

---

# Representação de Objetos e Cenas

**Soraia Musse**



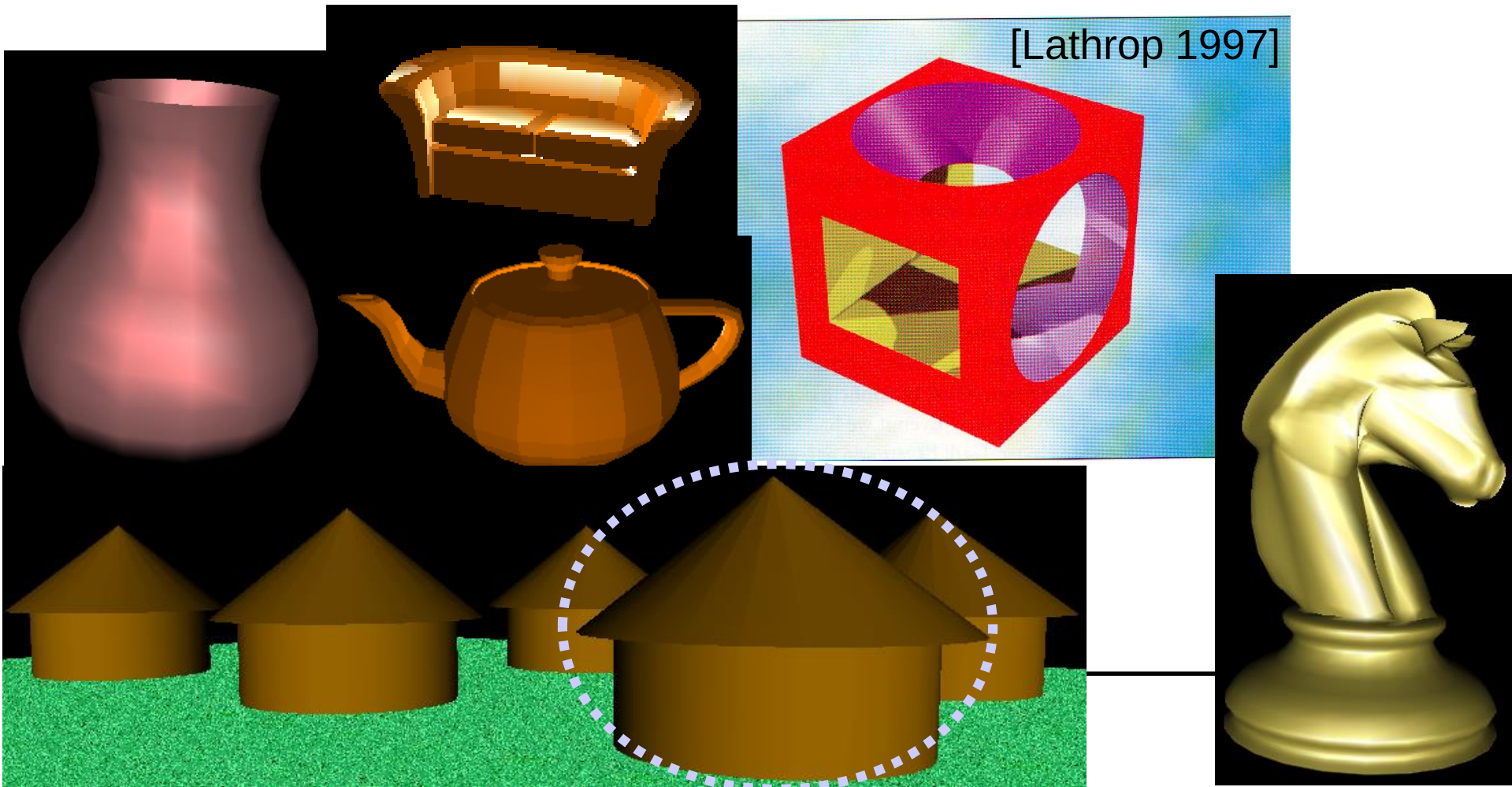
---

# Roteiro

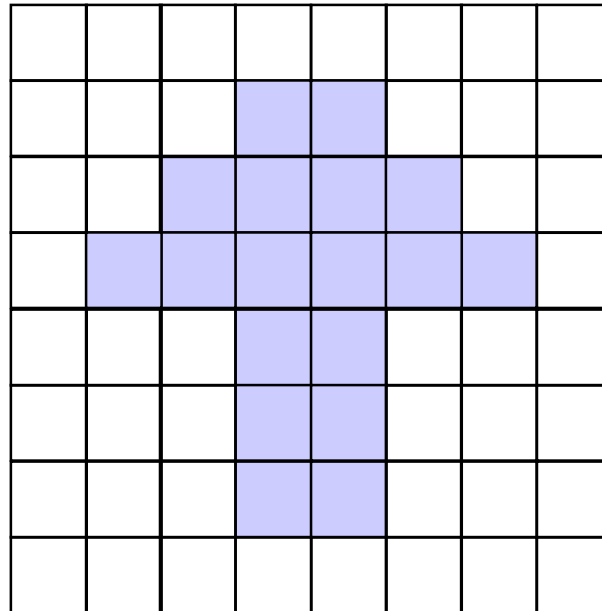
- 1.** Formas de Representação
    - 1.1. Representação Aramada
    - 1.2. Superfícies Limitantes
    - 1.3. Enumeração Espacial
    - 1.4. Representação Paramétrica
    - 1.5. Grafo de Cena
  - 2.** Técnicas de Modelagem 3D
    - 2.1. Varredura
    - 2.2. CSG
    - 2.3. Instanciamento de Primitivas
    - 2.4. Fractais
    - 2.5. Sistemas de Partículas
-

# Exercícios

- Quais técnicas de representação e modelagem podem ser utilizadas para os objetos abaixo? Justifique.



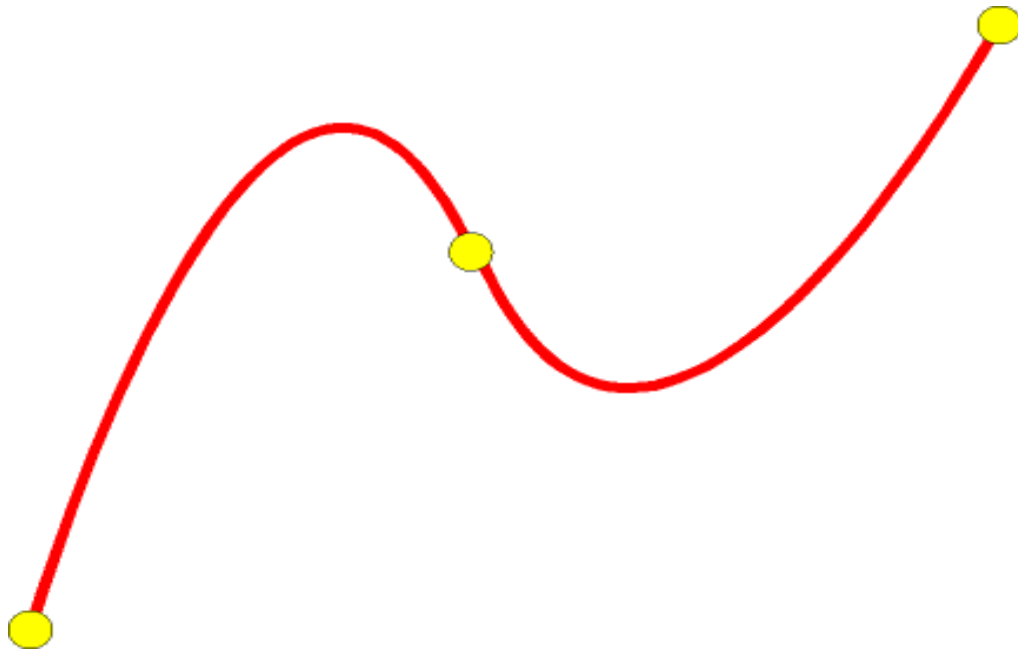
- Crie uma *quadtree* para representar a figura abaixo. Use uma árvore com no máximo 5 níveis



---

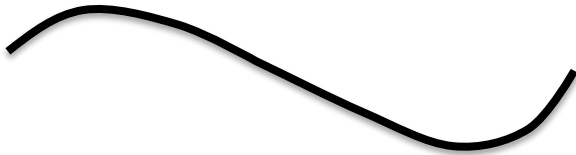
# Curvas

- Curvas
  - Apenas comprimento



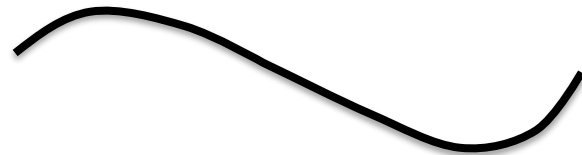
---

# Interpolação X Aproximação



Na interpolação, a curva passa sobre todos os pontos definidos.

**Hermite,  
Catmull-Rom spline**



Na aproximação, a curva começa sobre o ponto inicial e termina sobre o final. Os demais pontos são aproximados.

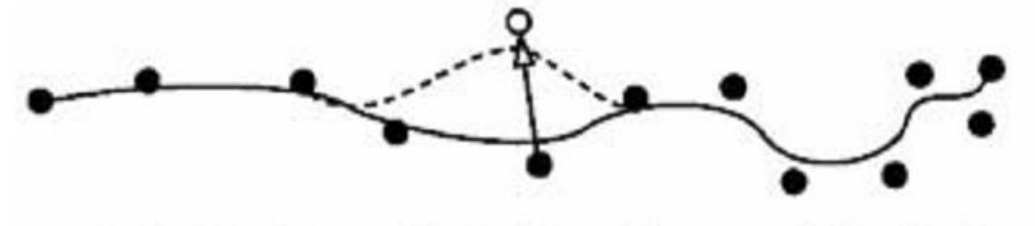
**Bézier, curvas B-spline**

---

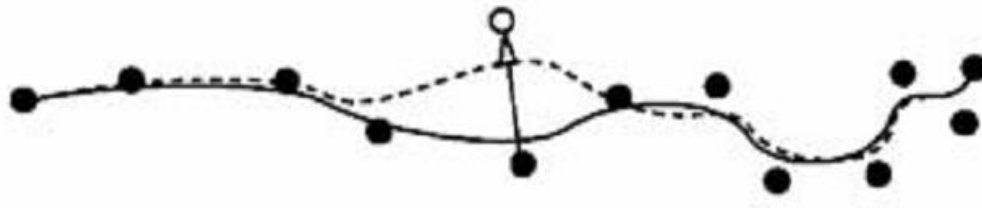
---

# Controle

- Local



- Global



---

# Métodos para representar curvas

- Hermite
- Bézier
- B-Spline



---

# Transformações 3D

Soraia Raupp Musse

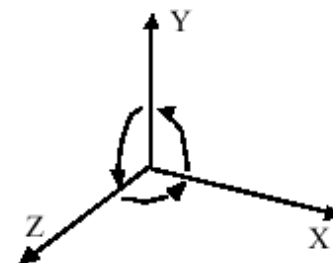
---

# Transformações 3D

- Translação – `glTranslatef(dx, dy, dz)`

–  $T(dx, dy, dz)$ :

$$\begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Sistema de Coordenadas  
“mão direita”

- Escala – `glScalef(Sx, Sy, Sz)`

–  $S(Sx, Sy, Sz)$ :

$$\begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

---

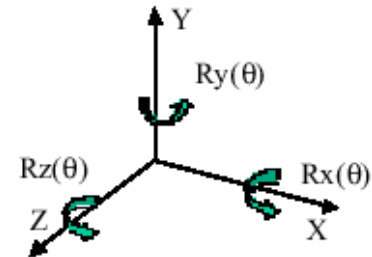
# Transformações 3D

- Rotação – `glRotatef(angle,x,y,z)`

$$R_z(\theta): \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\theta): \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

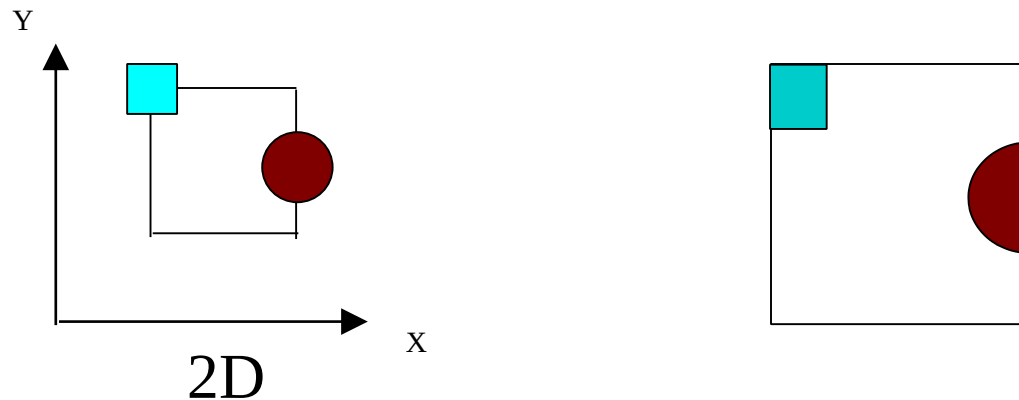
$$R_y(\theta): \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



---

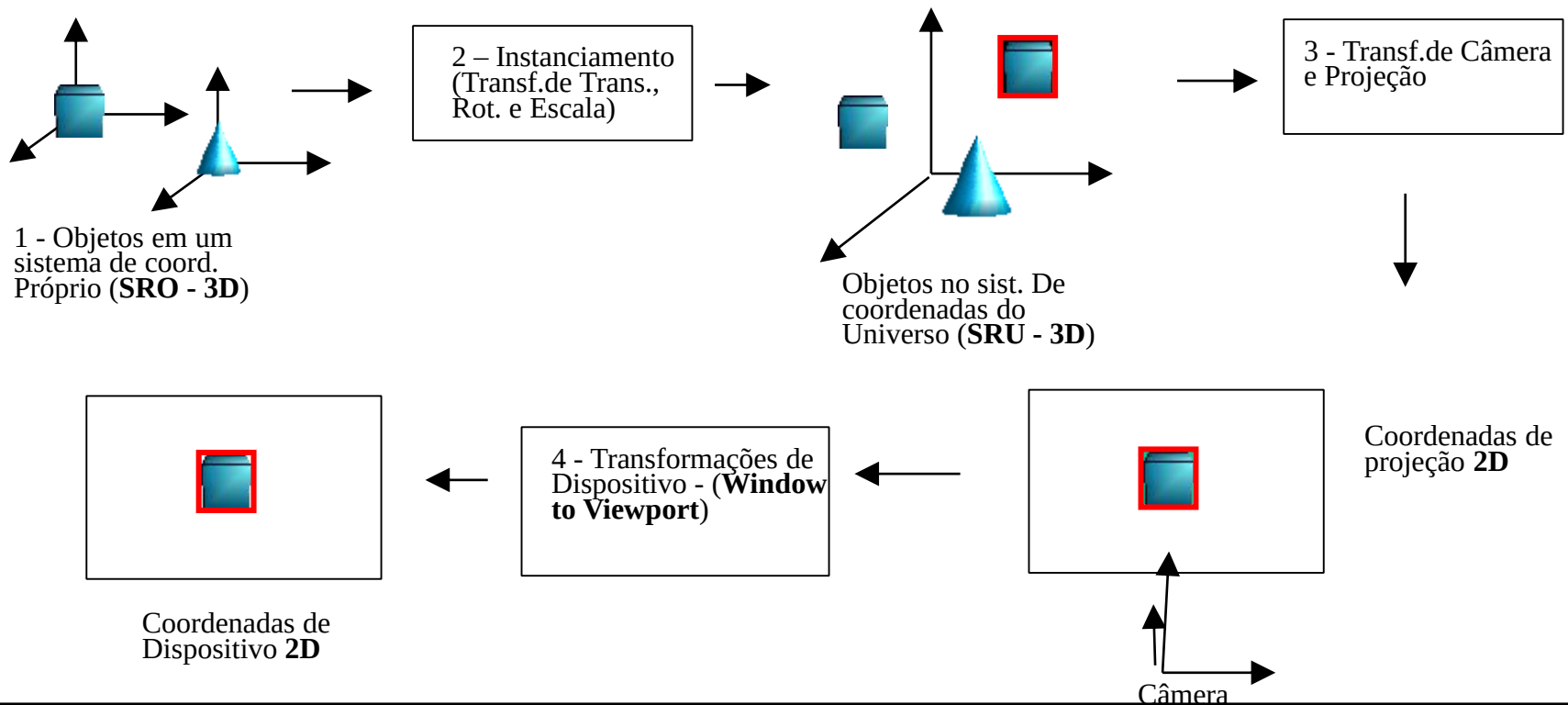
# Pipeline de Visualização

- Em 2D as coisas são mais simples
  - Simplesmente especificar uma janela do mundo 2D e uma viewport na superfície de visualização
- A complexidade começa em 3D, pelo fato de termos uma dimensão a mais, mas também pelo fato do dispositivo de exibição ser 2D



# Pipeline de Visualização 3D

- Para tirarmos uma foto virtual, precisamos executar os seguintes passos



---

# Recorte 3D

- Para o recorte em 3D, veremos a extensão do algoritmo de Cohen-Sutherland em 2D
- O algoritmo de Cohen-Sutherland classifica as extremidades das linhas de acordo com sua posição no espaço em relação ao volume de visualização
- Em 3D, esta classificação é formada por 6 bits de acordo com o seguinte
  - bit 1: ponto está acima do volume
  - bit 2: ponto está abaixo do volume
  - bit 3: ponto está à direita do volume
  - bit 4: ponto está à esquerda do volume
  - bit 5: ponto está atrás do volume
  - bit 6: ponto está à frente do volume

---

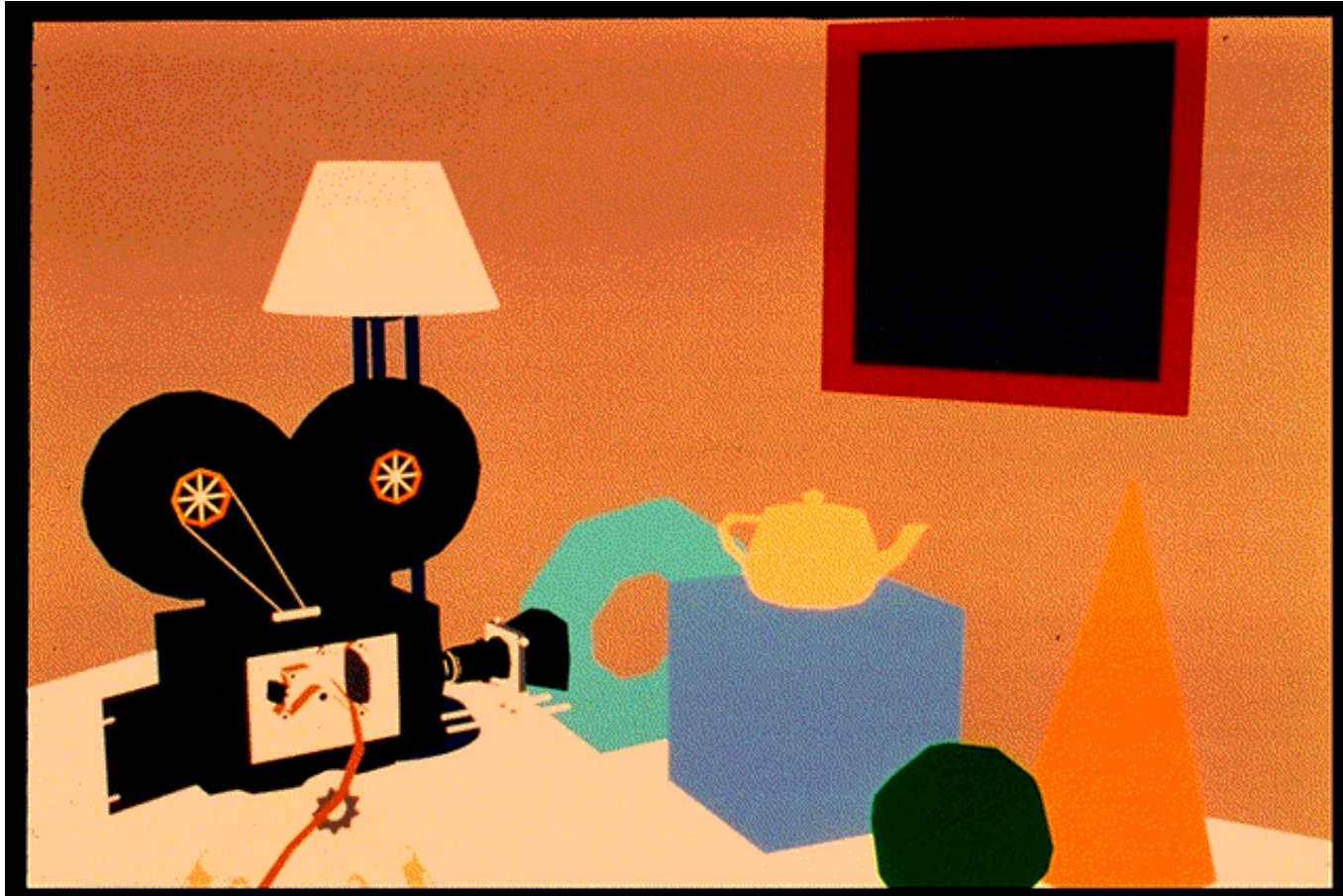
# Iluminação

**Soraia Musse**



---

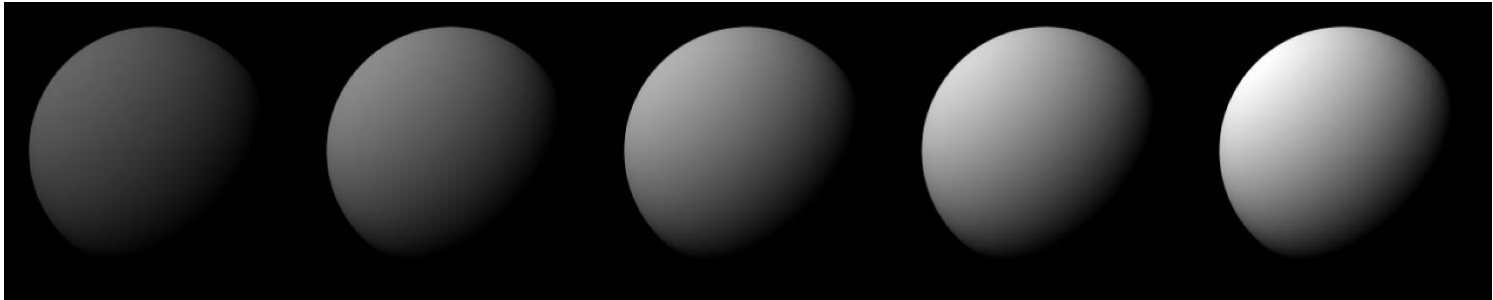
# Luz Ambiente



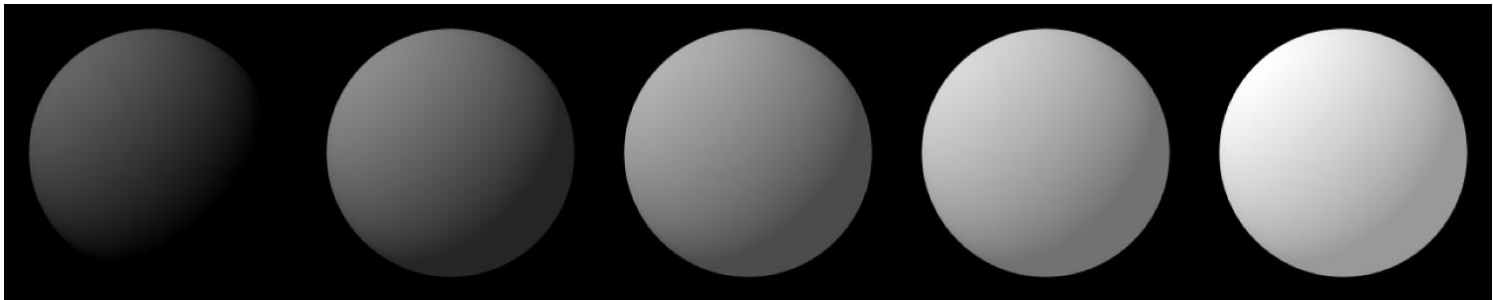


---

# Reflexão Difusa



$K_d = 0.4, 0.55, 0.7, 0.85, 1.0$



$I_a = I_p = 1.0, K_d = 0.4, K_a = 0.0, .015, 0.3, 0.45, 0.6$

# Reflexão Especular de Phong

- Phong desenvolveu em 1975, um modelo para a parcela de reflexão especular que assumia que:
  - Máxima reflexão especular ocorre quando  $\alpha^*$  é zero e cai rapidamente quando  $\alpha$  cresce
  - Essa mudança é representada por  $\cos^n \alpha$ , onde  $n$  é o expoente de reflexão especular
    - Valores de  $n$  variam de 1 até centenas, dependendo do material simulado
- A equação de iluminação fica ( $O$ =cor)

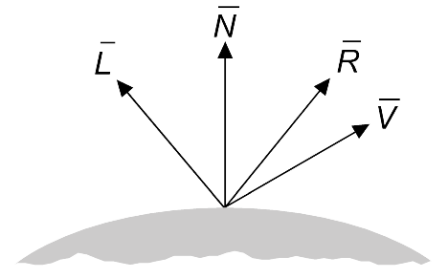
considerando vetores normalizados,

$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [K_d O_{d\lambda} (N \cdot L) + K_s (R \cdot V)^n]$$

incluindo a cor especular

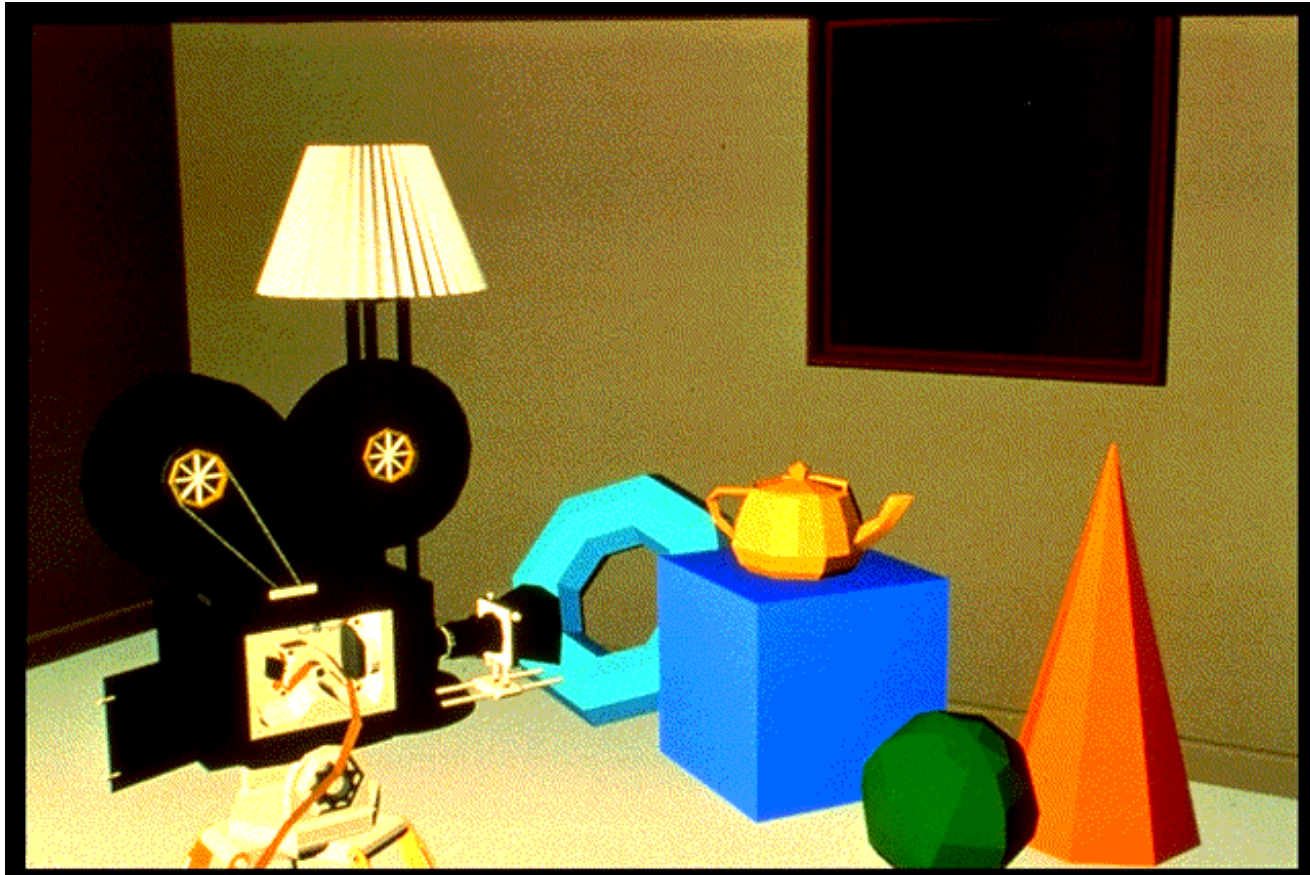
$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [K_d O_{d\lambda} (N \cdot L) + K_s O_s (R \cdot V)^n]$$

$R$  = reflexão e  $V$ =viewpoint



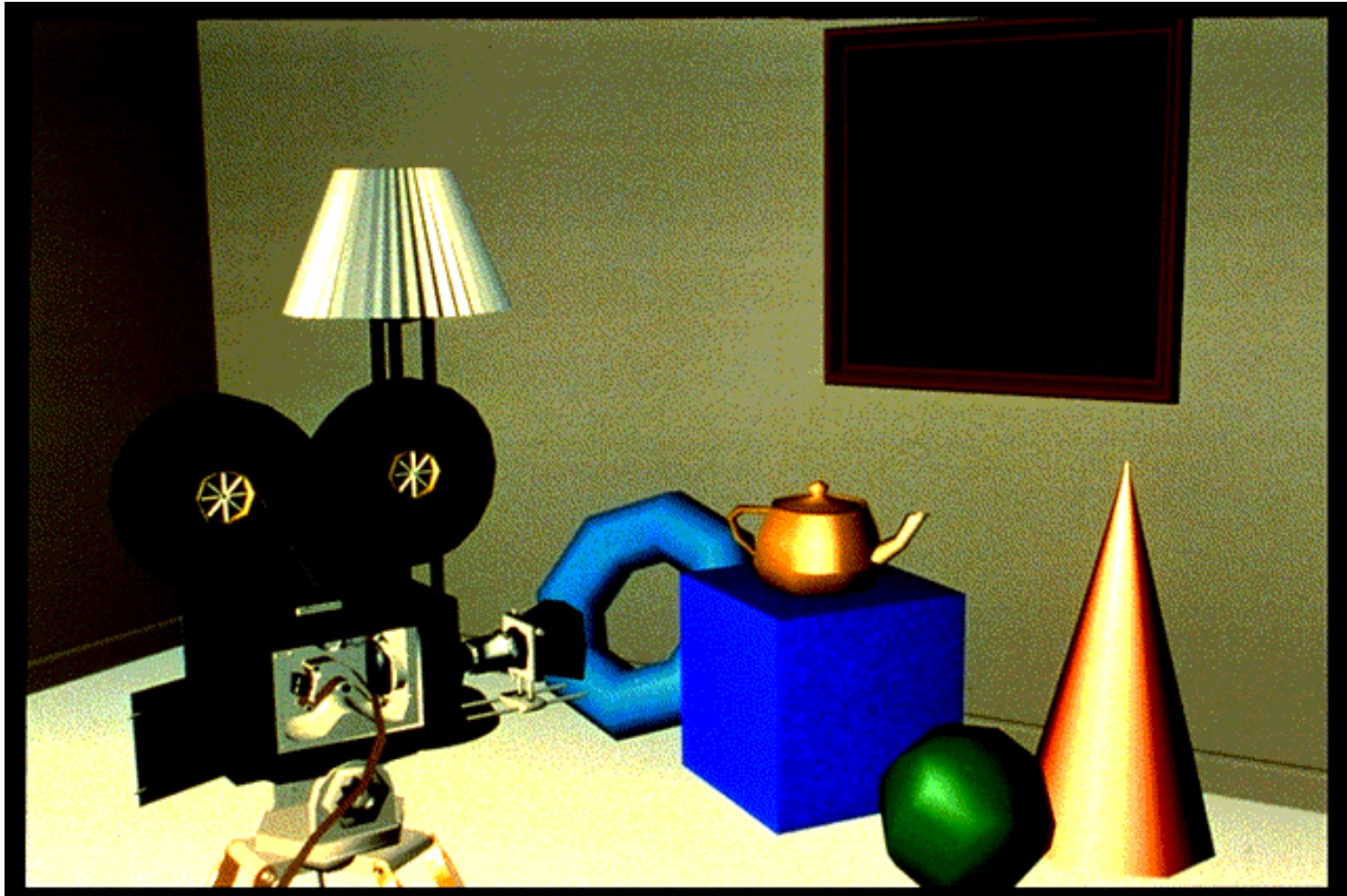
---

# Sombreamento de Malhas Poligonais



---

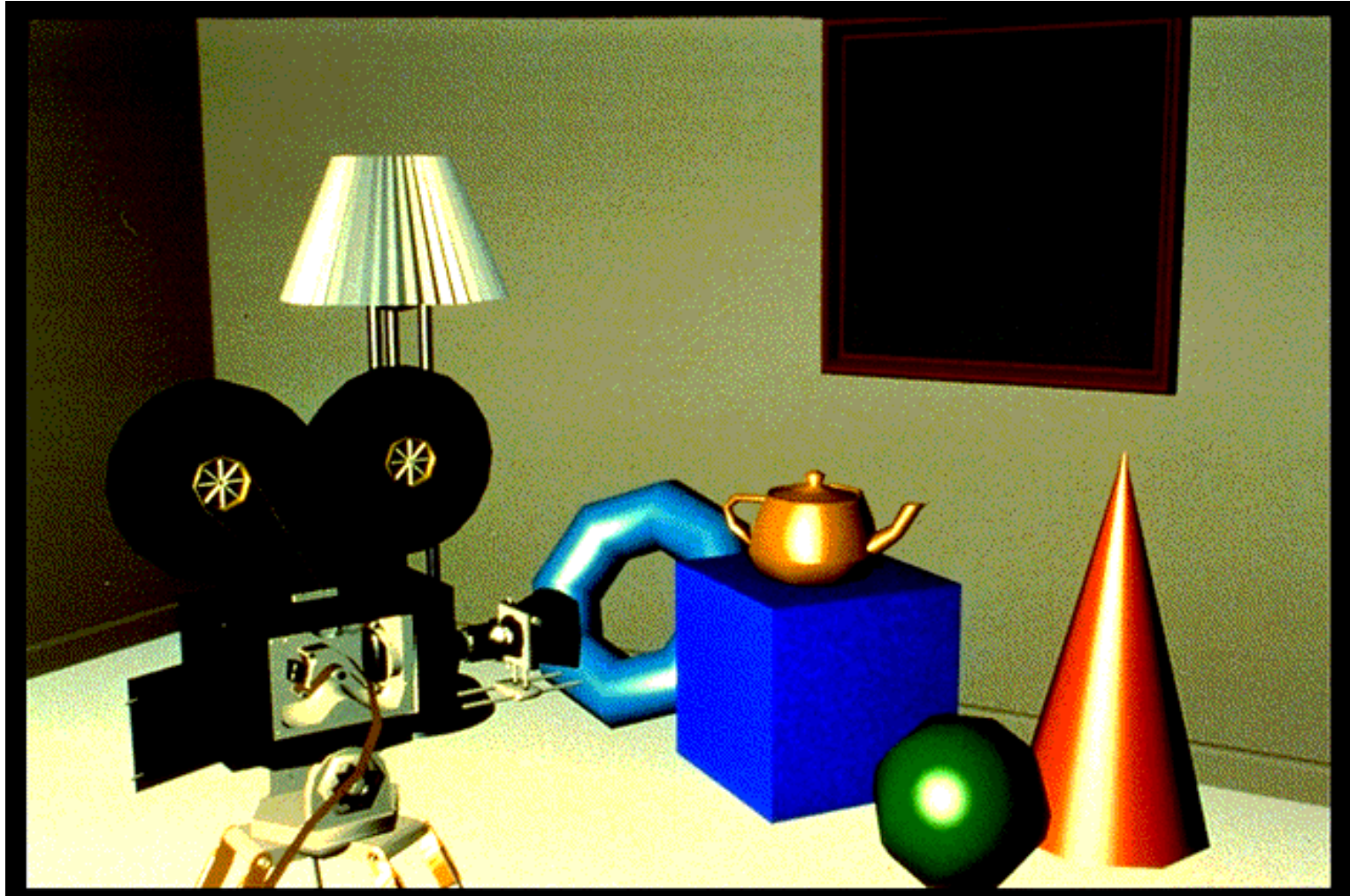
# *Gouraud Shading*





---

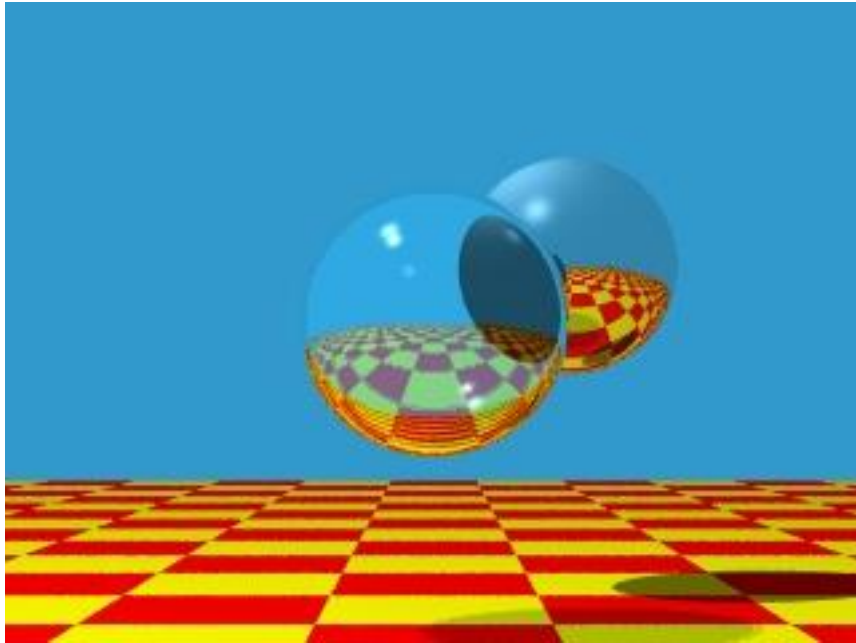
# *Phong Shading*



---

# Traçado de Raios

- Primeiras idéias em 1968  
(*Ray Casting*)
- 1980 Turner Whitted
  - *Communications of the ACM*  
N. 23, V. 6, June 1980, p.  
343-349)



---

# O que significa o termo *"Animação Computadorizada?"*

- *"Geração de frames consecutivos que são exibidos numa frequência suficiente para que o olho humano não consiga diferenciá-los e tenha então a sensação de movimento"*

O quão diferente eles  
devem ser?

OK! Mas, o que são  
frames???

Em que frequência  
devem ser exibidos?

---

---

# Técnicas de Animação

## *Keyframe (Animação por quadros-chave)*

- keyframes são especificados pelo animador e o computador gera os quadros intermediários
- Os quadros intermediários são gerados baseados numa lei de *interpolação*



---

# Técnicas de Animação

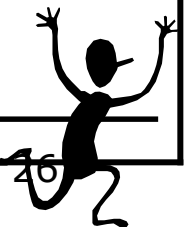
## *Scripting Systems*

- Visa oferecer uma linguagem de script (com comandos pré-definidos) que possa descrever a animação de objetos
  - Ex: #ACTORS 5
    - » ACTOR\_1 FRAME=1 POSITION 10 10 10
    - » ACTOR\_1 FRAME=100 POSITION 20 20 20

---

# Classificação de técnicas

| Diferenças                                       | Low-level | High-level |
|--------------------------------------------------|-----------|------------|
| Intervenção do usuário                           | Muita     | Pouca      |
| Nível de abstração                               | Pouca     | Muita      |
| Precisão do resultado em relação ao especificado | Muita     | Pouca      |



---

# Simulação de Multidões

**Soraia Musse**

