

Iluminação

José Luis Seixas Junior

Ciência da Computação
Universidade Estadual do Paraná

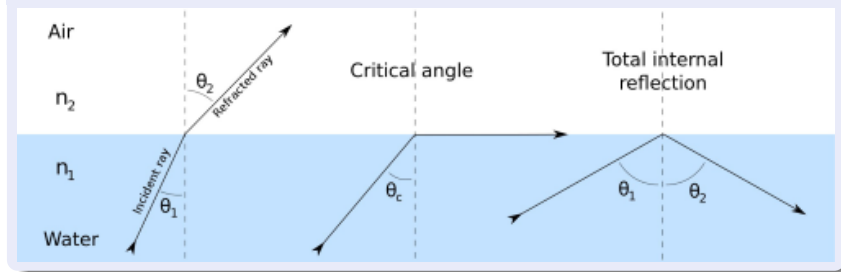
Computação Gráfica
2017

Índice

- 1 Reflexão
- 2 Shadings
- 3 Fontes
- 4 OpenGL

Reflexão

Turbidez de Materiais



Reflexão

Vídeo recomendado/Material complementar

- Physics Girl:
- *Why do mirrors flip horizontally (but not vertically)?*
- Porque espelhos invertem a imagem horizontalmente (mas não verticalmente)?
- <https://www.youtube.com/watch?v=vBpxhfB1VLU>

Reflexão

Definição

- Alteração da propagação de energia, com a diferença do ângulo de incidência sobre a normal do objeto;
- **Normal:** Perpendicular da tangente no ponto de incidência;
- **Ângulo de saída:** O mesmo, porém com *flip* sobre a normal;

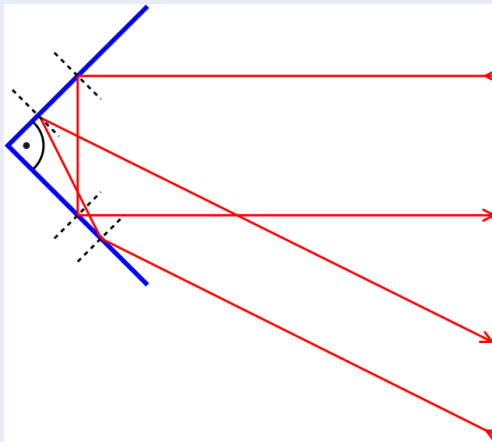
Reflexão

Luz

- Quando a luz incide sobre uma superfície opaca, parte dela é absorvida, o restante é refletido;
- A parte refletida que chega até a retina é o que 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

Angulação de Reflexão



Shadings

Radiância de cena

- Potência da luz idealmente emitida por cada ponto de uma superfície no espaço 3D;

Irradiância da imagem

- Potência da luz chegando em cada ponto do plano de imagem;

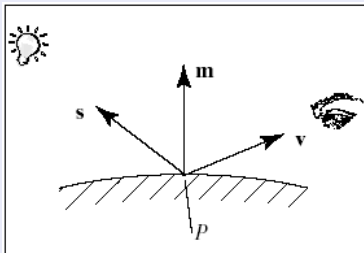
Reflectância

- Relação entre elas: Razão entre fluxo incidente e refletido;

Shadings

Incidência a partir de p

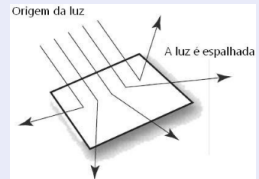
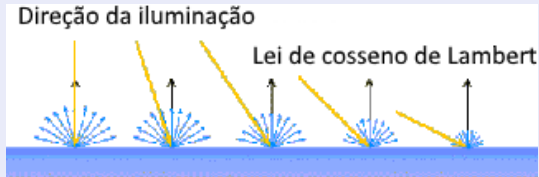
- Devemos considerar:
 - O vetor v de p até o olho;
 - O vetor s de p até a fonte de luz;
 - O vetor m normal da superfície sobre o ponto p ;



Shadings

Lei de Lambert

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

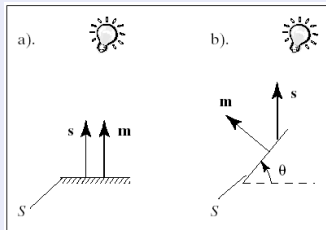


Shadings

Lei de Lambert

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

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



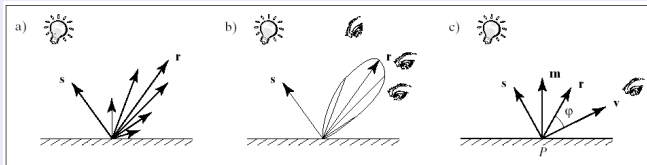
Shadings

Reflexão Especular



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

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



Shadings

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;

Materiais

Material	Ambient	Difusa	Especular	Expoente: f
Asfalto	0.0	0.01	0.50	32
	0.0	0.01	0.50	
	0.0	0.01	0.50	
Latão	0.329412	0.780392	0.992157	27.8974
	0.223529	0.568627	0.941176	
	0.027451	0.113725	0.807843	
Bronze	0.2125	0.714	0.393548	25.6
	0.1275	0.4284	0.271906	
	0.054	0.18144	0.166721	
Cromado	0.25	0.4	0.774597	76.8
	0.25	0.4	0.774597	
	0.25	0.4	0.774597	

Materiais

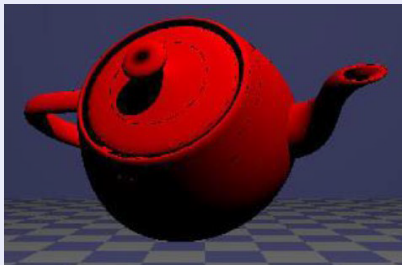
Material	Ambient	Difusa	Especular	Expoente: f
Cobre	0.19125	0.7038	0.256777	12.8
	0.0735	0.27048	0.137622	
	0.0225	0.0828	0.086014	
Ouro	0.24725	0.75164	0.628281	51.2
	0.1995	0.60648	0.555802	
	0.0745	0.22648	0.366065	
Prata	0.19225	0.50754	0.508273	51.2
	0.19225	0.50754	0.508273	
	0.19225	0.50754	0.508273	
Prata Polida	0.23125	0.2775	0.773911	89.6
	0.23125	0.2775	0.773911	
	0.23125	0.2775	0.773911	

Fontes



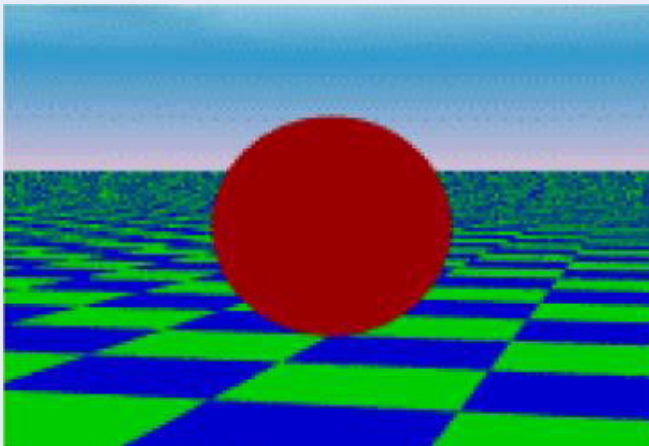
Fontes

Difusa X Especular



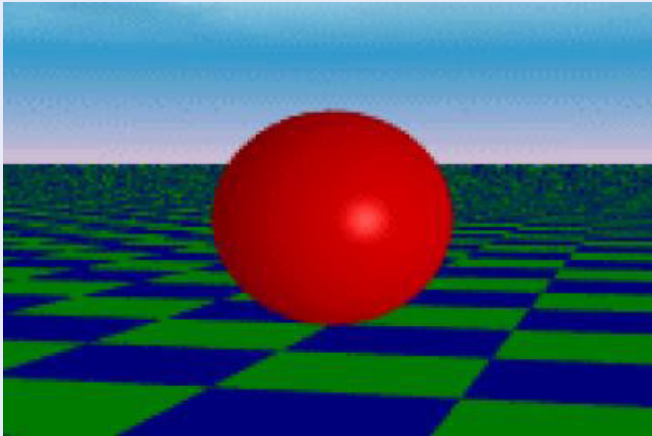
Fontes

Ambiente



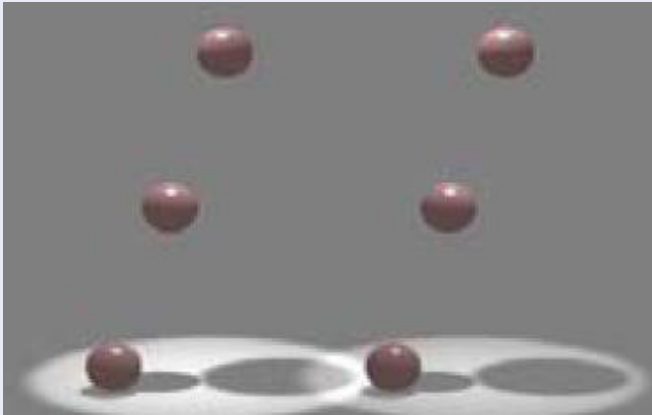
Fontes

Direcional



Fontes

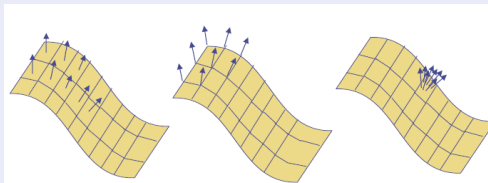
Spot



Fontes

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 Iluminação

- `glLightfv;`
- `glMaterialfv;`

Constante	Valor Padrão	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)	Posição da luz spot
GL_SPOT_DIRECTION	(1.0, 1.0, -1.0)	Direção da luz spot
GL_SPOTCUTOFF	180.0	Ângulo de corte da luz spot

OpenGL

Normal

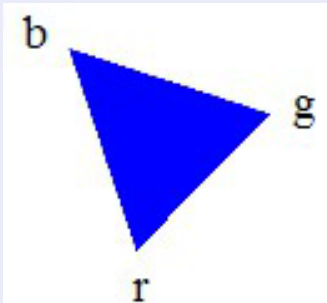
```
void Inicializa (void)
{
    // Habilita a definição da cor do material a partir da cor corrente
    glEnable(GL_COLOR_MATERIAL);
    //Habilita o uso de iluminação
    glEnable(GL_LIGHTING);
    // Habilita a luz de número 0
    glEnable(GL_LIGHT0);
    // Habilita o depth-buffering
    glEnable(GL_DEPTH_TEST);

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

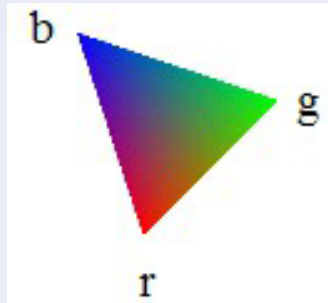
    ...
}
```

OpenGL

`glShadeModel(GL_FLAT)`



`glShadeModel(GL_SMOOTH)`



Atividade 10

Atividade 10/1

- Adicionar *Flat Shading* nas estruturas de arame contruídas no exercício 08/1.

Atividade 10/2

- Adicionar cor difusa sobre as cenas criadas no exercício 09/1.

Data

30 de novembro de 2017

Atividades Atrasadas

Qualquer atividade do segundo semestre

- Atenção a qualquer exigência da atividade;
- Nenhum trabalho será aceito após esta data;
- Valor máximo de 70%;

Data

30 de novembro de 2017

Referências I



Hill, F. S.

Computer Graphics Using OpenGL.

Prentice Hall, 2013.



Shreiner, D.; Woo M.; Neider, J.; Davis, T.

OpenGL Programming Guide.

Addison Wesley, 4º edição, 2013.