Módulo 10

Sistemas Gráficos e Interação

Instituto Superior de Engenharia do Porto

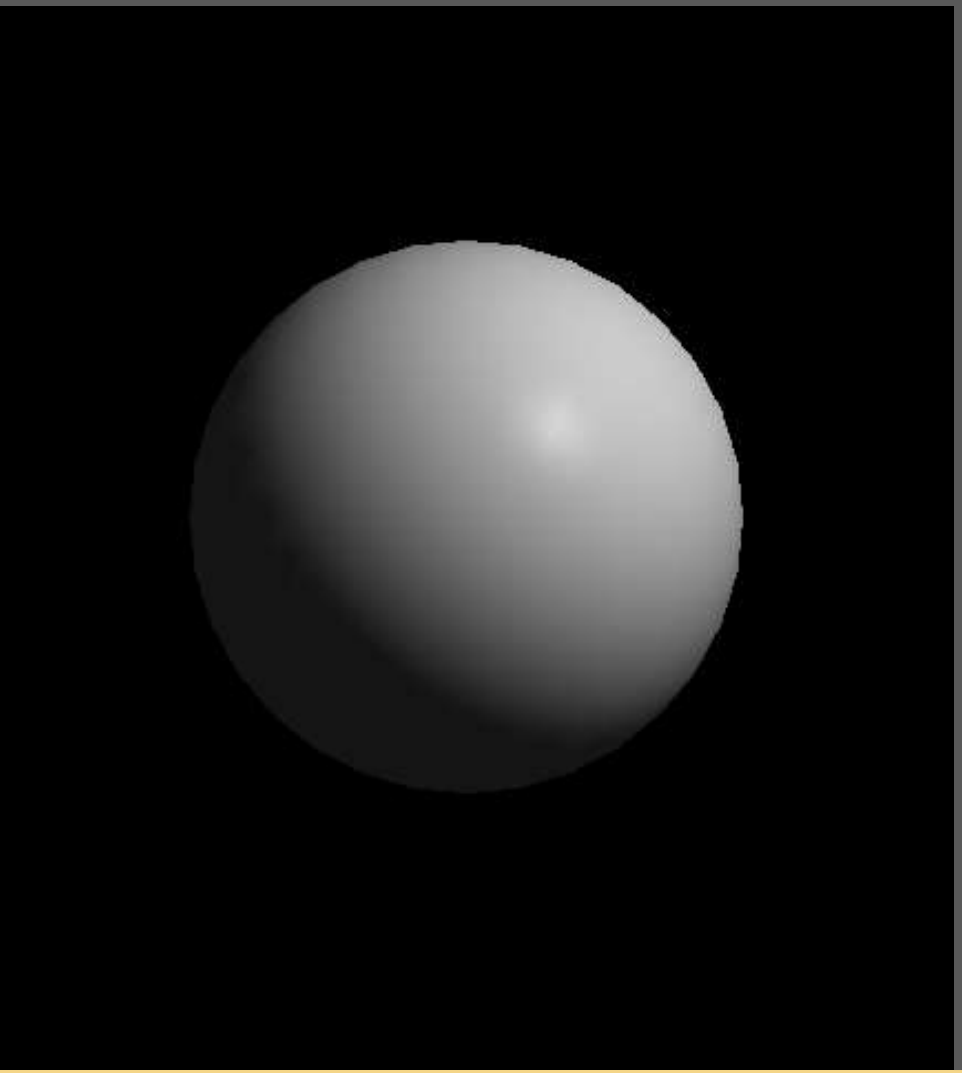
Filipe Pacheco

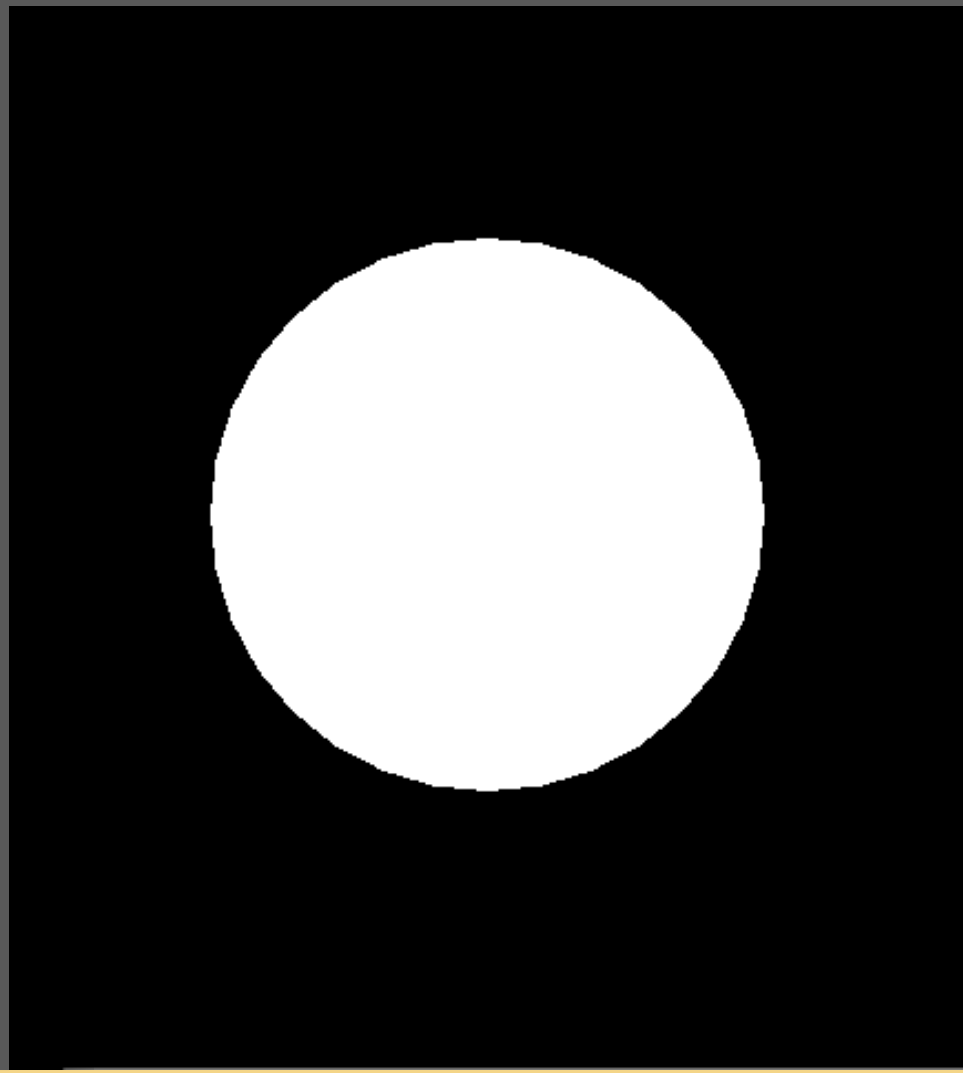
ffp@isep.ipp.pt

Iluminação

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

Sphere illuminated or not



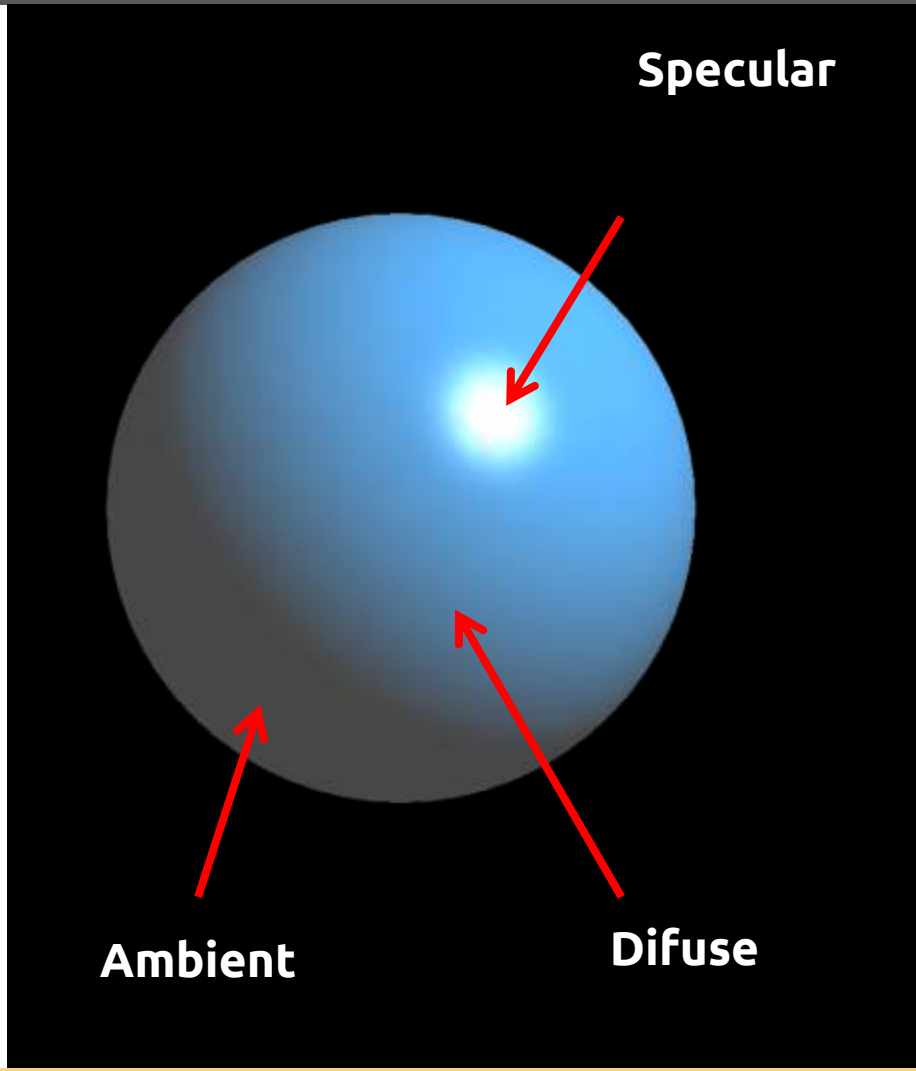


Iluminação

- OpenGL adapts real-world light to RGB components
- Light sources - emit light
- Objects (materials) - reflect light
  - Spreading it generically
  - Spreading it in a preferred direction
- A scene's light comes from various light sources
  - Positional, directional or environment

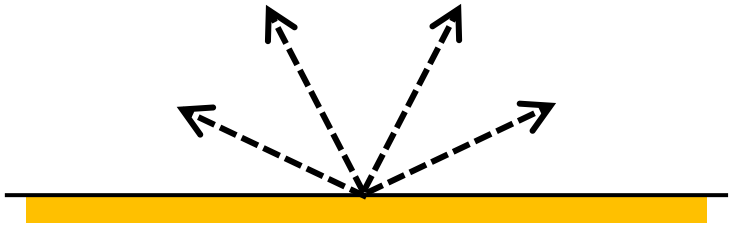
Tipos de reflexão

- **Ambient**
  Light spread evenly in all directions; results from light hitting and being reflected on surfaces
- **Diffuse**
  Light coming from a certain direction; when hitting a surface the light is spread evenly
- **Specular**
  Light coming from a certain direction; when hitting a surface the light is reflected in a specific direction

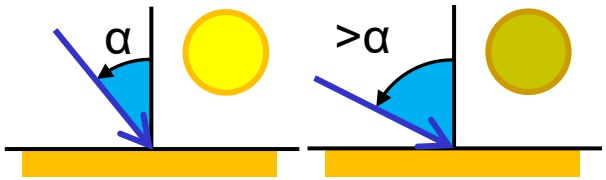


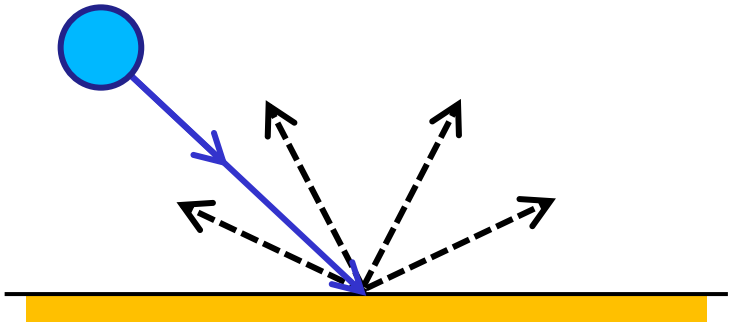
Tipos de reflexão

- **Ambient**

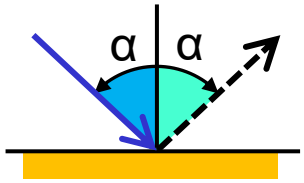


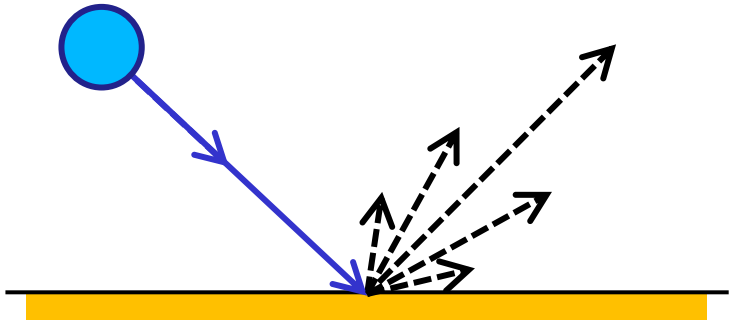
- **Diffuse**





- **Specular**





Iluminação em OpenGL

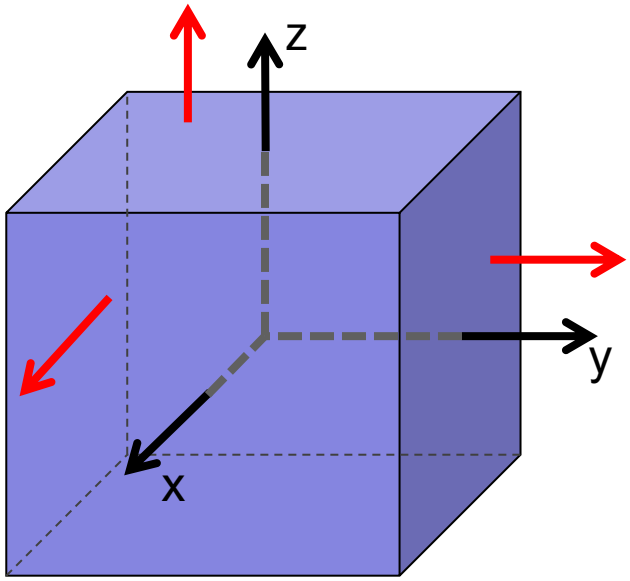
- Steps:
  1. Configure and position one or more light sources
  2. Configure and choose a lighting model (ambient light level and positioning of the point of view).
  3. Define material properties that make up scene objects
  4. Set normals for each vertex (will determine the object's orientation in relation to light sources)

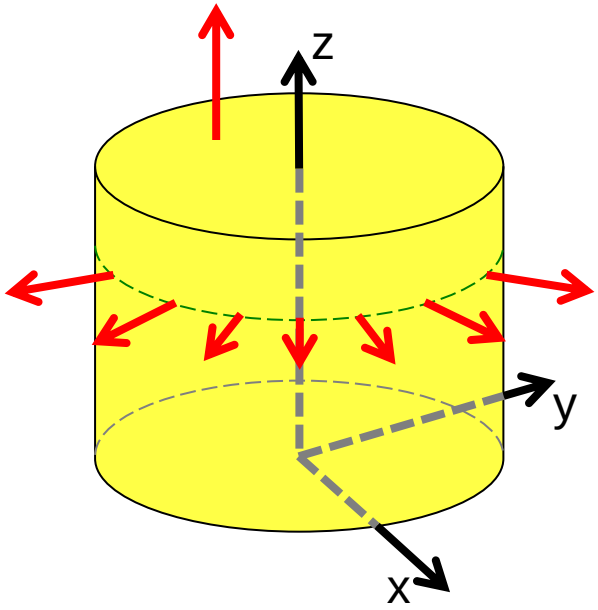
Vetor normal

- **`glNormal(x, y, z)`**
  - Used to calculate the way light reflects off the object's surface
  - Defines a vector perpendicular to the surface/vertex
  - Invoked inside glBegin/glEnd before glVertex
  - Must have unit length
    - Divide each component x, y, z by the length of the normal
    - Or use glEnable (GL_NORMALIZE)

$$\sqrt{x^2 + y^2 + z^2}$$

Normais

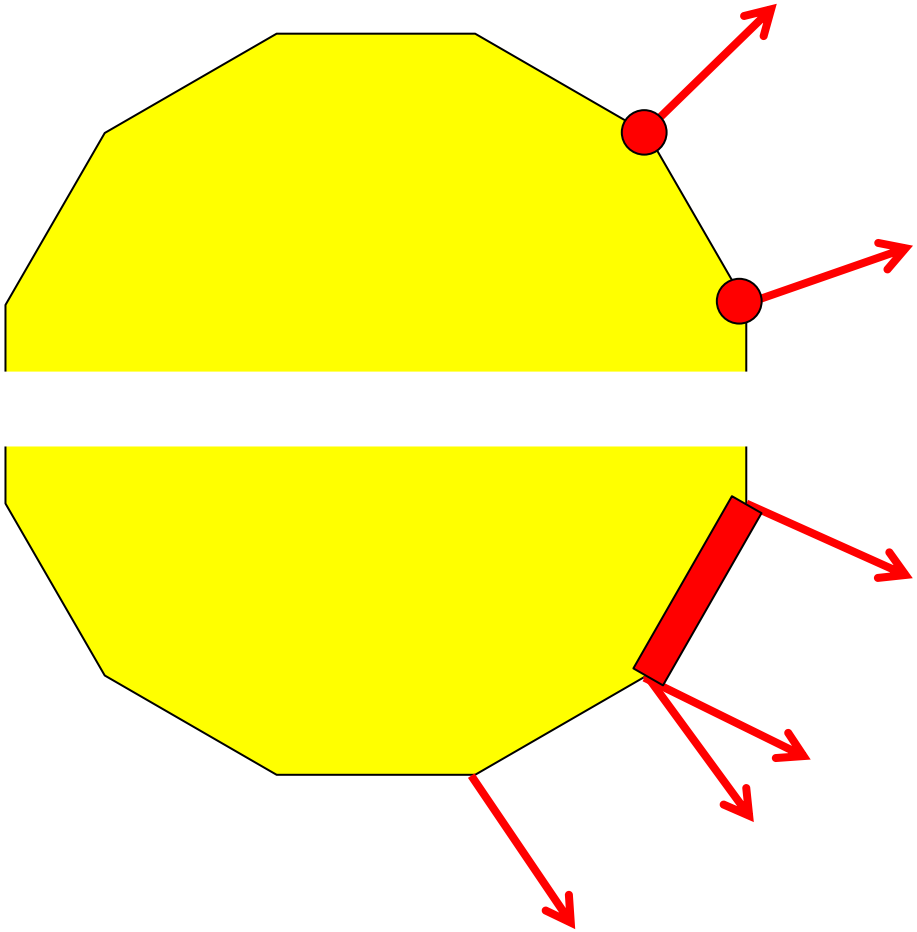




Exemplo

```
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

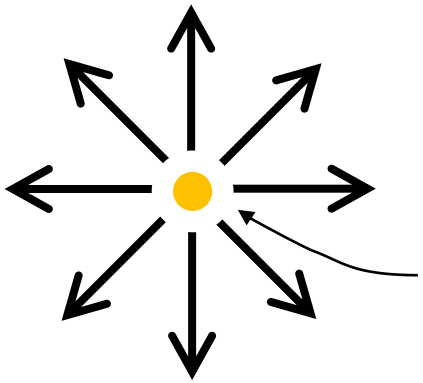
- glLightfv (luz, parâmetro, valor)
  - Luz
    - GL_LIGHT0 .. GL_LIGHT7
  - Parâmetro
    - GL_AMBIENT
    - GL_DIFFUSE
    - GL_SPECULAR
    - GL_POSITION
    - ...

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

- Positional light
  - (x, y, z, w)
  - w ≠ 0 ➔ posicional
    - *Saved in Eye coordinates (moving the camera, the light "goes along")*
    - *Attenuation enabled.*

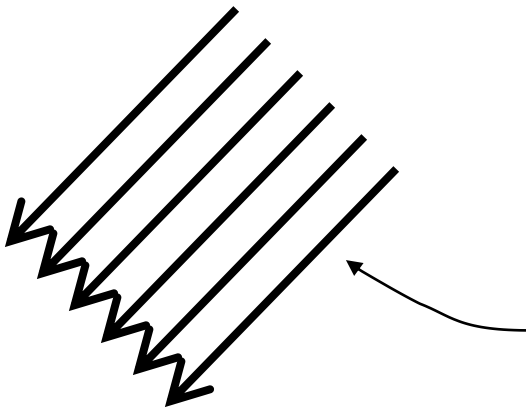


Position: (x, y, z)

(x, y, z, 1.0)

GL_POSITION

- Directional light
  - (x, y, z, w)
  - w = 0 ➔ directional
    - *We only indicate the direction of the light, position is irrelevant*
    - *Attenuation is disabled*

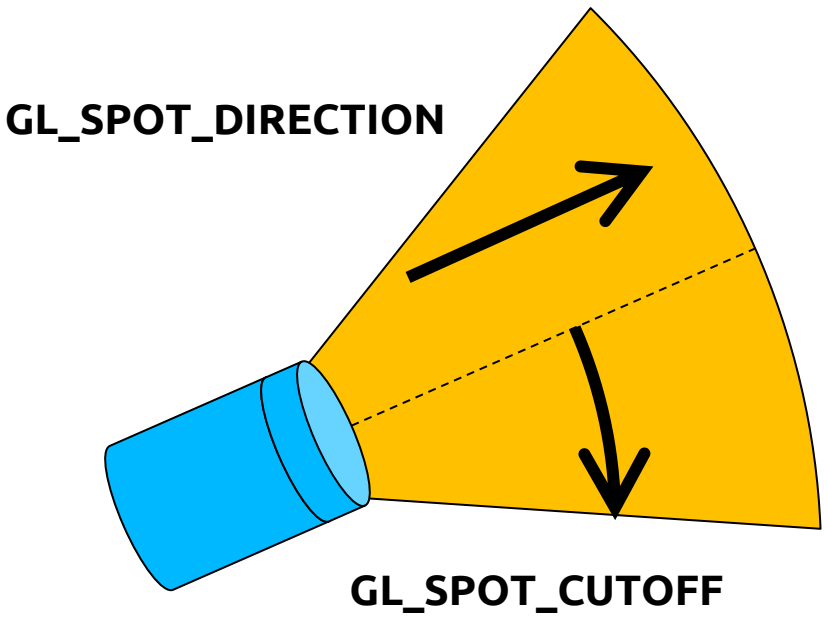


(x, y, z, 0.0)

Direction: (x, y, z)

Focos

- By default a light radiates in all directions
- It is possible to define a focus indicating the direction and aperture



- It is possible to define the “concentration” of light in the cone using **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

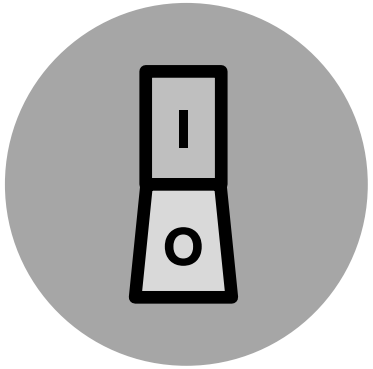
- glEnable(GL_LIGHTING)
  glDisable(GL_LIGHTING)

  Enables / disables the lighting, it is similar to a "general switch" of a house

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

  Turns a light source on and off, similar to the switch on a lamp

  n is an integer between 0 and 7



Modelo de iluminação

- glLightModel(parâmetro, valor)

  Parâmetro:
  - GL_LIGHT_MODEL_AMBIENT
    - Ambient light in the scene
      Default: (0.2, 0.2, 0.2, 1.0)
  - GL_LIGHT_MODEL_LOCAL_VIEWER
    - Reflection calculation mode
      Default: GL_FALSE
  - GL_LIGHT_MODEL_TWO_SIDE
    - Controls the lighting treatment for both sides of the polygons
      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

- Have ambient, diffuse, specular and emitting component
- Ambient and diffuse define the color of the object and are usually the same
- Specular is usually white to guarantee the color of the light sources
- Emitter simulates a light source within the object itself

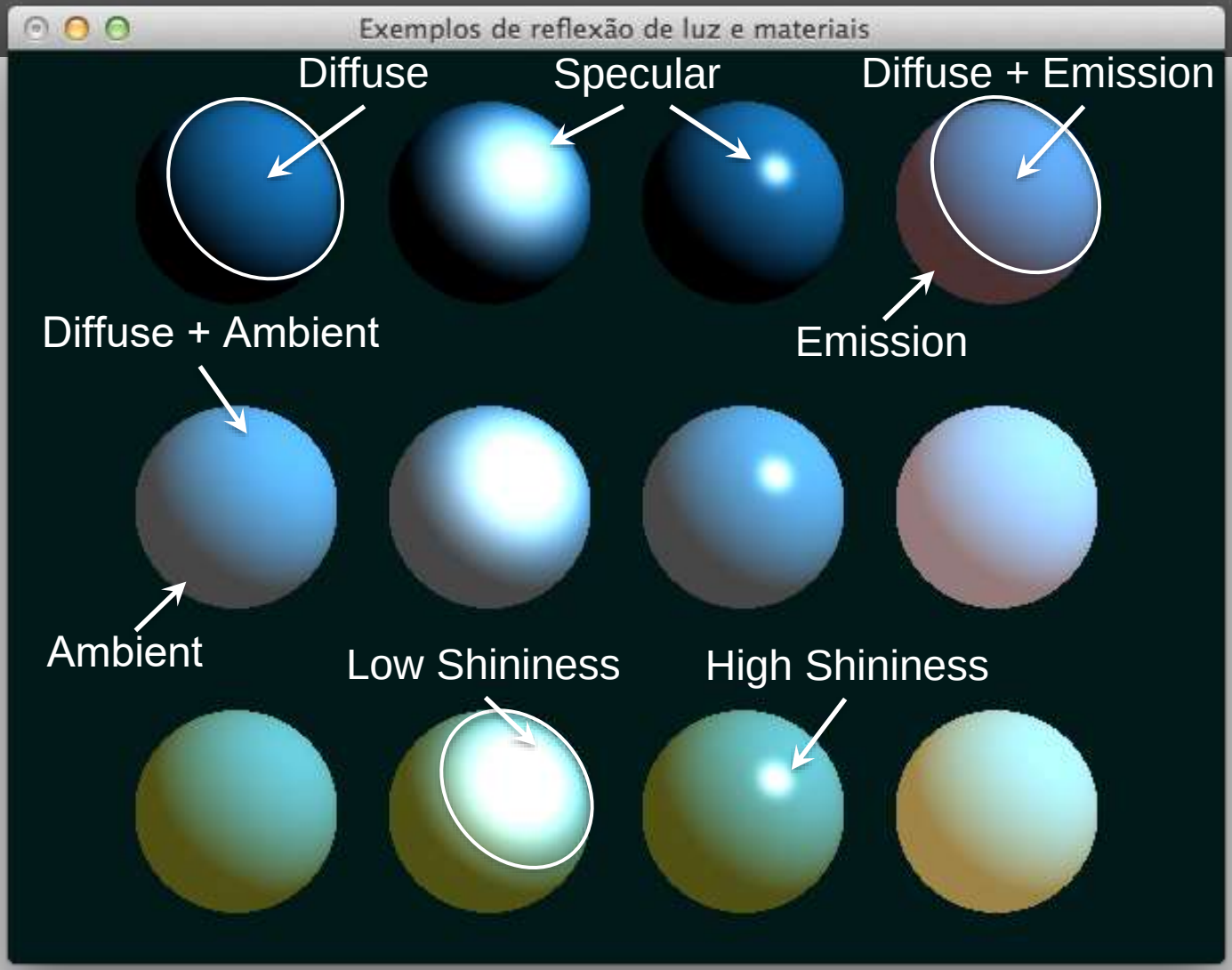
Materiais

- Reflection percentage of each RGB color component:
  - Example:
    - R = 1.0, G= 0.5, B = 0.0
    - **Reflects all** red
    - Reflects 50% of green
    - **Absorbs all** blue
  - Generic:
    - Light source emits (LR, LG, LB)
    - Material reflects (MR, MG, MB)
    - Visible “Color” = (LR*MR, LG*MG, LB*MB)

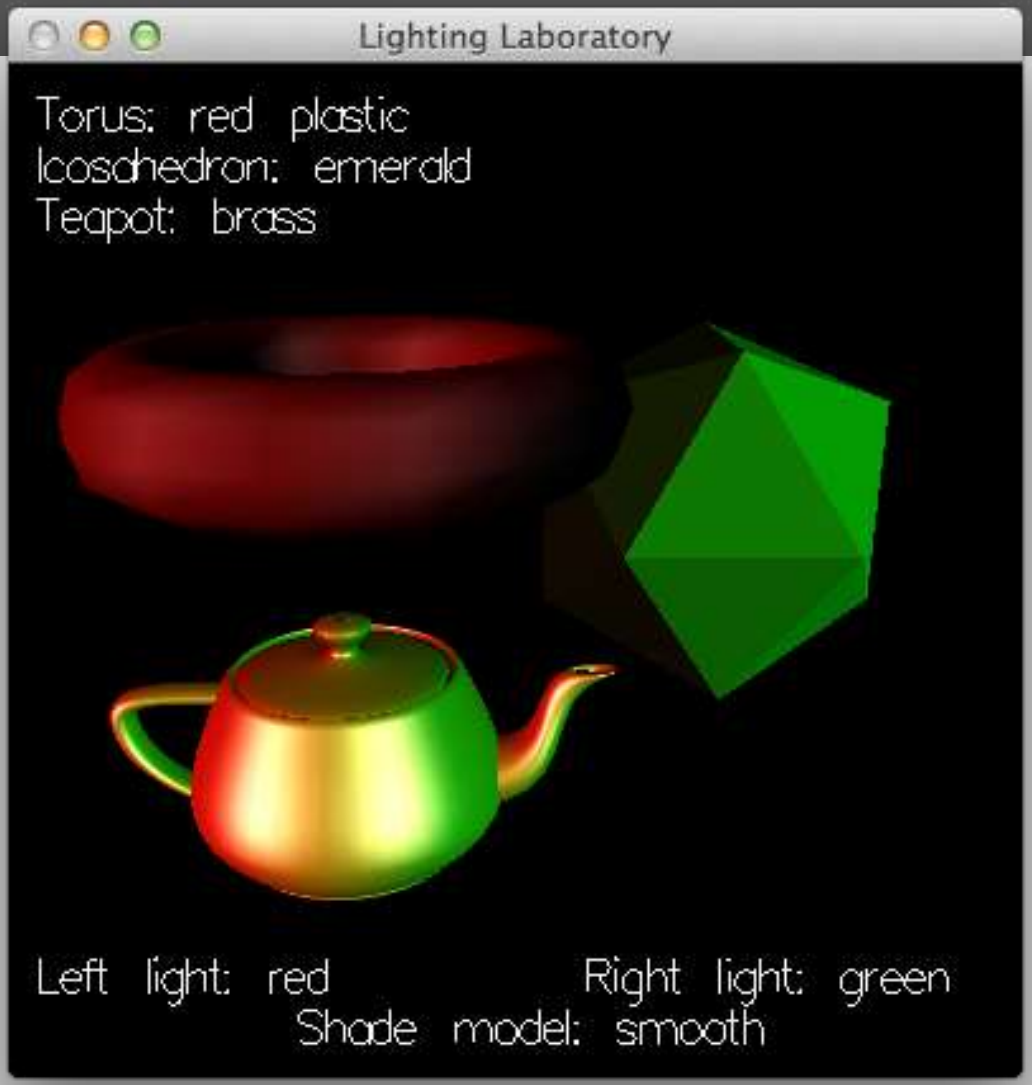
Materiais: exemplo

- 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

- In OpenGL the position of a light source (positional) is treated as a primitive, so it is transformed by the model/view matrix(es)
- Fixed light (world)
  - Define light position after view transformations
- Mobile light (world)
  - Apply transformation before defining light position
- Light accompanying the viewpoint (camera)
  - Define light position before any transformation (i.e., after glLoadIdentity)
  - Modify the view using gluLookAt

Fixed

```
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();
}
```

Mobile

```
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.

Point of view

```
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();
}
```

Point of view

```
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();
}
```