

# Computação Gráfica

## Iluminação

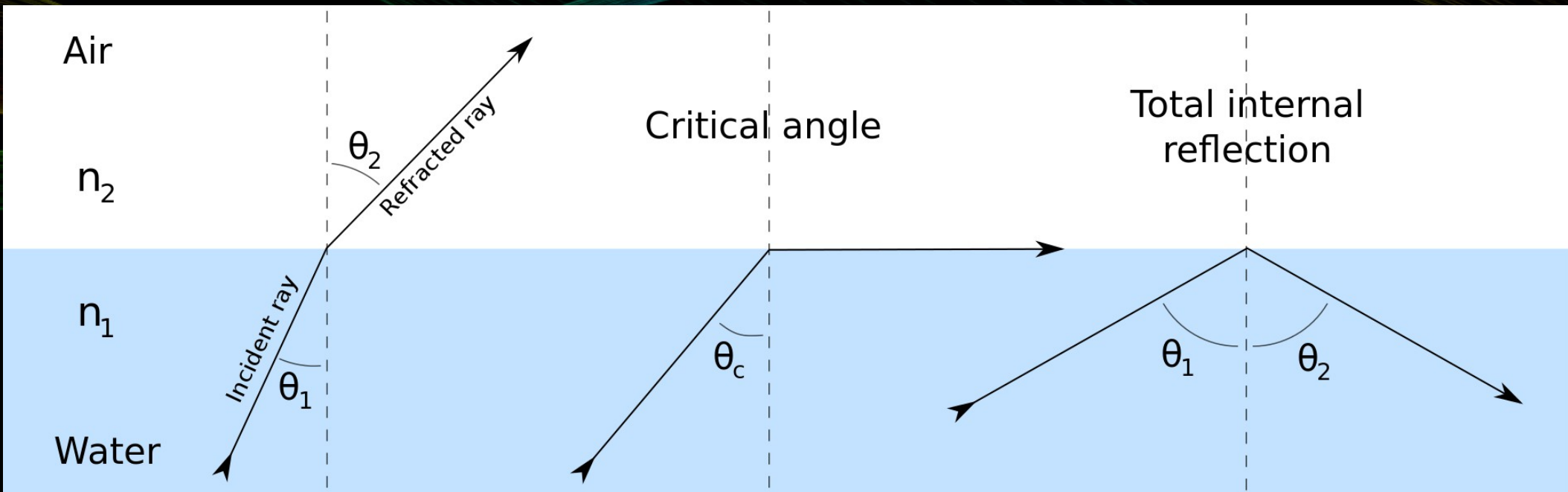
José Luis Seixas Junior

# Índice

- Reflexão;
- Shadings;
- Fontes;
- OpenGL;



# Reflexão



# Reflexão

- Alteração da propagação de energia, com a diferença do ângulo de incidência sobre a normal do objeto;
- Onde a normal é a perpendicular da tangente no ponto de incidência;
- Assim, o ângulo de saída da propagação é o mesmo, mas com flip na perpendicular da emissão;



# Reflexão

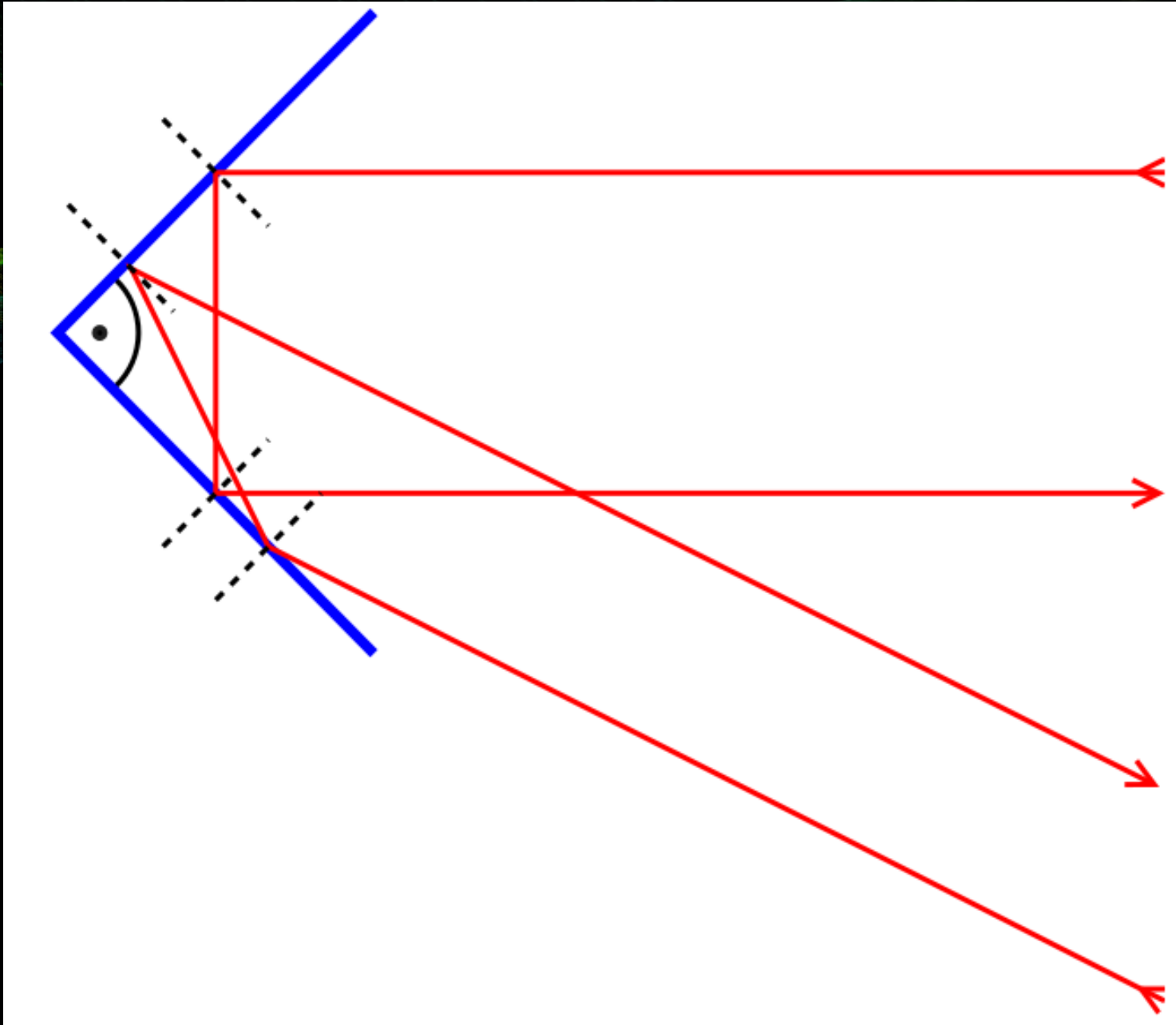
- Vídeo recomendado:
  - Physics Girls:
  - Why do mirrors flip horizontally (but not vertically)?
  - <https://www.youtube.com/watch?v=vBpxhfBIVLU>

# Reflexão

- Quando a luz incide sobre uma superfície opaca, parte dela é absorvida, o restante é refletido;
- A parte refletida que chega até nossa retina e assim, podemos ver (Young década de 60);
- Modelagem de reflexão depende de material:
  - Especular ou difusa;
  - Micro-estrutura;
    - Tudo que é maior que o comprimento de onda da propagação;

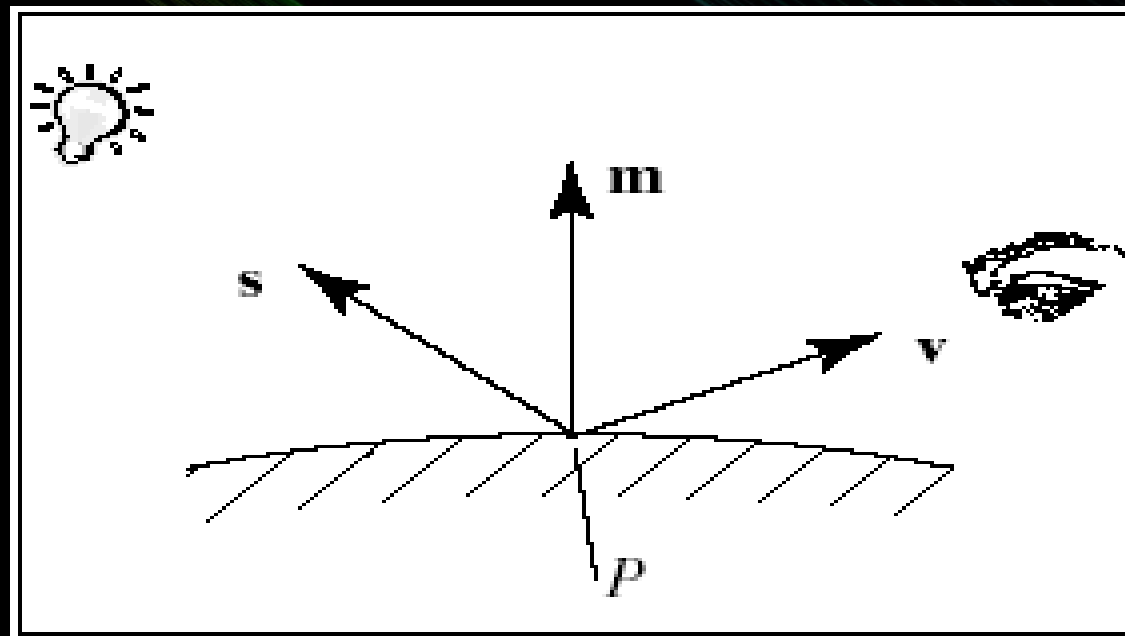


# Reflexão



# Shading

- Para calcularmos a incidência de luz a partir de um ponto  $p$  devemos considerar:
  - O vetor  $v$  de  $p$  até o olho;
  - O vetor  $s$  de  $p$  até a fonte de luz;
  - O vetor  $m$  normal da superfície sobre o ponto  $p$ ;





# *Shading*

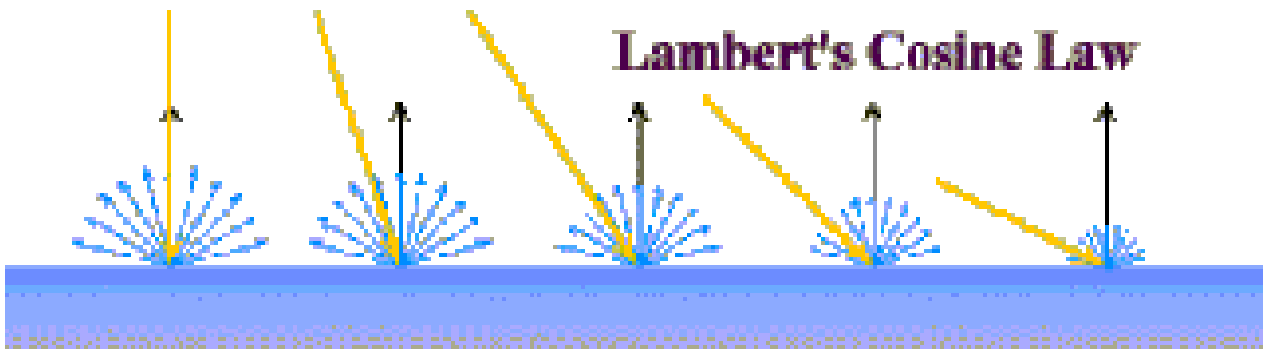
- Radiância de cena:
  - Potência da luz idealmente emitida por cada ponto  $p$  de uma superfície no espaço 3D;
- Irradiância da imagem:
  - Potência da luz chegando em cada ponto  $p$  do plano de imagem;
- Reflectância:
  - Relação entre elas: Razão entre fluxo incidente e refletido;

# Lei de Lambert

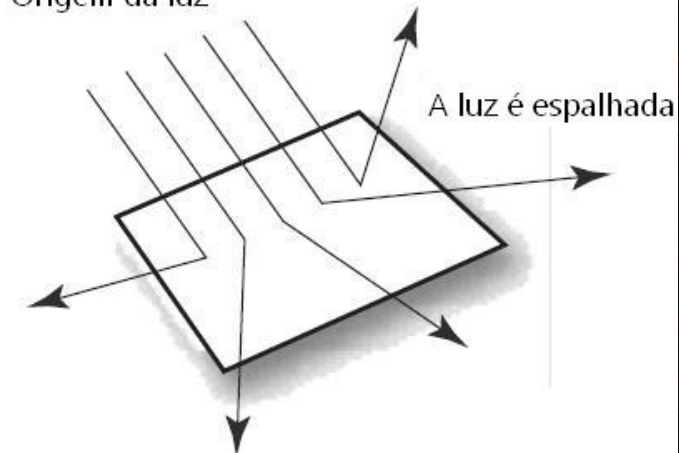
- Modelo mais simples, onde a luz recebida é igualmente refletida em todas as direções;
- O brilho não depende da direção;

Illumination direction

Lambert's Cosine Law



Origem da luz

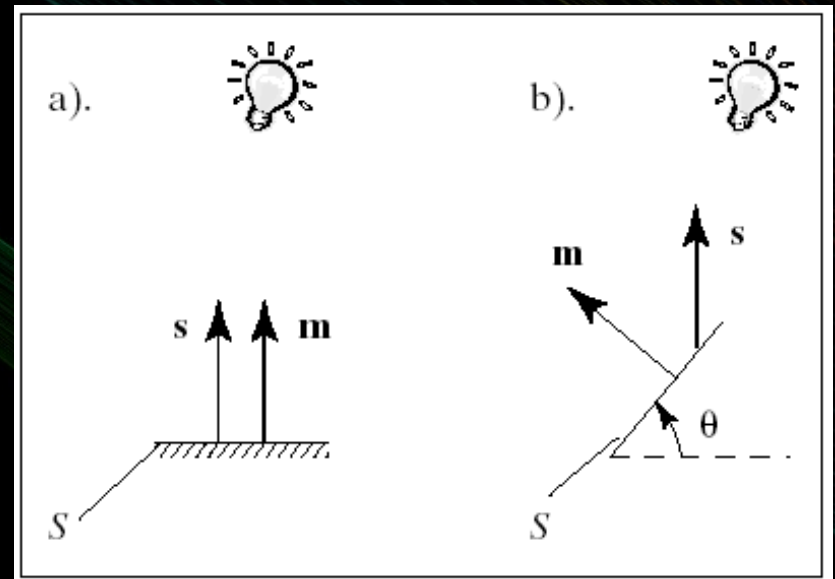




# Lei de Lambert

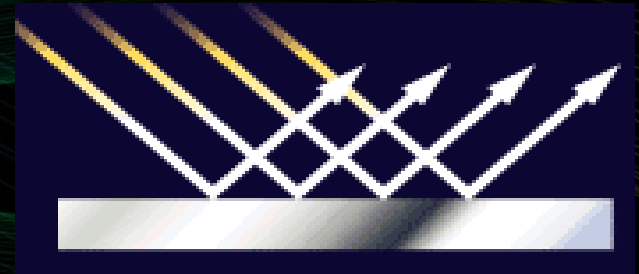
- $I_d$  = intensidade de luz difusa;
- $I_s$  = intensidade da fonte de luz ;
- $\rho_d$  = coeficiente de reflexão do material;

$$I_d = I_s \rho_d \frac{s \cdot m}{|s||m|}$$



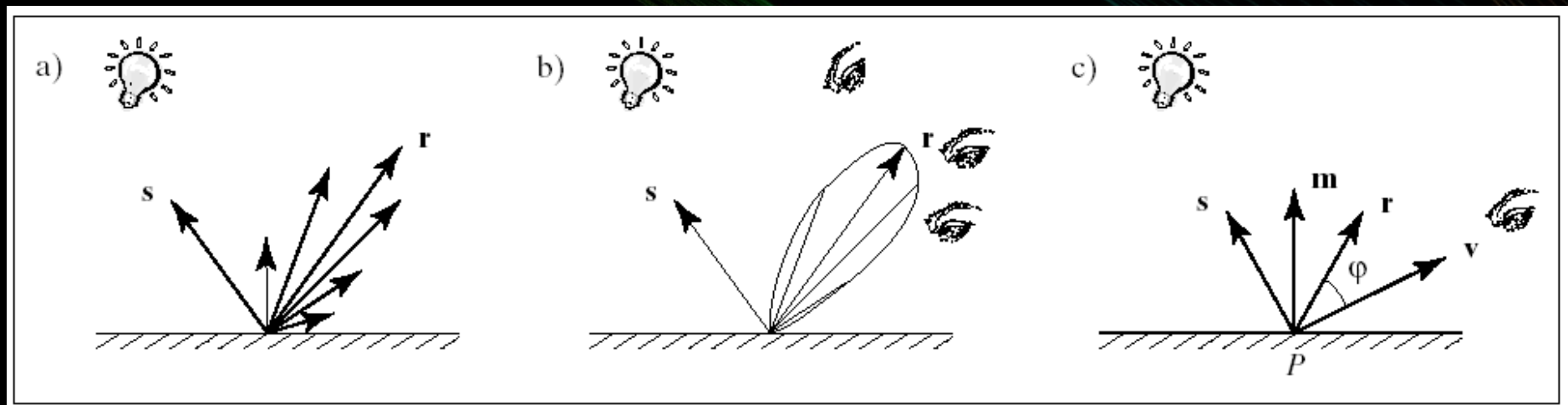
# Reflexão Especular

- Brilho depende da direção:



$$r = -s + 2 \frac{(s \cdot m)}{|m|^2} m$$

$$I_{sp} = I_s \rho_s \left( \frac{r \cdot v}{|r||v|} \right)^f$$





# Luz Ambiente

- Se não existisse luz ambiente, toda penumbra causaria sombra;
- Luz ambiente é composta pela intensidade de luz da cena sobre o material (coeficiente de reflexão);

Material	ambient: $\rho_{ar}, \rho_{ag}, \rho_{ab}$	diffuse: $\rho_{dr}, \rho_{dg}, \rho_{db}$	specular: $\rho_{sr}, \rho_{sg}, \rho_{sb}$	exponent: f
Black Plastic	0.0 0.0 0.0	0.01 0.01 0.01	0.50 0.50 0.50	32
Brass	0.329412 0.223529 0.027451	0.780392 0.568627 0.113725	0.992157 0.941176 0.807843	27.8974
Bronze	0.2125 0.1275 0.054	0.714 0.4284 0.18144	0.393548 0.271906 0.166721	25.6
Chrome	0.25 0.25 0.25	0.4 0.4 0.4	0.774597 0.774597 0.774597	76.8
Copper	0.19125 0.0735 0.0225	0.7038 0.27048 0.0828	0.256777 0.137622 0.086014	12.8
Gold	0.24725 0.1995 0.0745	0.75164 0.60648 0.22648	0.628281 0.555802 0.366065	51.2
Pewter	0.10588 0.058824 0.113725	0.427451 0.470588 0.541176	0.3333 0.3333 0.521569	9.84615
Silver	0.19225 0.19225 0.19225	0.50754 0.50754 0.50754	0.508273 0.508273 0.508273	51.2
Polished Silver	0.23125 0.23125 0.23125	0.2775 0.2775 0.2775	0.773911 0.773911 0.773911	89.6



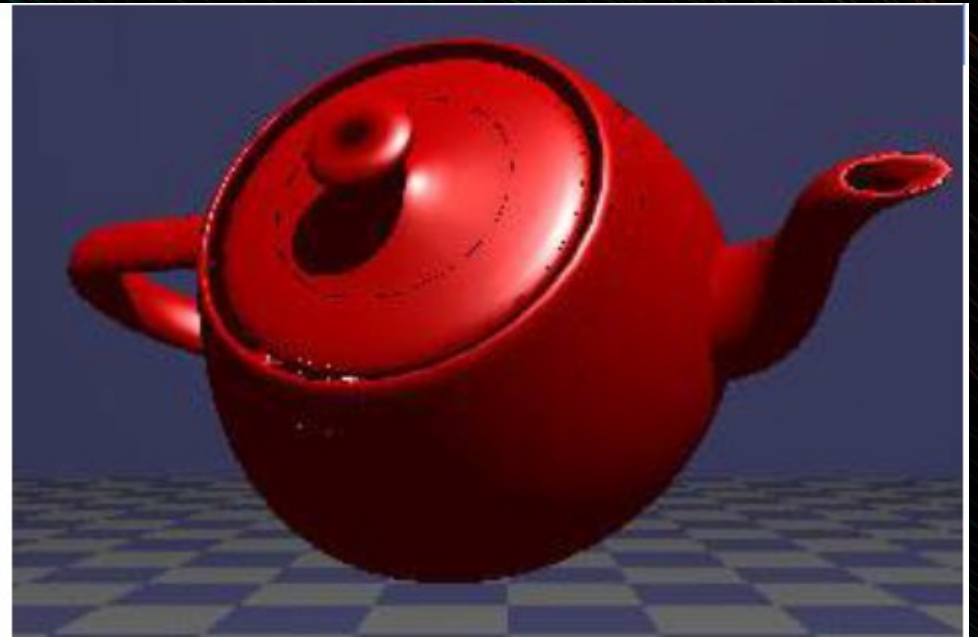
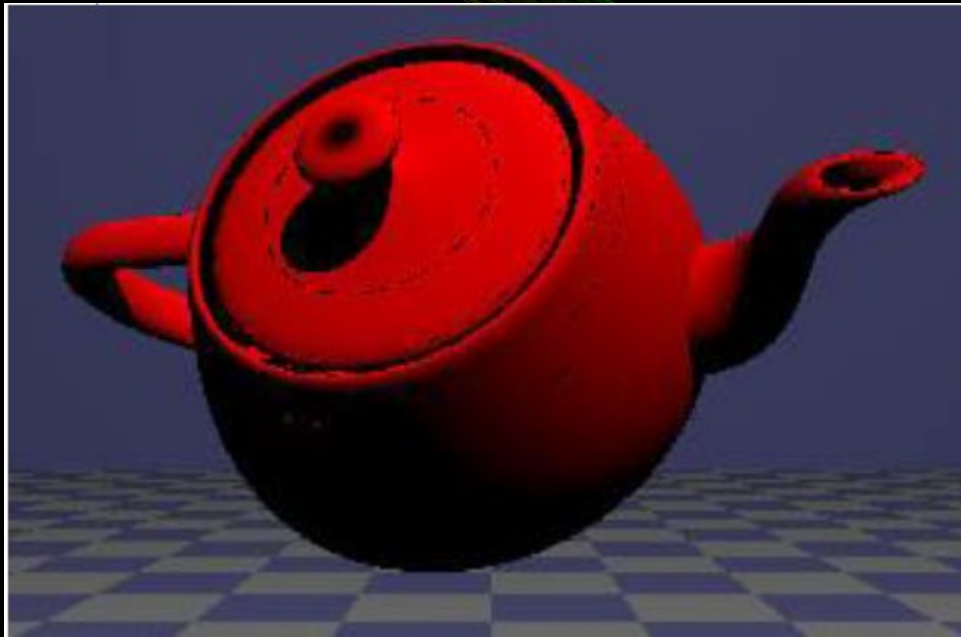
# Fonte

- Ambiente X Difusa X Especular:



# Fonte

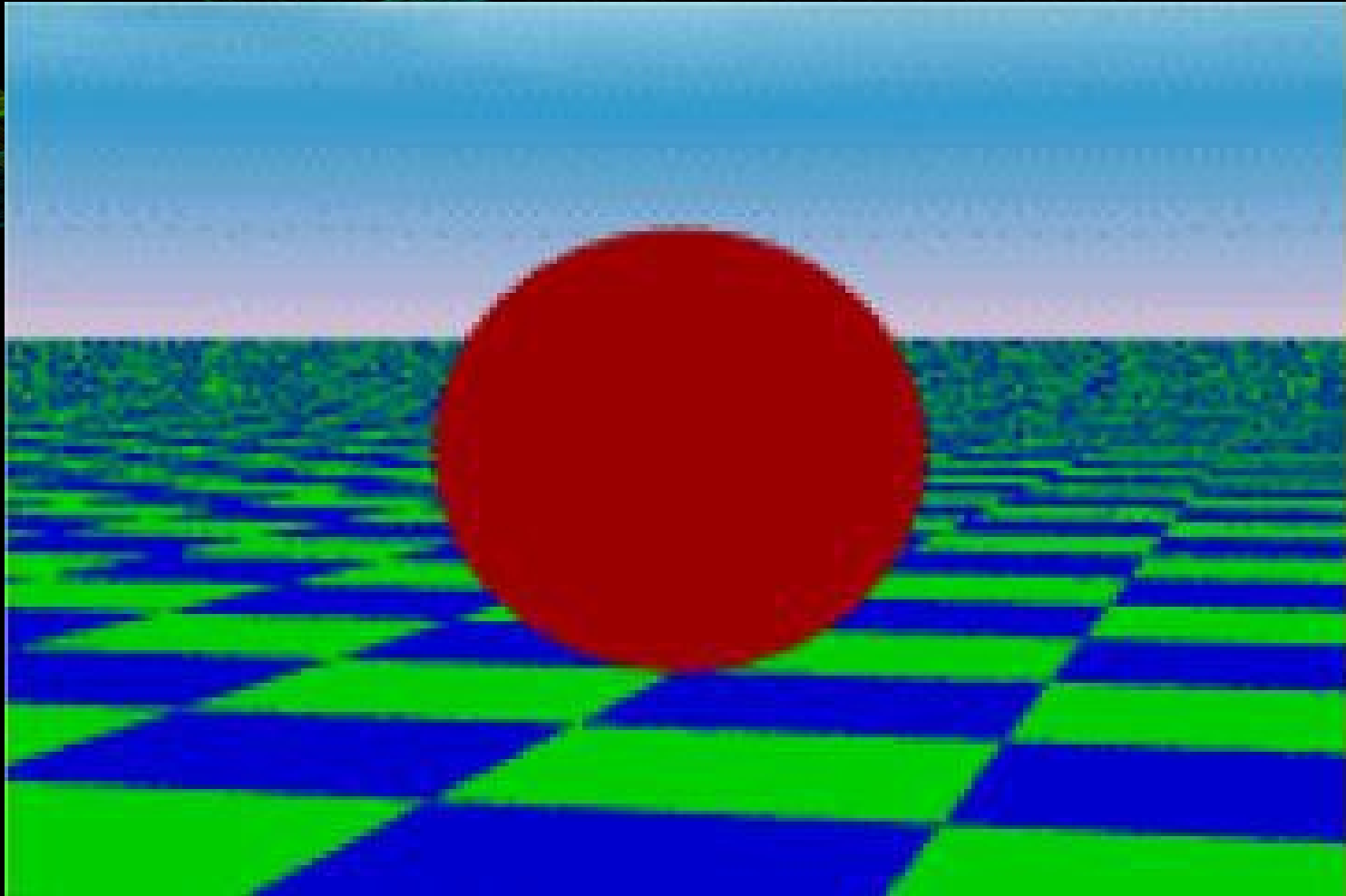
- Difusa X Especular:





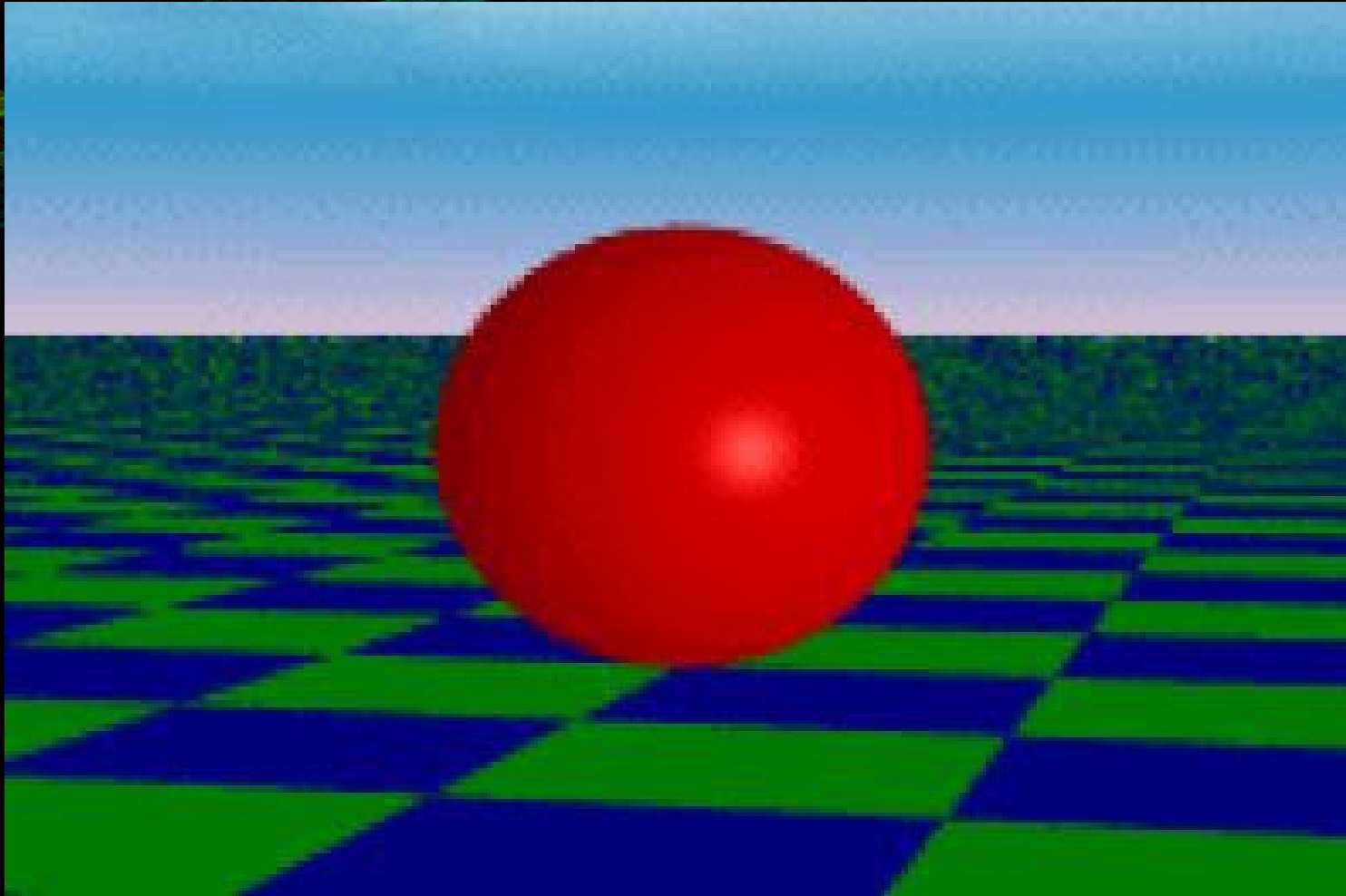
# Fonte

- Ambiente:



# Fonte

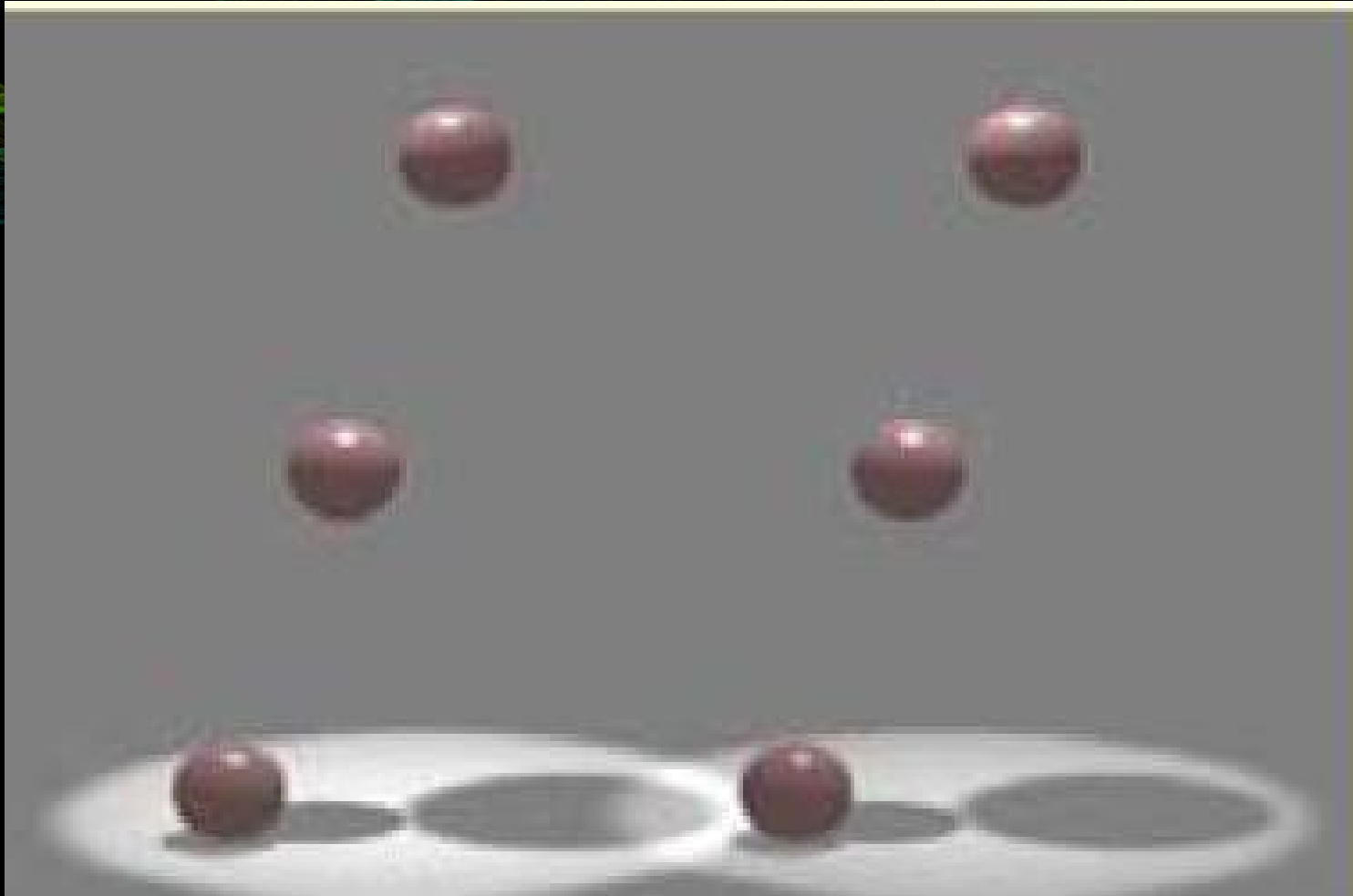
- Direcional:





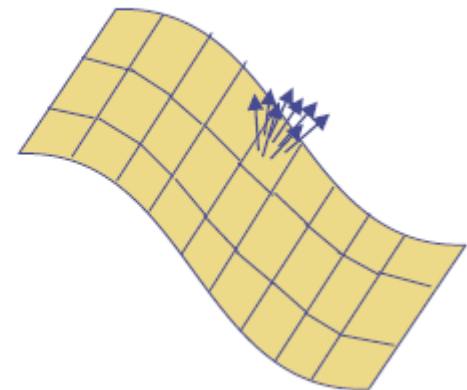
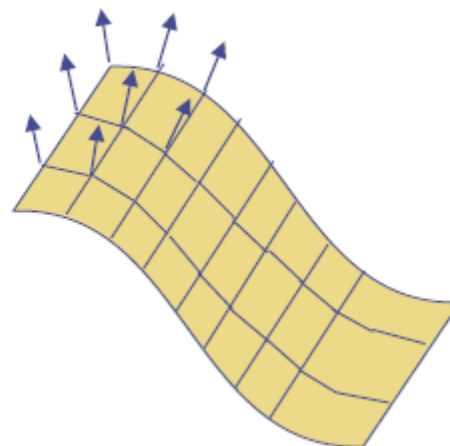
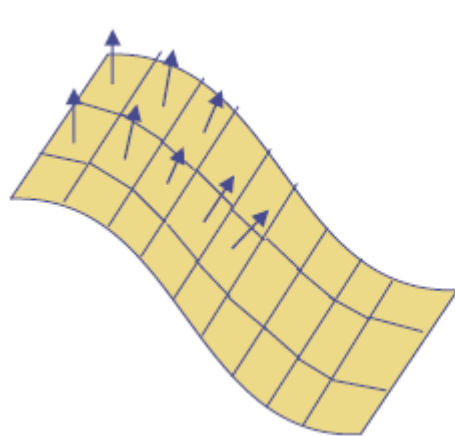
# Fonte

- Spot:



# Normal

```
glBegin(GL_POLYGON);  
    for (int i = 0; i < 3; i++)  
    {  
        glNormal3f(norm[i].x, norm[i].y, norm[i].z);  
        glVertex3f(pt[i].x, pt[i].y, pt[i].z);  
    }  
glEnd();
```





# OpenGL

- Constantes de OpenGL prontas para modelos de iluminação:
  - glLightfv;
  - glMaterialfv;

Constante	Valor default	Significado
GL_AMBIENT	(0.0,0.0,0.0,1.0)	Cor ambiente da luz
GL_DIFFUSE	(1.0,1.0,1.0,1.0)	Cor difusa da luz
GL_SPECULAR	(1.0,1.0,1.0,1.0)	Cor especular da luz
GL_POSITION	(0.0,0.0,1.0,0.0)	Posição da luz
GL_SPOT_DIRECTION	(0.0,0.0,-1.0)	Direção da luz spot
GL_SPOTCUTOFF	180.0	Ângulo de corte da luz spot

# OpenGL

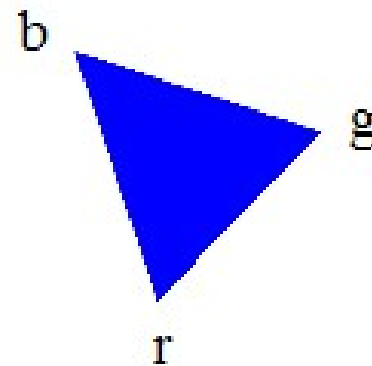
```
void Inicializa (void)
{
    // Habilita a definição da cor do material a partir da cor corrente
    glEnable(GL_COLOR_MATERIAL);
    //Habilita o uso de iluminação
    glEnable(GL_LIGHTING);
    // Habilita a luz de número 0
    glEnable(GL_LIGHT0);
    // Habilita o depth-buffering
    glEnable(GL_DEPTH_TEST);

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



# OpenGL

```
glShadeModel(GL_FLAT);
```



```
glShadeModel(GL_SMOOTH);
```

