012-13/Aulas TP/S...
Welcome to movelight.
Right mouse button for menu.
Hold down the left mouse button
and move the mouse horizontally
to change the light position.

Luz no ponto de vista

```
void reshape (int w, int h)
{
  glViewport(0, 0, (GLint) w, (GLint) h);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  gluPerspective(40.0, (GLfloat) w/(GLfloat) h, 1.0, 100.0);
  glMatrixMode(GL_MODELVIEW);
  glLoadIdentity();
}
```

Luz no ponto de vista

```
void display(void)
{
    GLfloat light_position() = {0.0, 0.0, 0.0, 1.0};

    glClear(GL_COLOR_BUFFER_MASK | GL_DEPTH_BUFFER_MASK);
    glLoadIdentity();

    glLightfv(GL_LIGHT0, GL_POSITION, light_position);
    gluLookAt (ex, ey, ez, 0.0, 0.0, 0.0, upx, upy, upz);

    glutSolidTorus (0.275, 0.85, 8, 15);
    glFlush();
}
```