

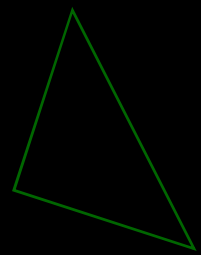
# *Técnicas de Animação*

Soraia Raupp Musse

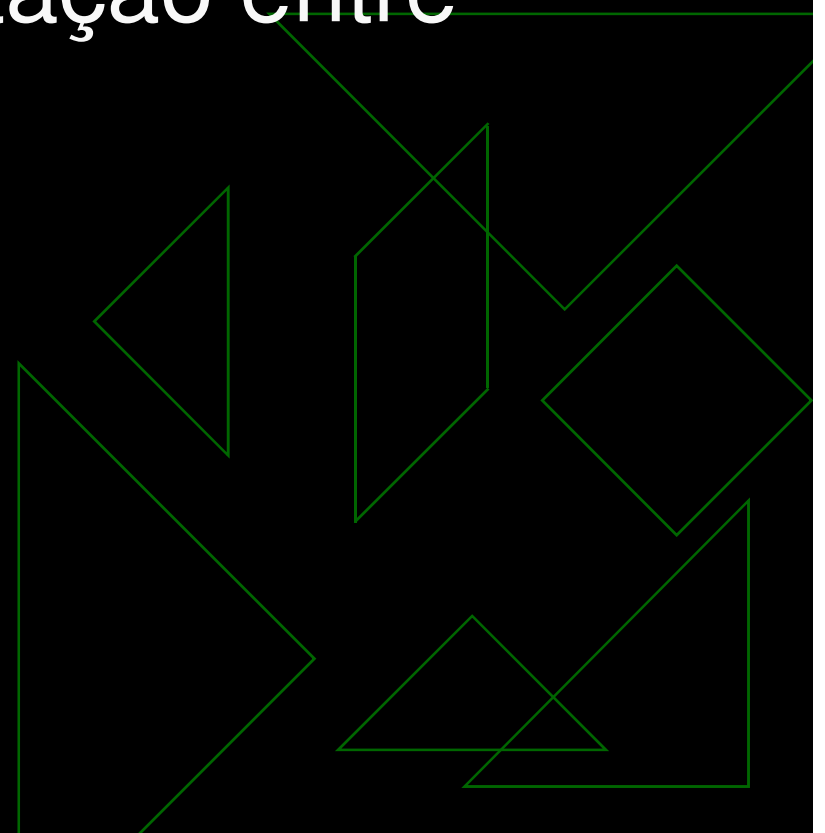
01/04/2014

# Aula passada: 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



Como gerar movimentação entre pontos no espaço?



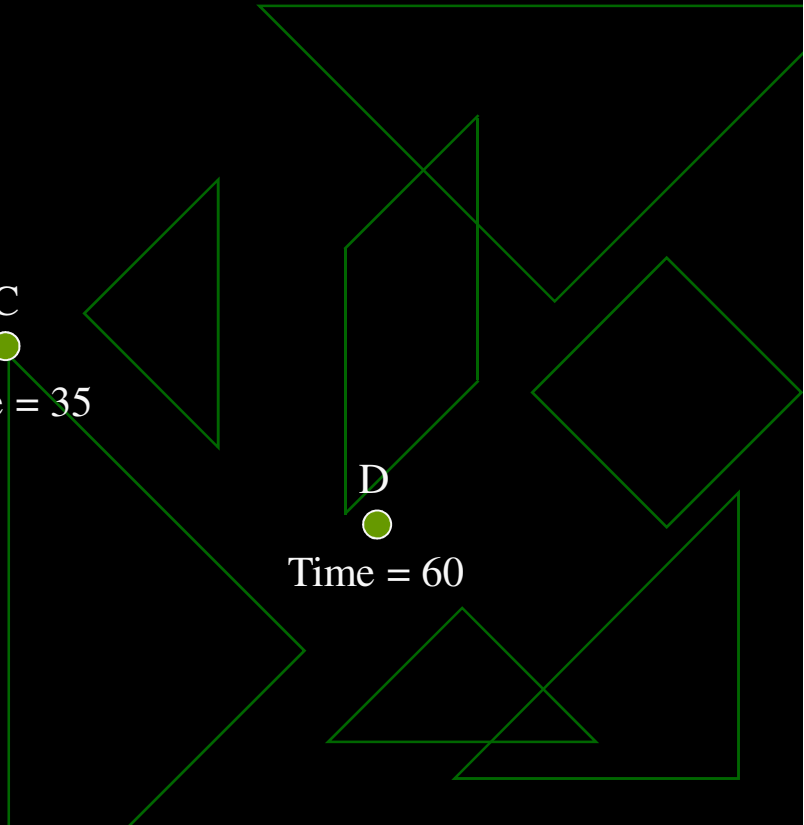
Problema: gerar uma curva SUAVE  
que passe por PONTOS  
específicos, em um TEMPO  
designado

A  
●  
Time = 0

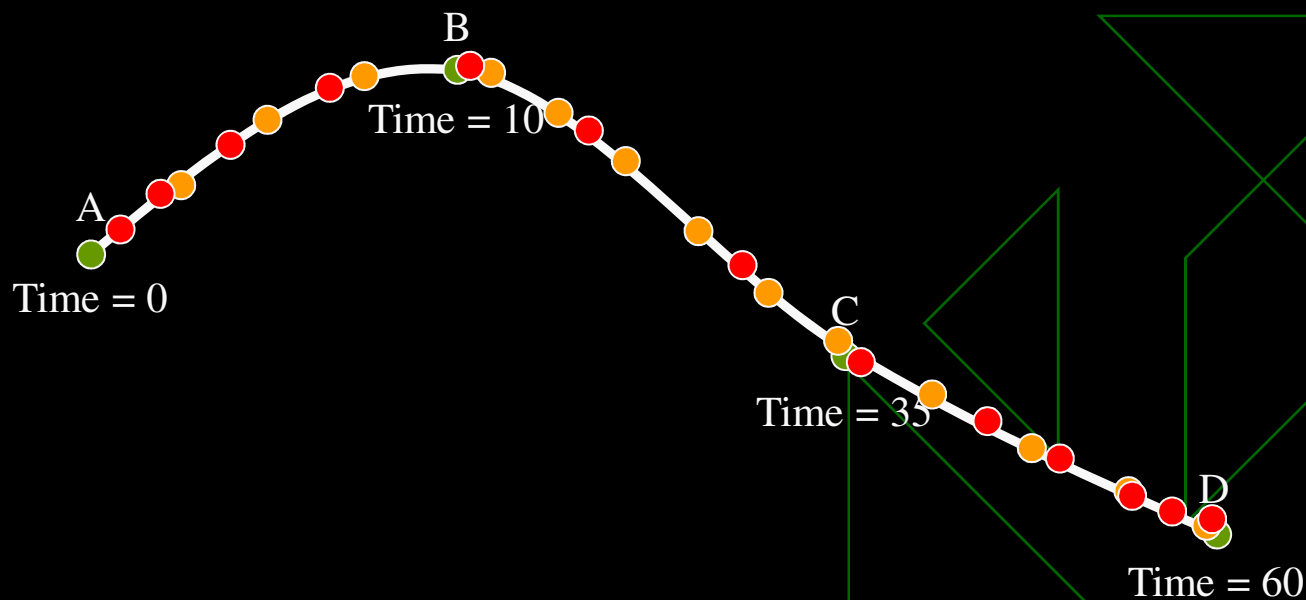
B  
●  
Time = 10

C  
●  
Time = 35

D  
●  
Time = 60

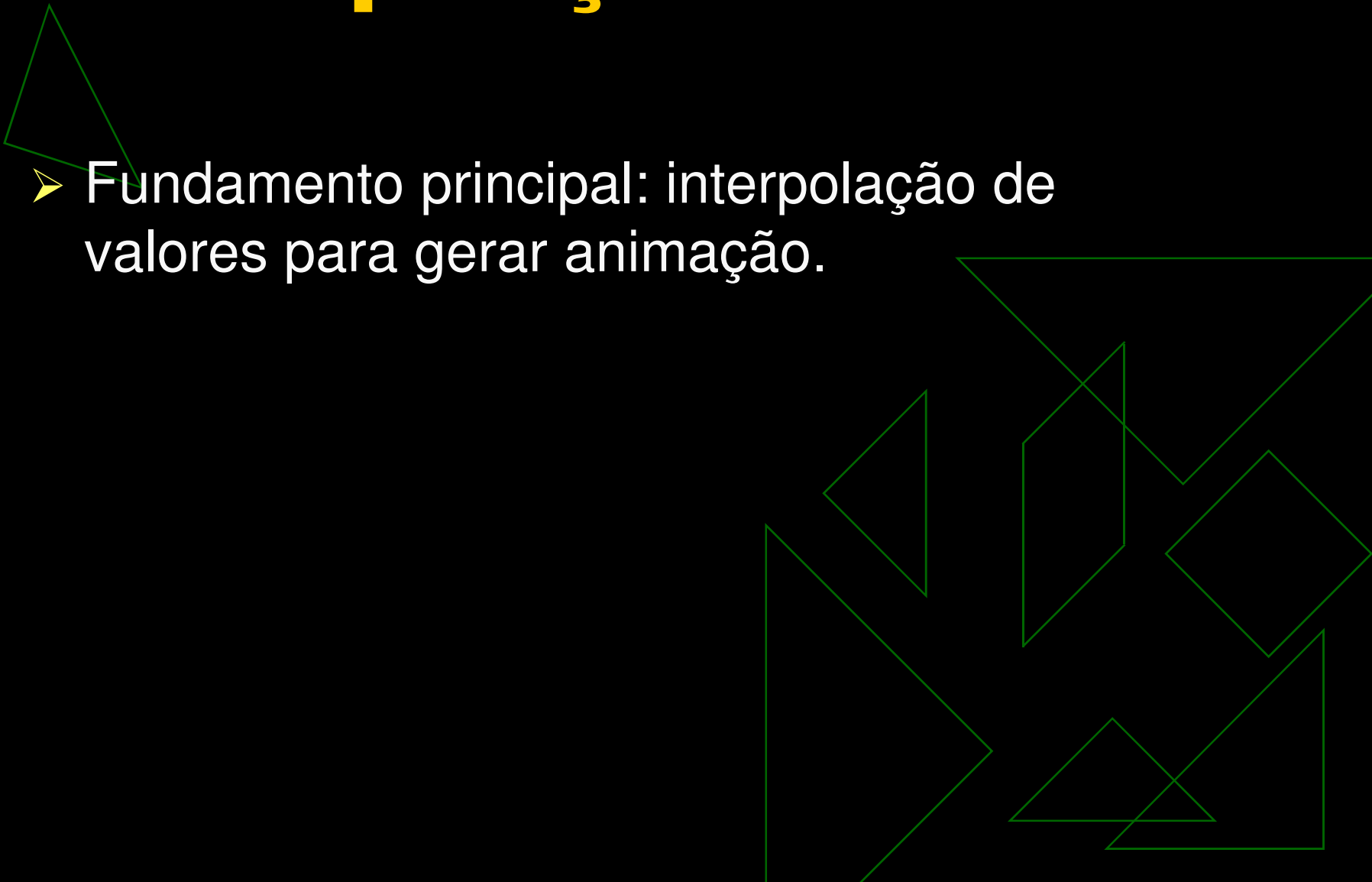


Solução: gerar uma curva no espaço, distribuindo PONTOS de maneira SUAVE e que possa variar em função do TEMPO



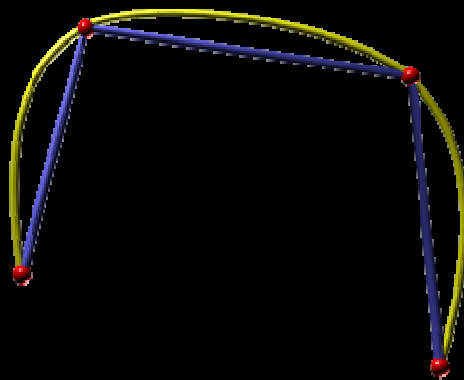
# Interpolação

- Fundamento principal: interpolação de valores para gerar animação.

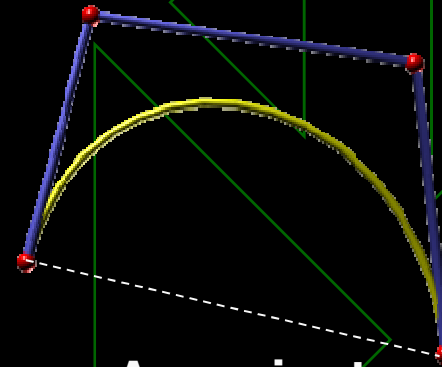


# Aproximação x Interpolação

- ◆ Dado um número  $n$  de pontos para traçar uma curva:
  - *interpolare* os pontos (curva passando por todos os pontos)
  - *aproximar* os pontos (pontos definem *convex hull* da curva)

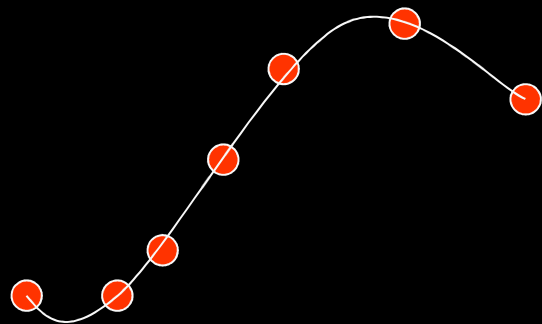
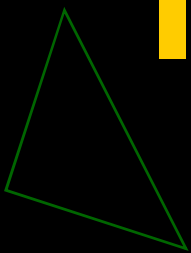


Interpolate

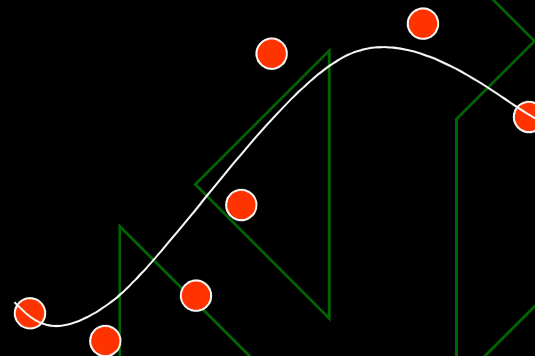


Approximate

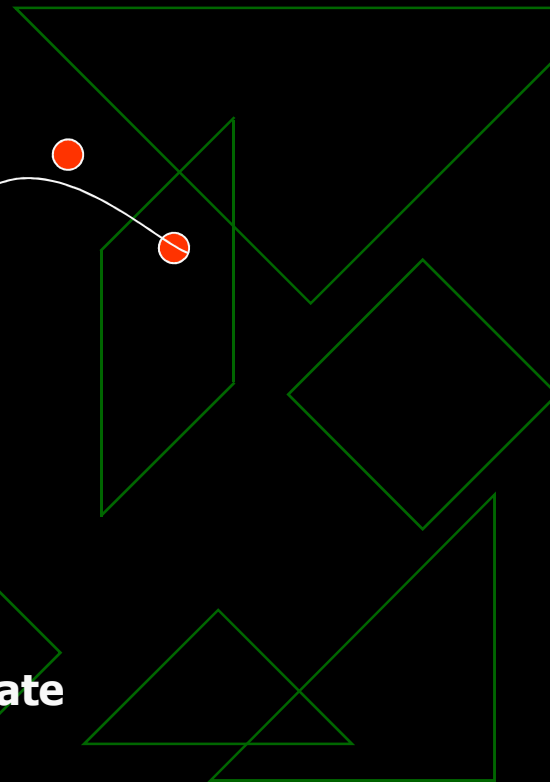
# Aproximação x Interpolação



**Interpolate**



**Approximate**

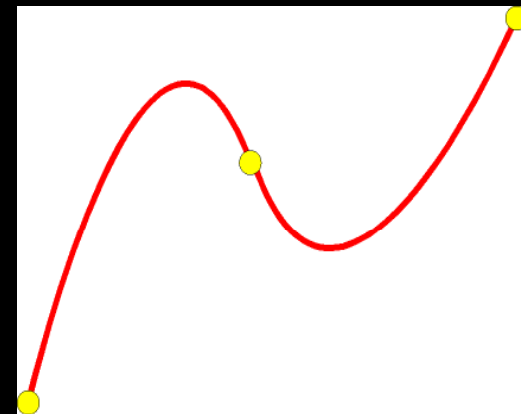




# Classificação

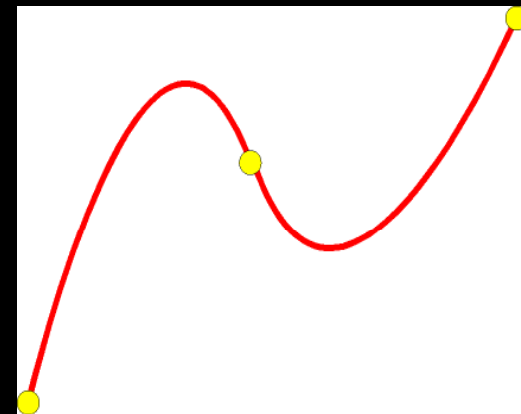
- ◆ Curvas

- apenas comprimento



# Como podemos representar uma curva?

- ◆ Localização no espaço de um ponto que se move
- ◆ Como podemos descrever este conceito?



# Possibilidades de Representações

- ◆ Algumas trajetórias podem ter mais de uma possibilidade de representação
- ◆ Exemplo: círculo centrado na origem com raio=1

$$x^2 + y^2 = 1$$

$$\begin{aligned}x(\theta) &= \cos \theta \\y(\theta) &= \text{sen } \theta\end{aligned}$$

# Tipos de Representação (2D)

◆ Explícita

$$y = f(x)$$

$$y = 3x^2$$

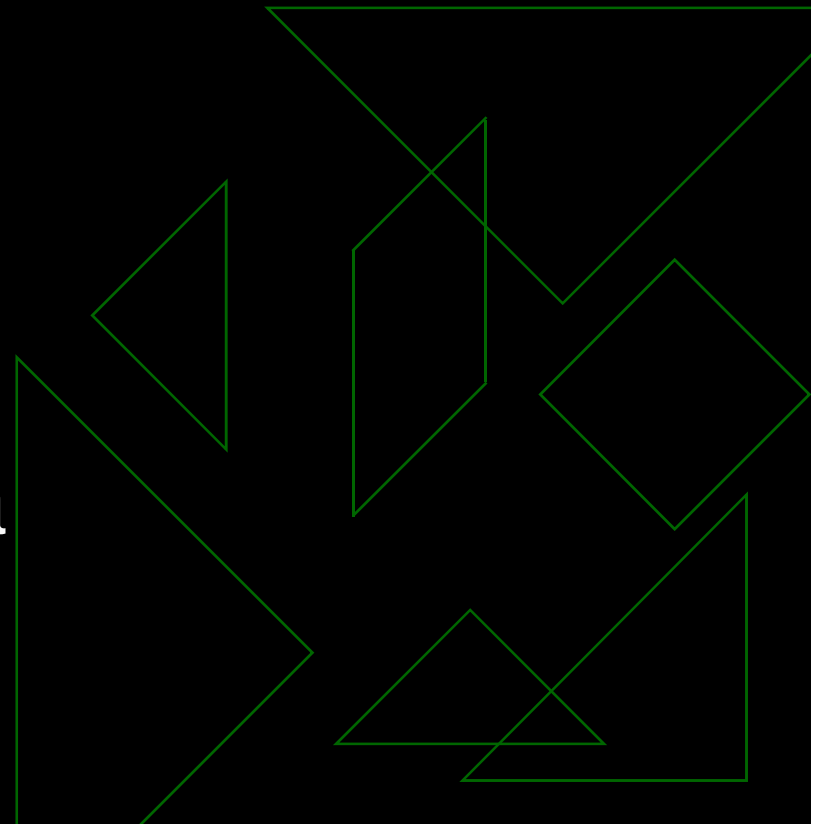
◆ Implícita

$$f(x, y) = 0$$

$$x^2 + y^2 = 0$$

# Tipos de Representação (2D)

- ◆ Explícita
- ◆ Implícita
- ◆ Em ambos os casos, as representações são dependentes do sistema de coordenadas (x,y)



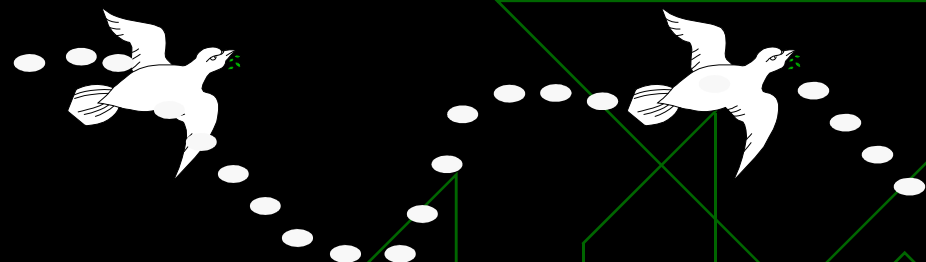
# Representação Paramétrica

$$x = x(t)$$

$$y = y(t)$$

$$x(t) = t^2$$

$$y(t) = t$$



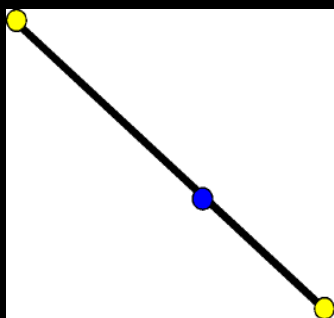
**Onde está o pássaro no tempo  $t$ ?**

**Dependentes de  $t$ ...**

# Exemplo: Funções de Base

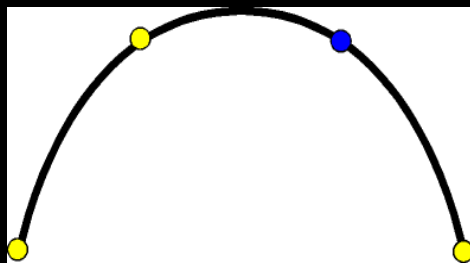
- ◆ Normalmente polinômios de grau 3
- ◆ Porque grau 3?
  - Grau 0 = nenhuma inflexão
  - Grau 4 = custo computacional

$$f(t) = at + b$$



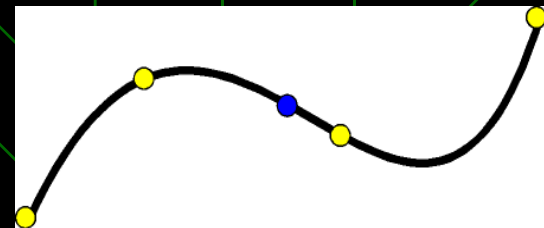
Linear

$$f(t) = at^2 + bt + c$$



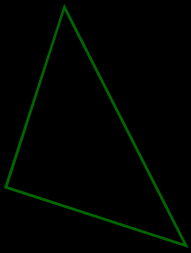
Quadratic

$$f(t) = at^3 + bt^2 + ct + d$$



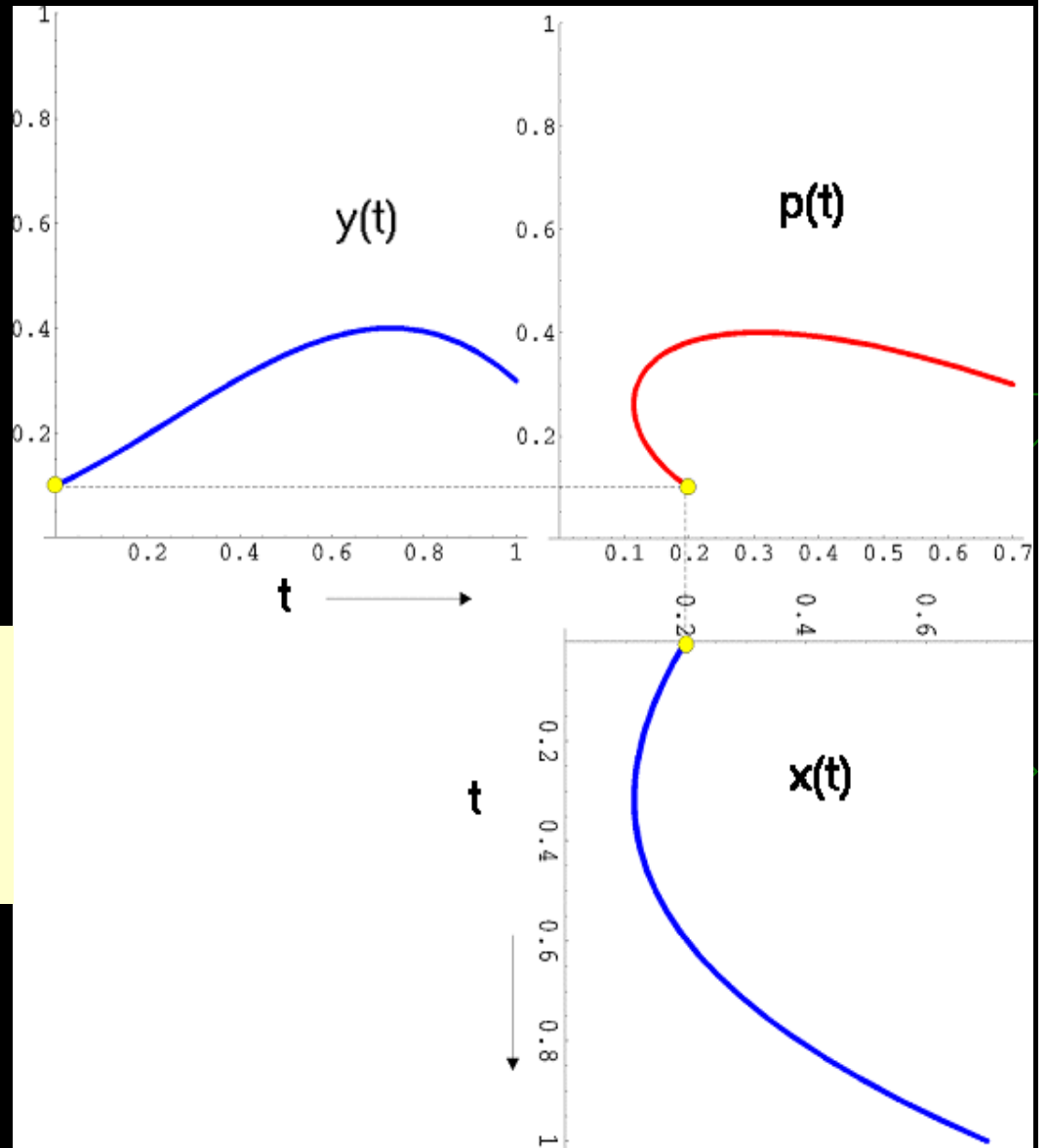
Cubic

# Curvas Paramétricas Cúbicas 2D



$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$

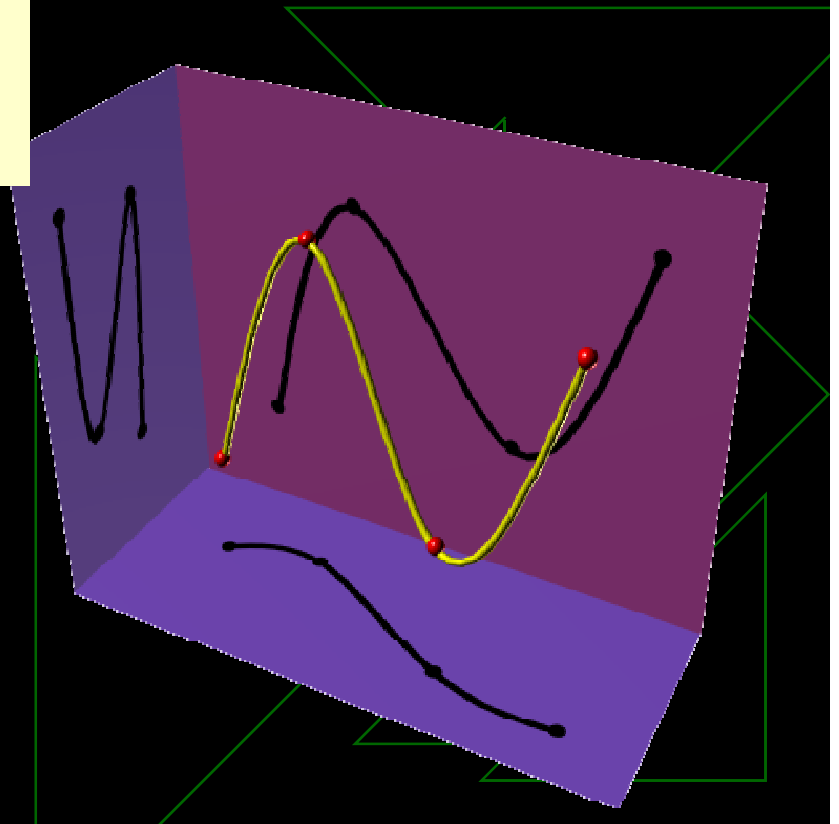
$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$






# Em 3D

$$\left. \begin{aligned} x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) &= a_z t^3 + b_z t^2 + c_z t + d_z \end{aligned} \right\}$$



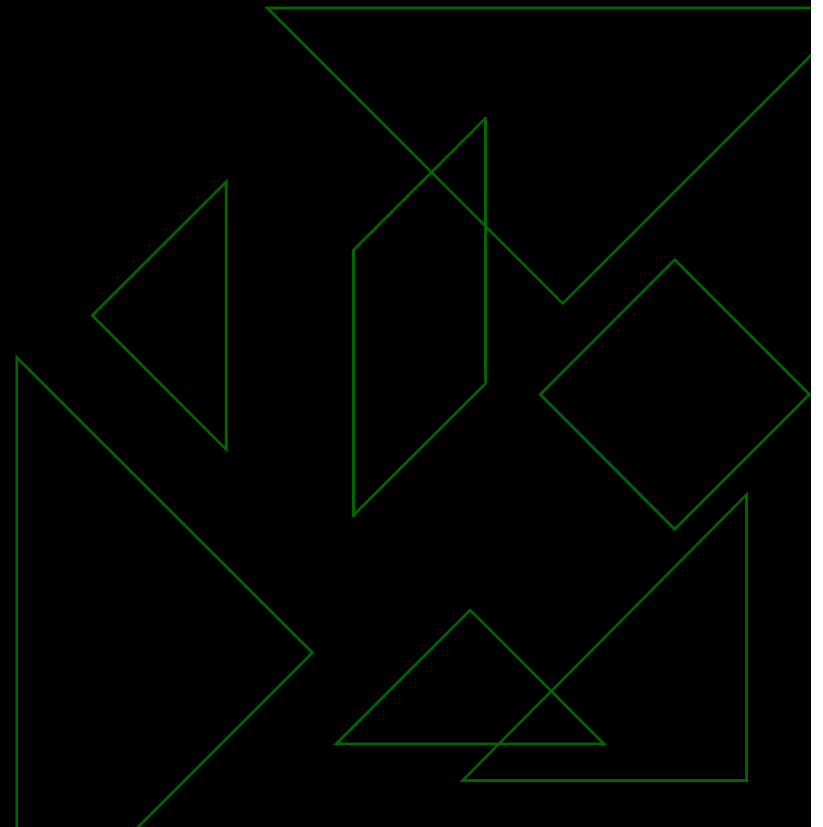


Mas, como podemos especificar os movimentos em função do tempo?

- ◆ Interpolação
  - ◆ Física
  - ◆ Performance driven Animation (PDA)
- 

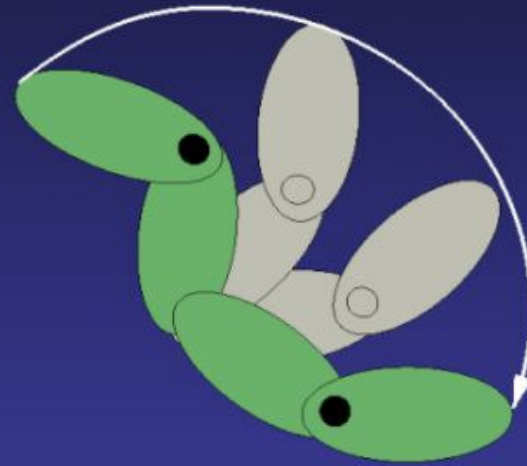
# Métodos de Interpolação

- ◆ Keyframe
- ◆ Animation Scripts
- ◆ 3D shape interpolation
- ◆ Animação Explícita

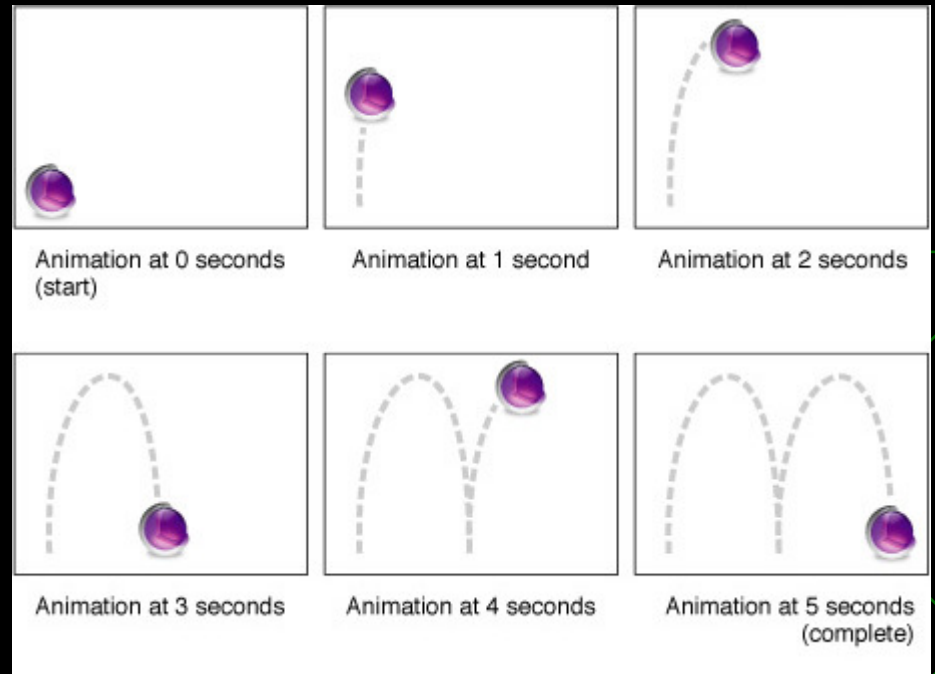
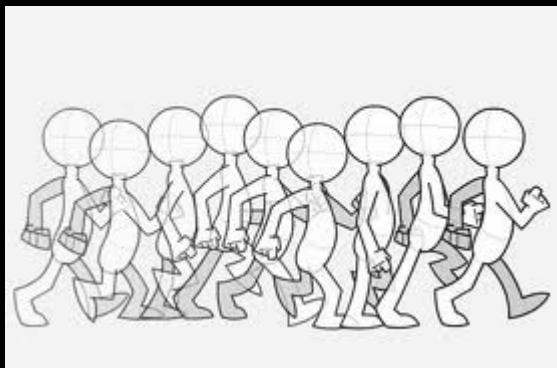
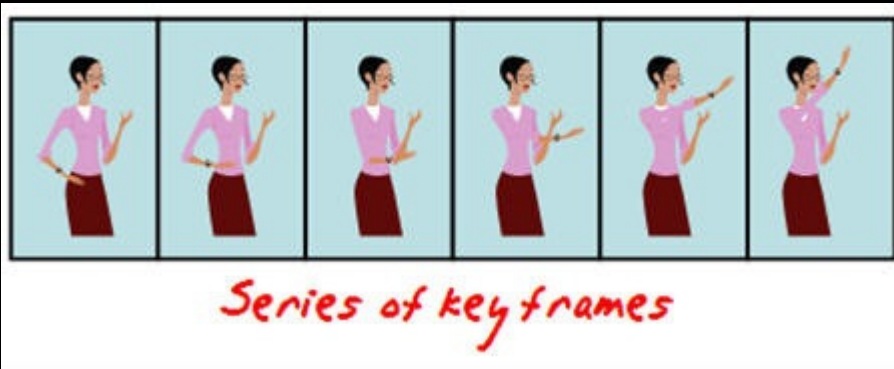
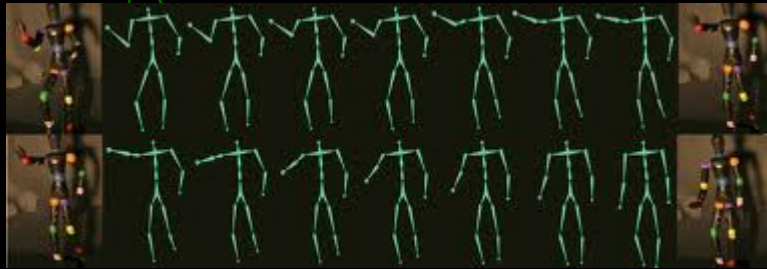


# Animation Techniques

## Keyframing

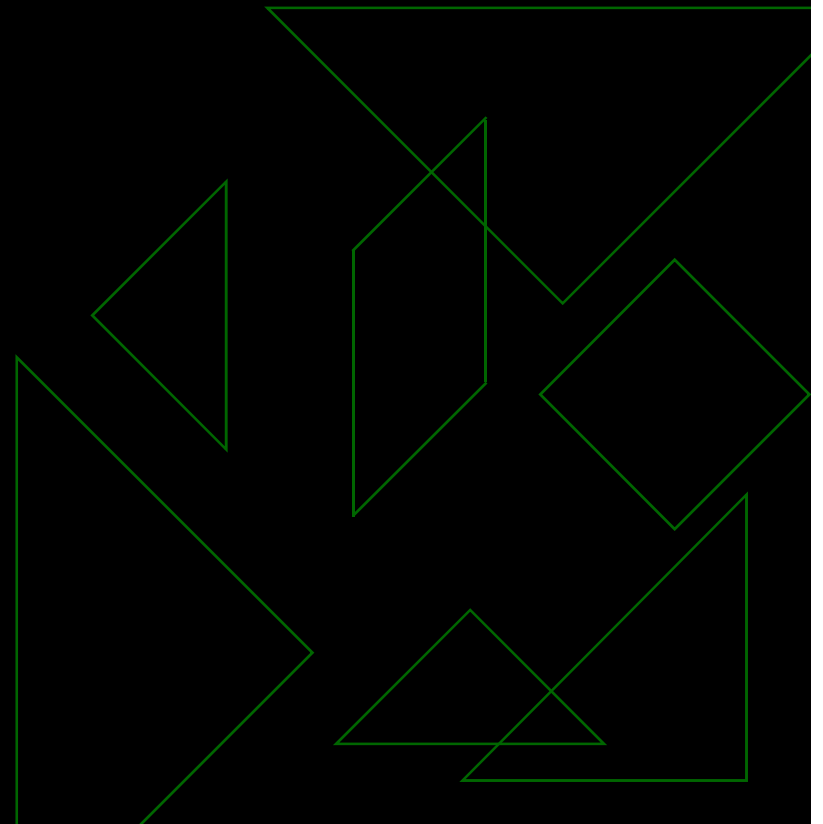


# Keyframe



# Automatic Camera Control

- ◆ The problem of camera location
- ◆ Intelligent camera



# Animation Scripts

- ◆ The **Maya Embedded Language (MEL)** is a **scripting language** used to simplify tasks in Autodesk's 3D Graphics Software **Maya**
- ◆ **MAXScript** is the built-in **scripting language** for Autodesk® **3ds Max**

# Examples of Animation Scripts

<http://www.youtube.com/watch?v=QqWmZI7BRRs>

```
int $c=0;
while ($c<=200) {
$c=$c+1;

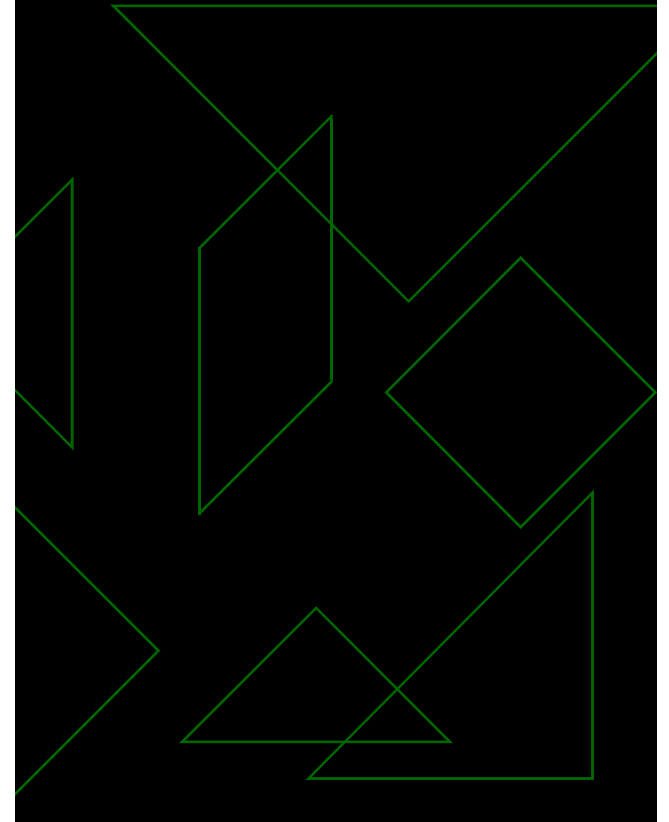
polyCube -w 11;
move 0 (2*$c) -5;
setKeyframe -v 0 -t 0 -at visibility;
setKeyframe -v 1 -t (2*$c) -at visibility;

polyCube -w 11;
move 0 (2*$c) 5;
setKeyframe -v 0 -t 0 -at visibility;
setKeyframe -v 1 -t (2*$c+1) -at visibility;

polyCube -w 11;
move 5 (2*$c+1) 0;
rotate 0 90 0;
setKeyframe -v 0 -t 0 -at visibility;
setKeyframe -v 1 -t (2*$c+2) -at visibility;

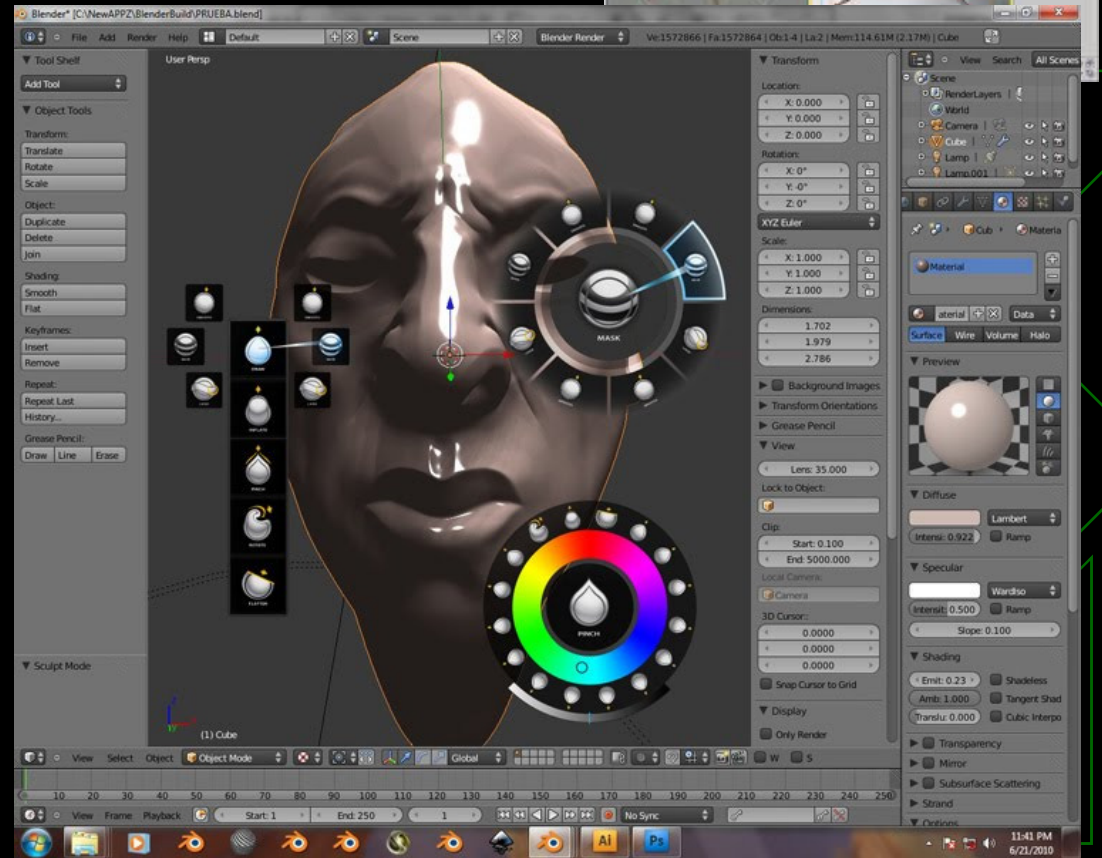
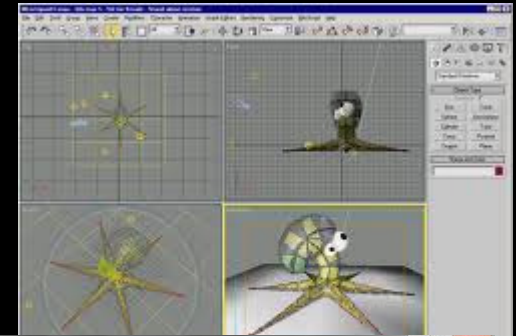
polyCube -w 11;
move -5 (2*$c+1) 0;
rotate 0 90 0;
setKeyframe -v 0 -t 0 -at visibility;
setKeyframe -v 1 -t (2*$c+3) -at visibility;
}
```

Word-wrap is OFF

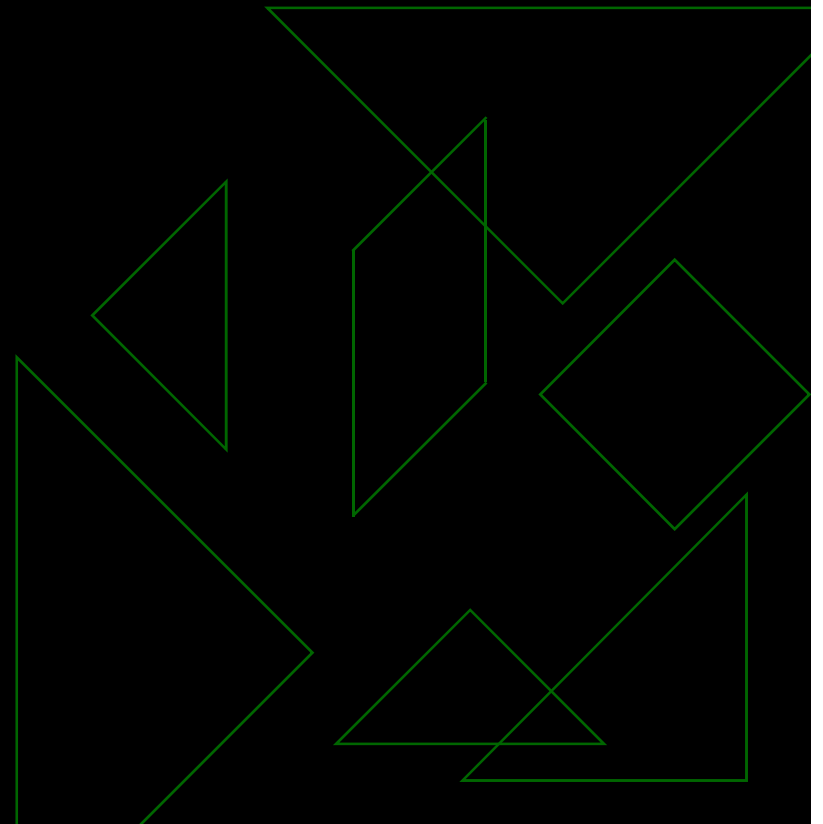




# Interfaces Gráficas



# **Interface gráfica para Animação de personagens**



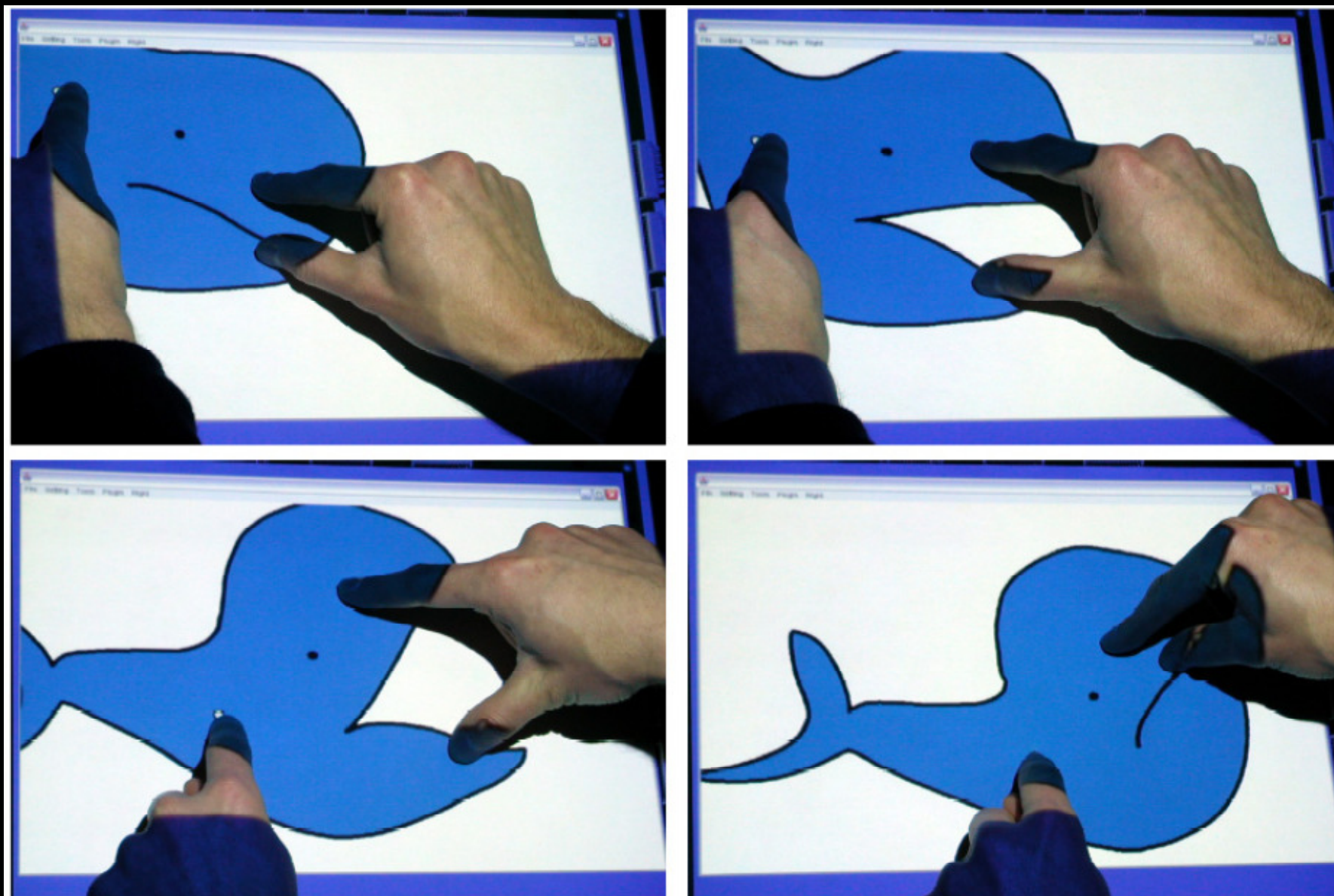
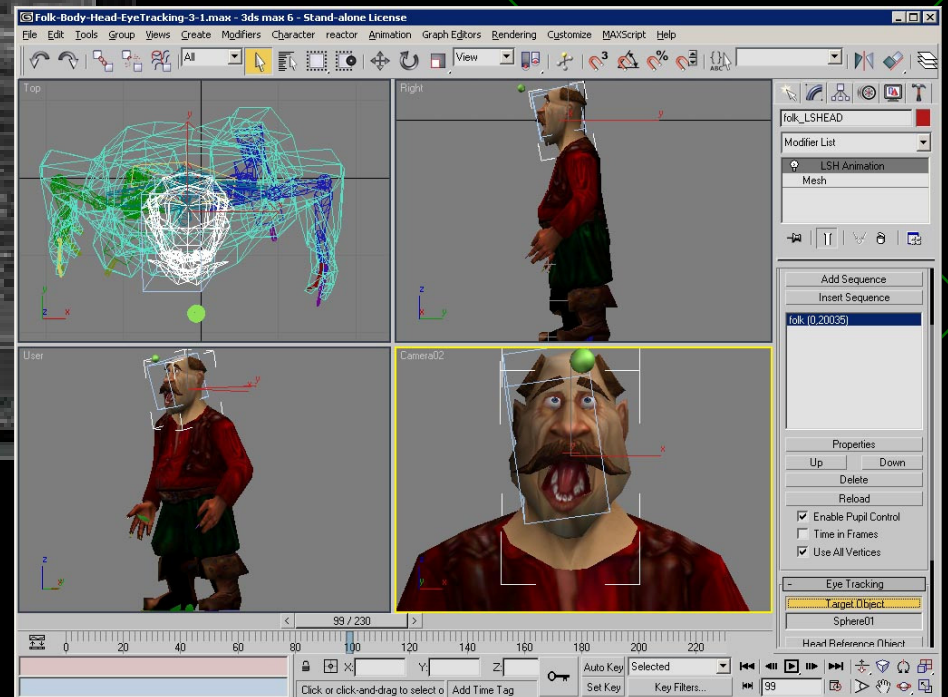


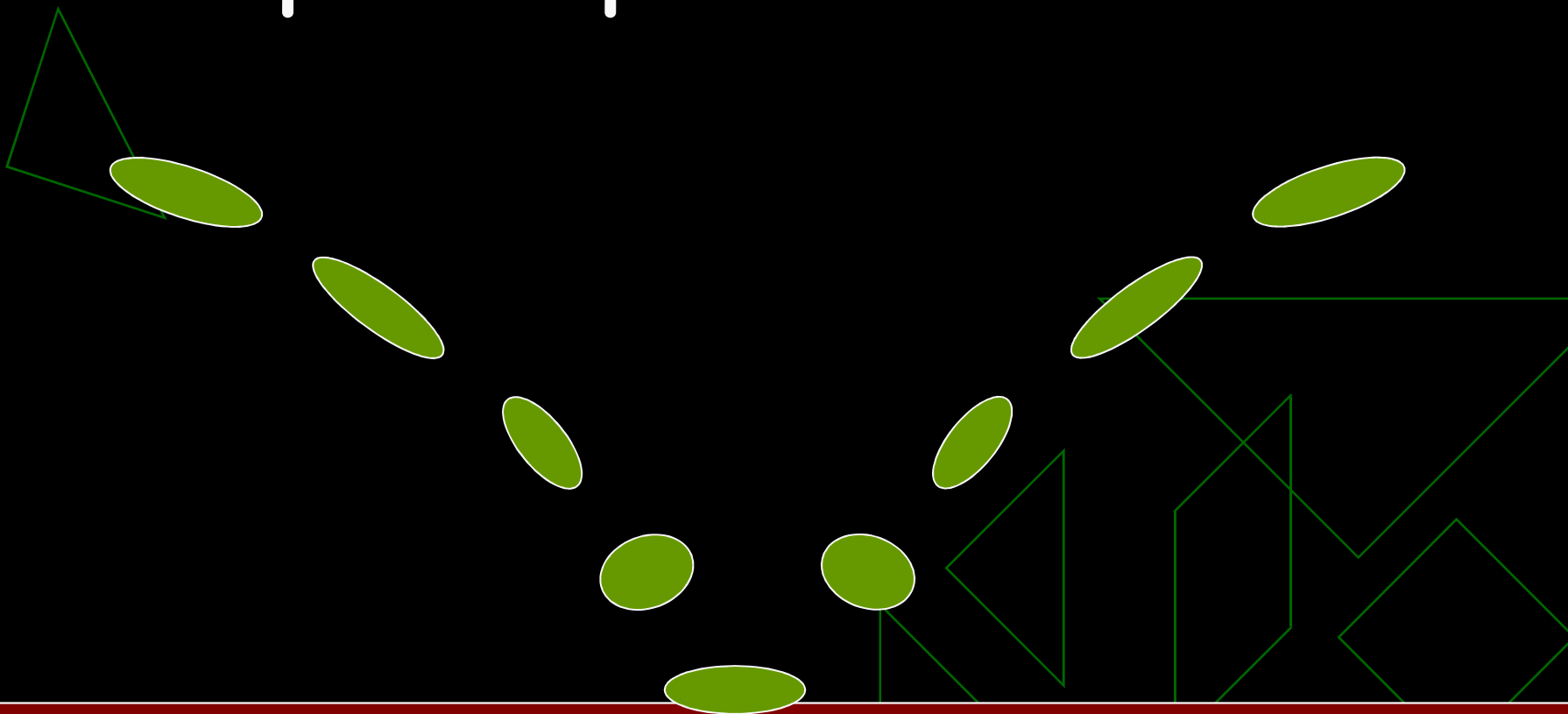
Figura 1: Manipulação da figura usando um SmartSkin. O usuário pode interativamente mover, rotacionar e deformar figuras com ambas as mãos, como se estivesse manipulando o objeto real.



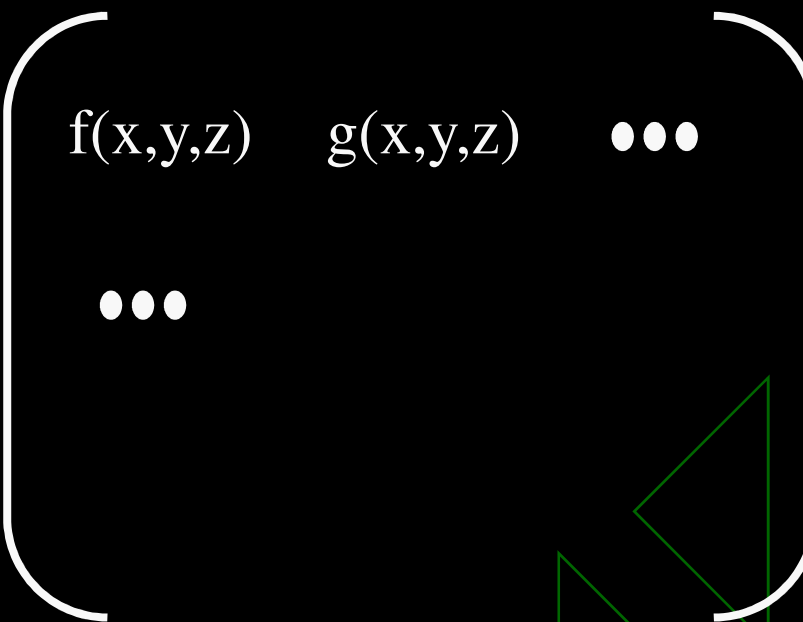
# Interface gráfica para Animação de personagens



# Shape interpolation

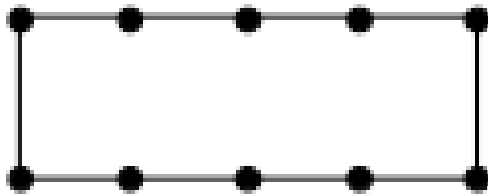


# Global Deformations

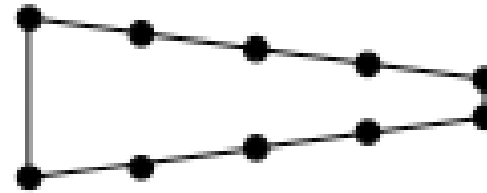

$$\begin{pmatrix} f(x,y,z) & g(x,y,z) & \dots \\ \dots & & \end{pmatrix}$$

Transformation matrix elements - functions of coordinates

# Global Deformations - taper



a) original object



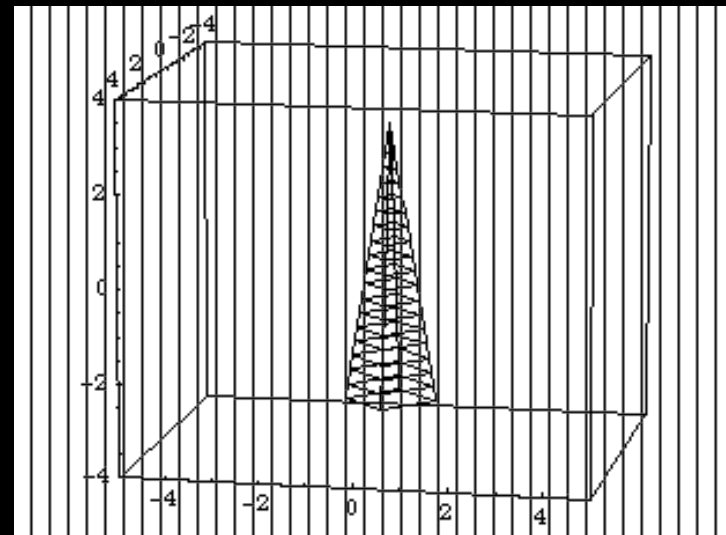
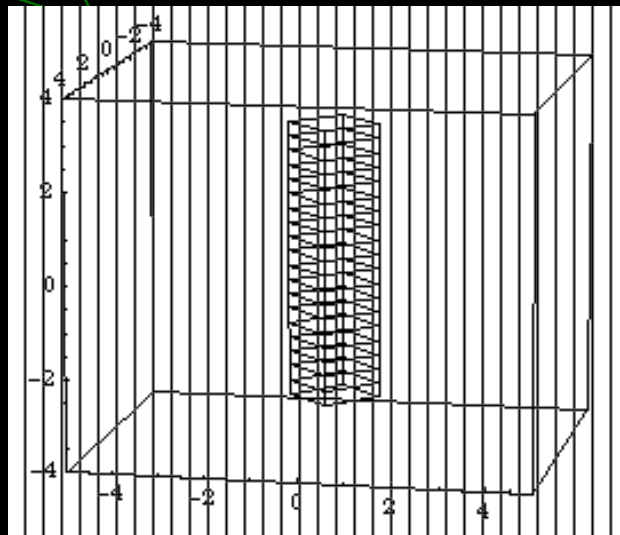
b) tapered object

$$\begin{aligned}x' &= x \\y' &= f(x)\end{aligned}$$

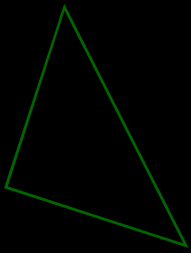
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & f(x) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$
$$P' = M(P) \cdot P$$



# Global Deformations - taper



# Global Deformations - twist



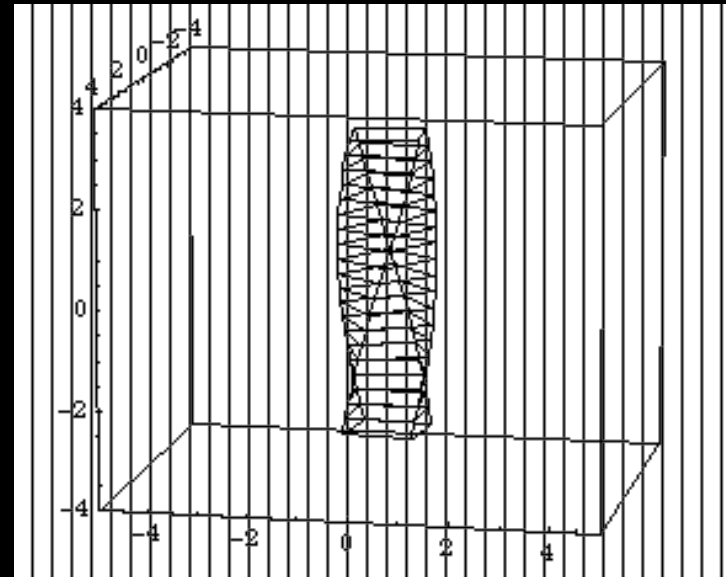
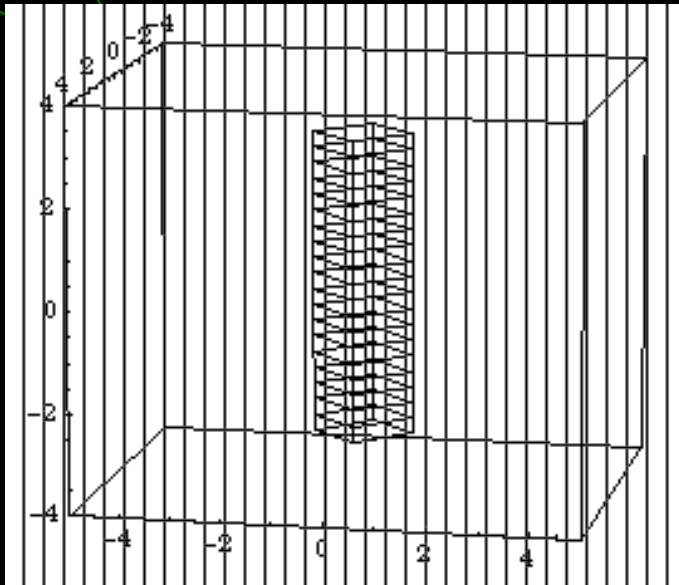
$$x' = x * \cos(f(y)) - z * \sin(f(y))$$

$$y' = y$$

$$z' = x * \sin(f(y)) + z * \cos(f(y))$$



# Global Deformations - twist



# Global Deformations - rotate

$y_0$  - center of bend  
 $1/k$  - radius of bend  
 $y_{min}$ - $y_{max}$  - bend region

$$\hat{y} = \begin{matrix} y_{min} & y \leq y_{min} \\ y & y_{min} < y < y_{max} \\ y_{max} & y \geq y_{max} \end{matrix}$$

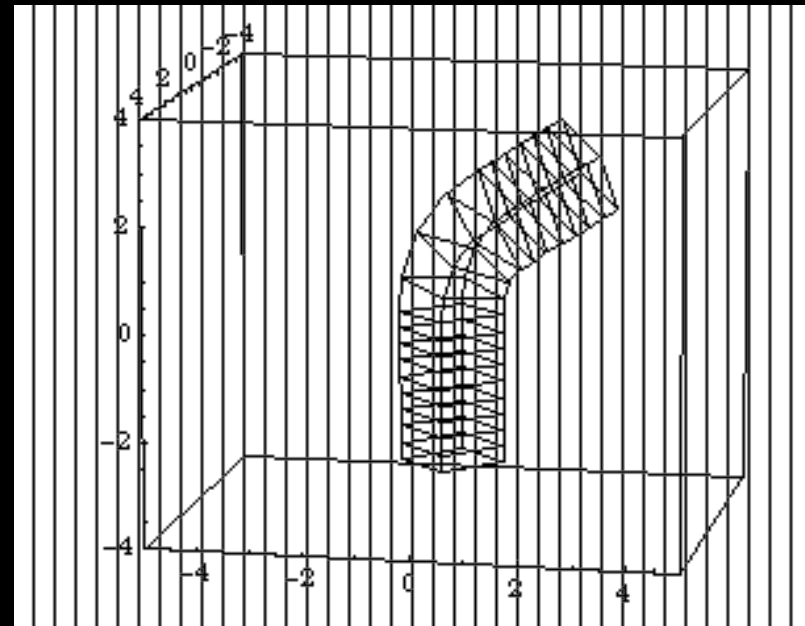
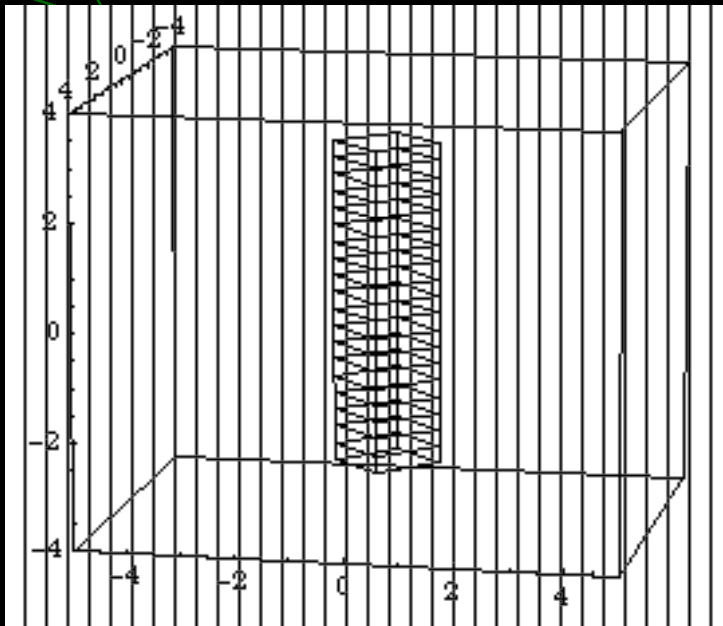
$$\begin{aligned} \theta &= k \cdot (\hat{y} - y_0) \\ C_\theta &= \cos\theta \\ S_\theta &= \sin\theta \end{aligned}$$

$$x' = x$$

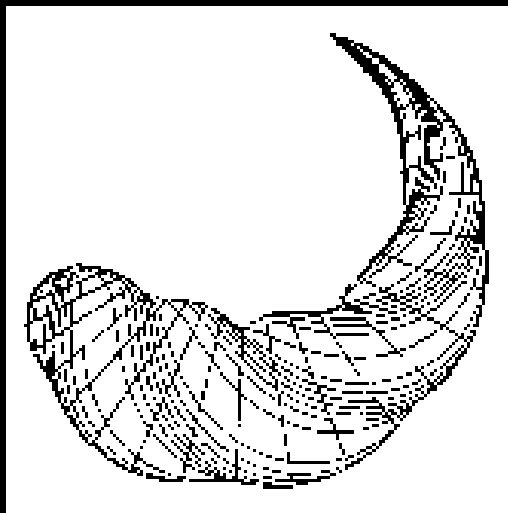
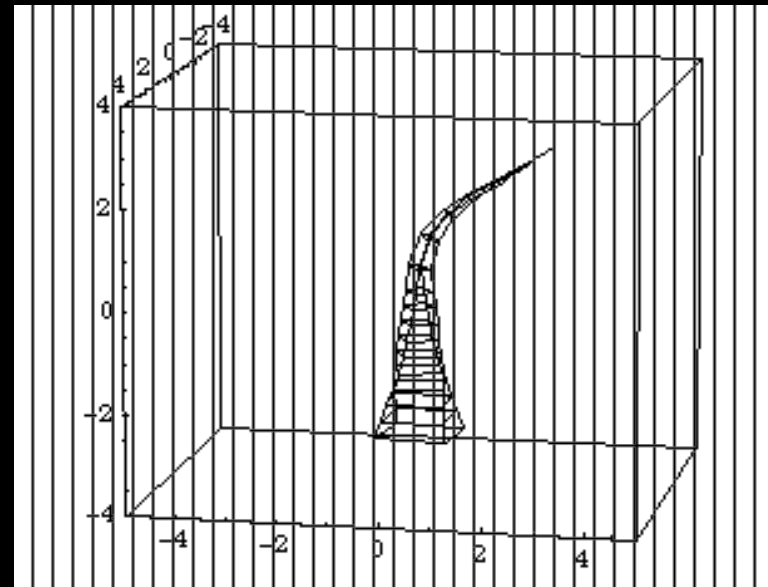
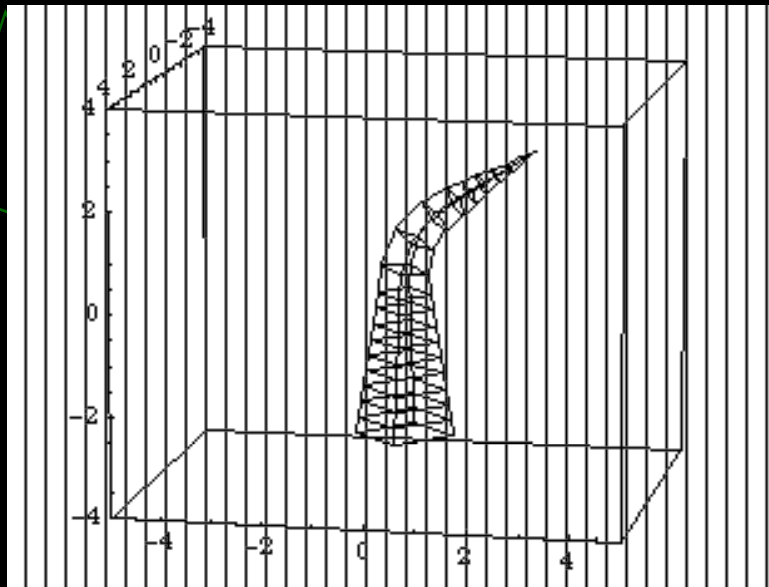
$$y' = \begin{cases} -S_\theta \cdot z - \frac{1}{k} + y_0 & y_{min} \leq y \leq y_{max} \\ -\left(S_\theta \cdot \left(z - \frac{1}{k}\right)\right) + y_0 + C_\theta \cdot (y - y_{min}) & y < y_{min} \\ \left(-\left(S_\theta \cdot \left(z - \frac{1}{k}\right)\right) + y_0 + C_\theta \cdot (y - y_{max})\right) & y \geq y_{max} \end{cases}$$

$$z' = \begin{cases} -C_\theta \cdot z - \frac{1}{k} + \frac{1}{k} & y_{min} \leq y \leq y_{max} \\ -\left(C_\theta \cdot \left(z - \frac{1}{k}\right)\right) + \frac{1}{k} + S_\theta \cdot (y - y_{min}) & y < y_{min} \\ \left(-\left(C_\theta \cdot \left(z - \frac{1}{k}\right)\right) + \frac{1}{k} + S_\theta \cdot (y - y_{max})\right) & y \geq y_{max} \end{cases}$$

# Global Deformations - rotate

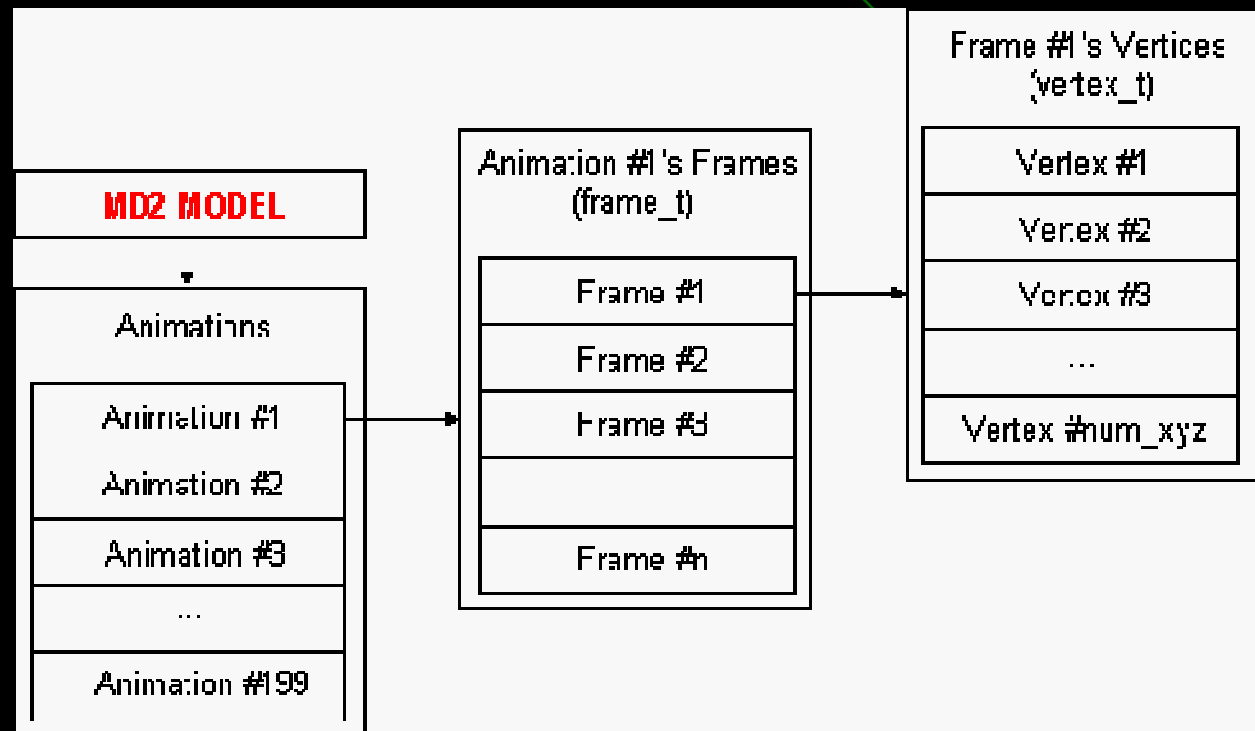


# Global Deformations - compound



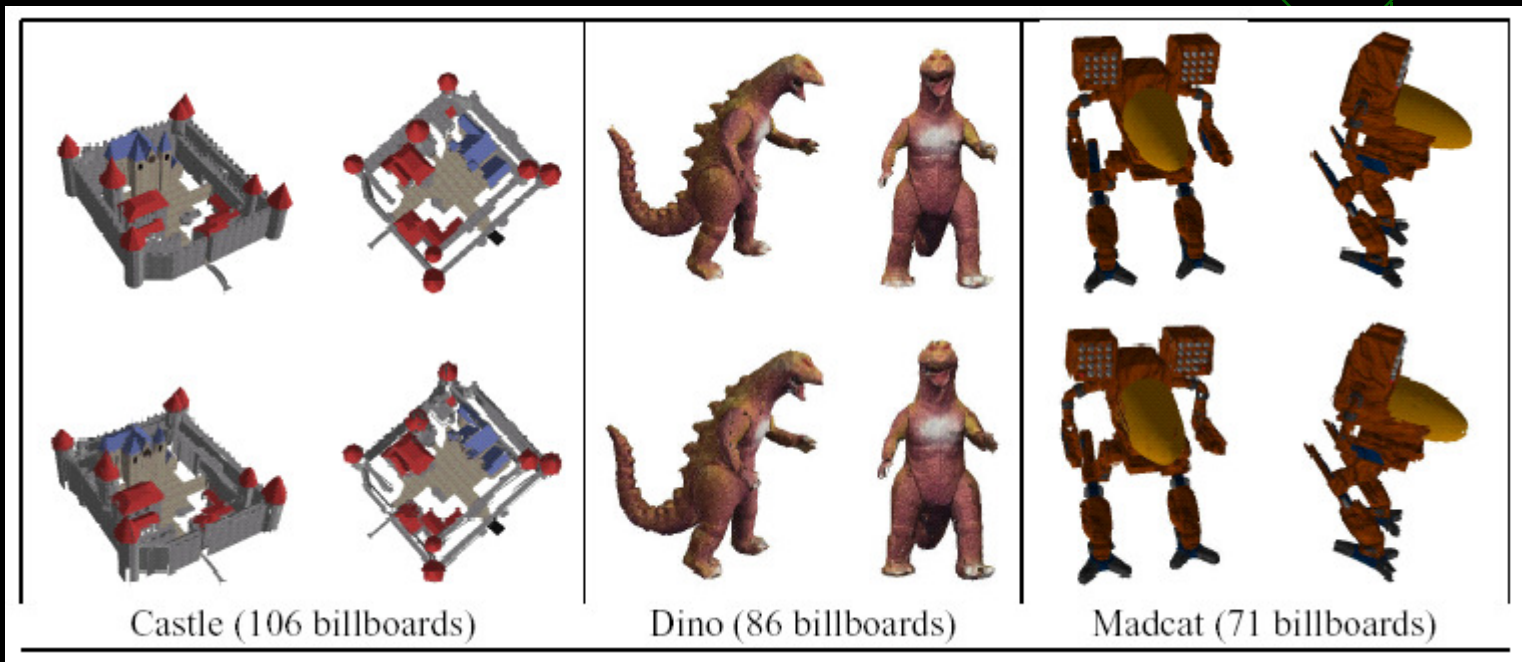
# Animação Pré-processada

- ◆ Pré-processada (ex. MD2 – Quake)
  - Lista de vértices que variam em função do tempo



# Animação Pré-processada

- ◆ Sprites animados
- ◆ Billboards, impostores





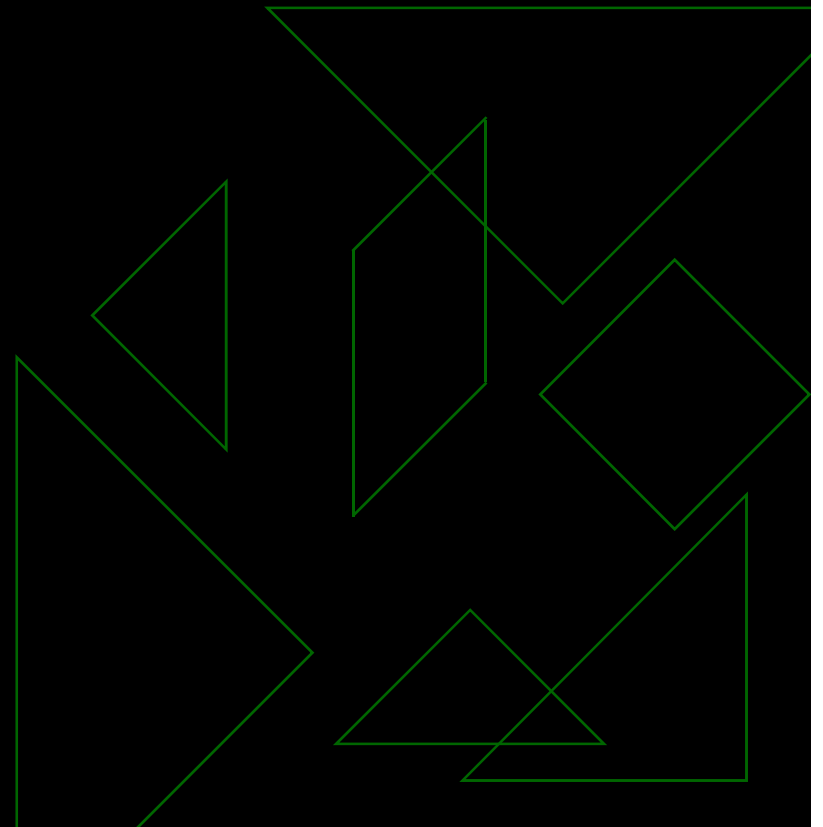
# Hamster

- ◆ 15,000 usable sprite frames, or 30,000 after mirroring



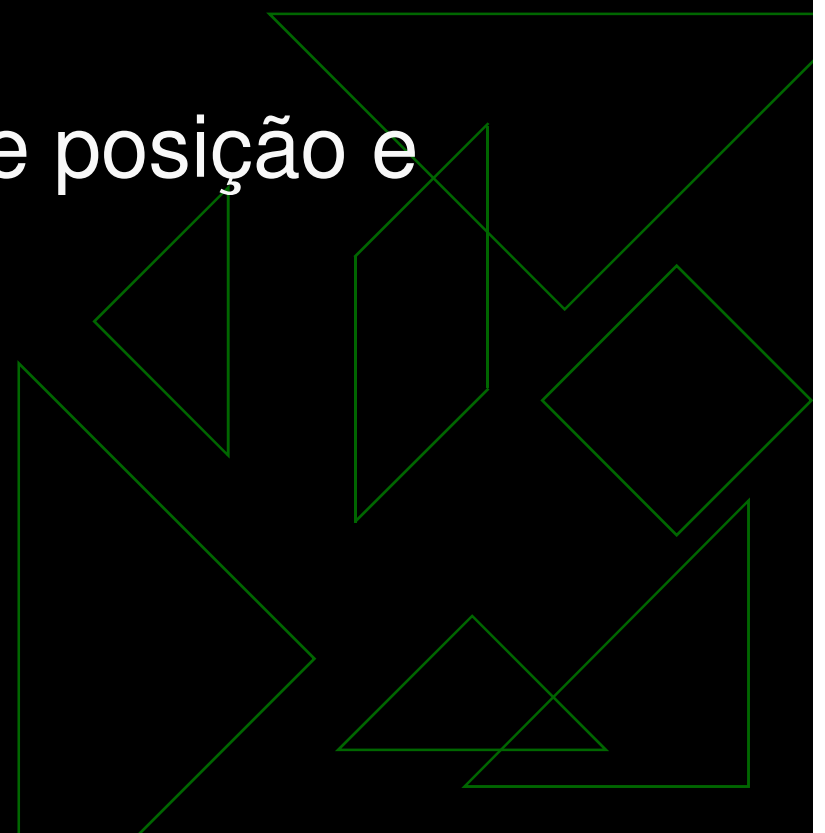
# Animacão baseada em Física

- ◆ Partículas
- ◆ Corpos rígidos
- ◆ Corpos deformáveis
- ◆ Corpos articulados



# Sistema material

- ◆ Composto por partículas ligadas entre si por relações
- ◆ Movimento: variação de posição e orientação



# Tipos de movimento

- ◆ Movimento interno (variação de distância entre os pontos internos) – Não existe para corpos rígidos
- ◆ Movimento externo (variação de distância entre os pontos externos)

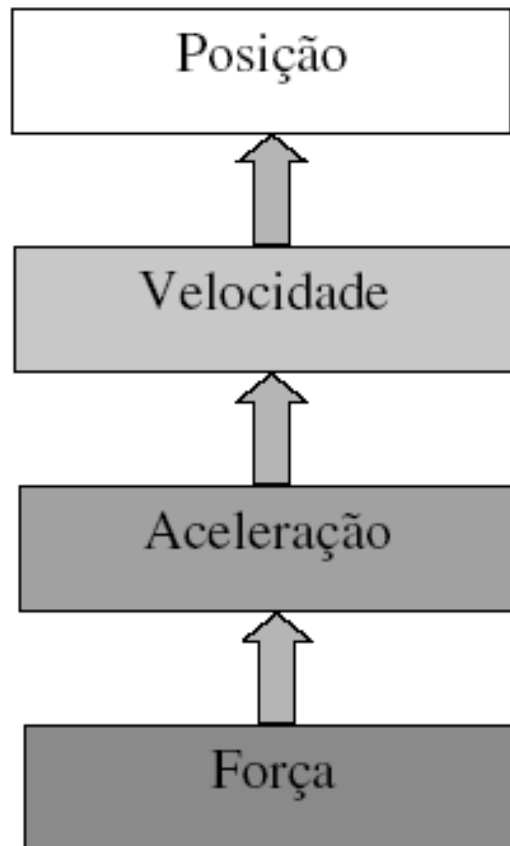
# Partículas

- ◆ São corpos que não possuem dimensão
- ◆ Só possuem movimento translacional (não tem CM)



# Física

- Animação → Mudança da Posição ao longo do tempo:



$$\mathbf{x} = (x, y)$$

$$\mathbf{v} = (v_x, v_y)$$
$$x = x_0 + v \cdot dt$$

$$\vec{v} = \frac{d\vec{x}}{dt} = \dot{\vec{x}}$$

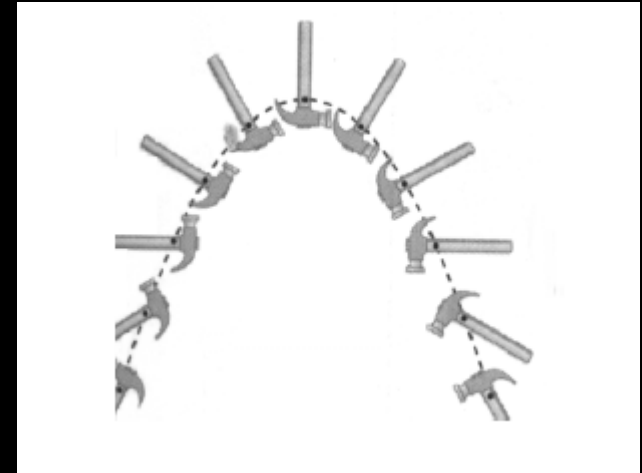
$$\mathbf{a} = (a_x, a_y)$$
$$v = v_0 + a \cdot dt$$

$$\vec{a} = \frac{d\vec{v}}{dt} = \ddot{\vec{x}}$$

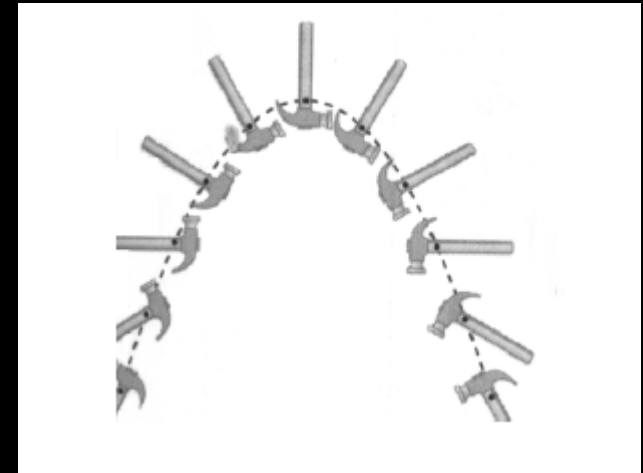
$$\mathbf{F} = m \cdot \mathbf{a}$$

# Corpos rígidos

- ◆ Corpos com massa
- ◆ Possuem movimento translacional e rotacional
  - Translacional (como se houvesse somente o CM)
  - Rotacional: física considerando o torque
  - Não possui movimento interno



# Corpos rígidos



- ◆ Dados do modelo:
  - CM (vetor que descreve a localização do centro de rotação livre)
  - Dimensão (x,y,z)
  - Massa total
  - Distribuição da massa (simétrico ou não)
    - ◆ Para simétricos: calcula movimento translacional
    - ◆ Para assimétricos: calcula produto de inércia



# Corpos Rígidos

Translação do  
Centro de Massa

Posição

Velocidade

Aceleração

Força

Rotação do  
Corpo Rígido

Ângulo

Velocidade angular

Aceleração angular

Torque

$\theta(t)$

$\omega$

$$d\theta/dt = f(\omega)$$

$\alpha$

$$\omega = \omega_0 + \alpha \cdot dt$$

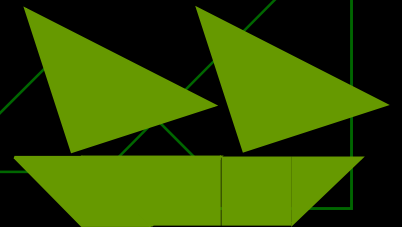
$$\tau = I \cdot \alpha$$
$$\tau = \mathbf{F} \times \mathbf{r}$$

# Exemplos de Técnicas de Animação

- ◆ Motion Control
  - Cinemática, dinâmica

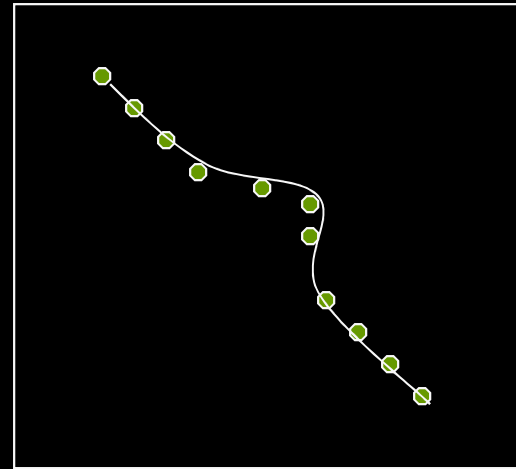
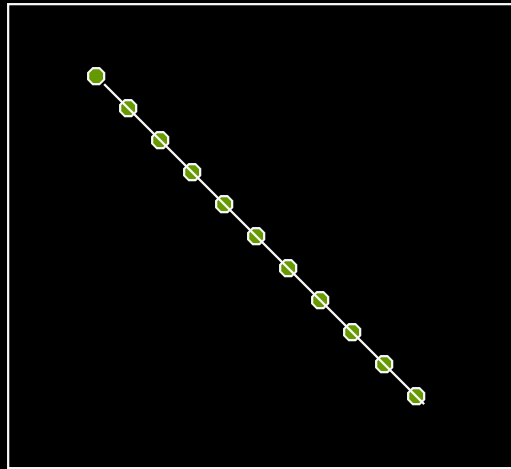


Kacic (2003)



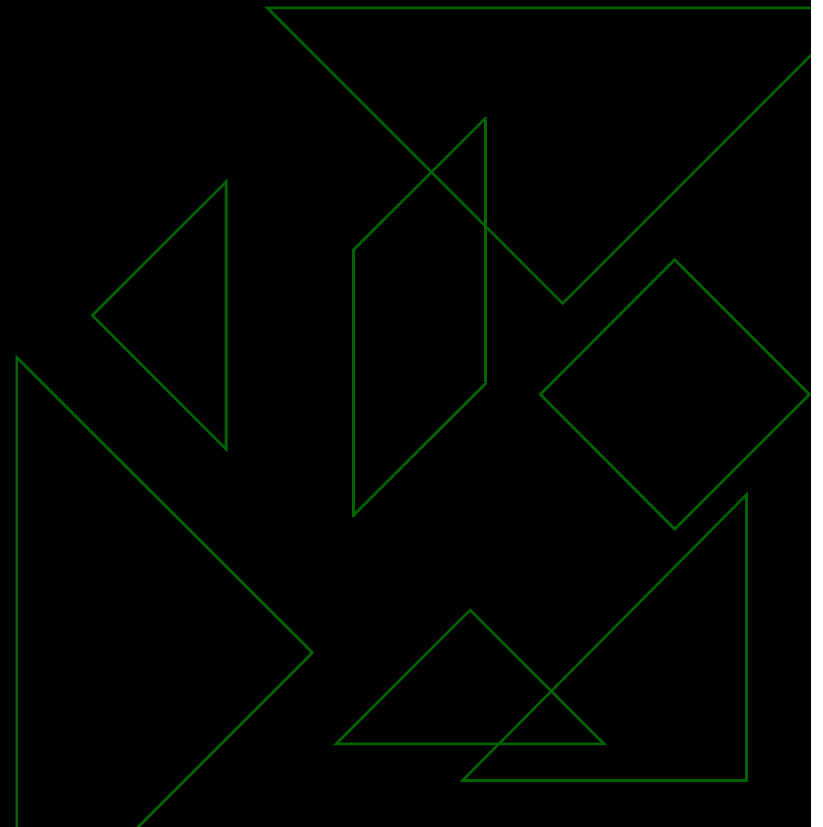
# Técnicas de Animação

## *Corpos deformáveis*



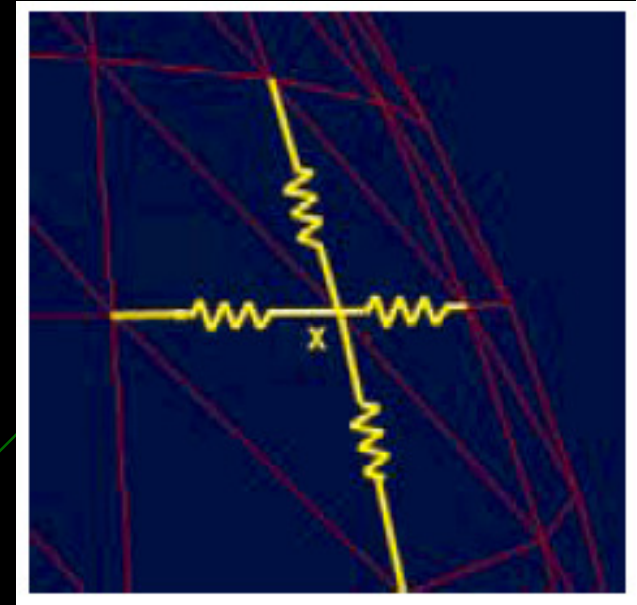
# Corpos flexíveis

- ◆ Formado por partículas com 3 DOFs translacionais
- ◆ Existe movimento interno
- ◆ Métodos:
  - Deformação física



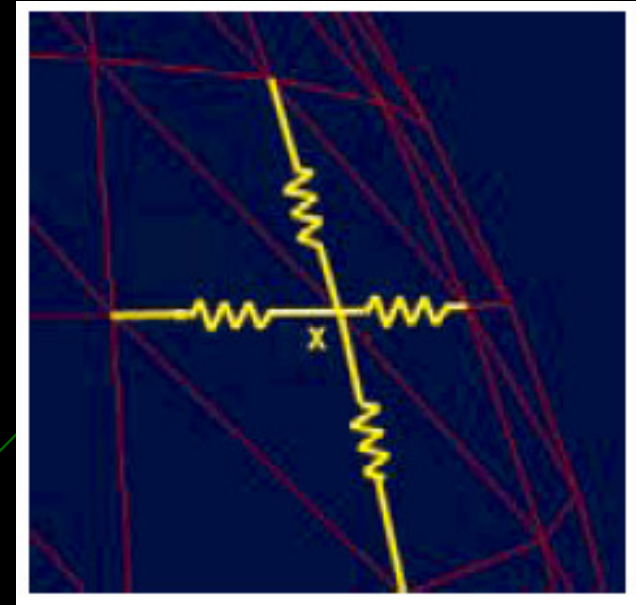
# Deformação física: Sistemas Massa-mola

- ◆ Cada vértice representa um ponto de massa
- ◆ Cada aresta representa uma mola
- ◆ O comprimento de repouso das molas corresponde ao comprimento da aresta no instante inicial
- ◆ Atribui-se uma massa a um objeto e esta é distribuída entre seus pontos
- ◆ As constantes das molas também são atribuídas pelo usuário (normalmente usa-se uma única)



# Sistemas Massa-mola

- ◆ Forças externas são aplicadas ao objeto globalmente (gravidade, vento, etc.) ou a um vértice específico (forças específicas), forçando seu deslocamento individual
- ◆ •Problemas:
  - *o efeito da aplicação de uma força externa se propaga lentamente pelo objeto ( $\Delta t$ )*
  - *Número de vértices e comprimento das arestas influenciam no resultado final*
  - *Distribuição dos vértices também influencia*



# Sistema Massa-mola: exemplo

## ◆ Instante t0

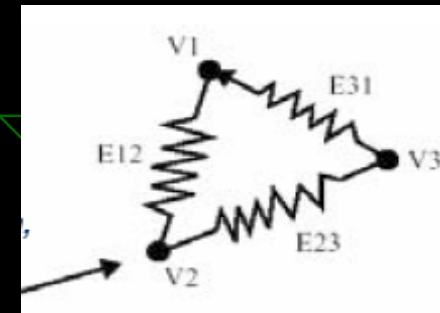
- A força momentânea  $F$  é aplicada sobre  $V2$
- Calcula-se a aceleração em  $V2$  e, em seguida, a velocidade e nova posição em  $V2$

## ◆ Instante t1

- $V2$  sofre força das molas  $E12$  e  $E23$
- $V1$  sofre força da mola  $E12$
- $V3$  sofre força da mola  $E23$

## ◆ Instante t2

- $V2$  sofre força de  $E12$  e  $E23$
- $V1$  sofre força de  $E12$  e  $E31$
- $V3$  sofre força de  $E23$  e  $E31$



$$F_{\text{MOLA}i} = -k_m \cdot (P_f - P_r)$$

\*  $-k_m$  = elasticidade da mola

\*  $P_f$  = Ponto de extremidade fixa

\*  $P_r$  = Ponto de repouso

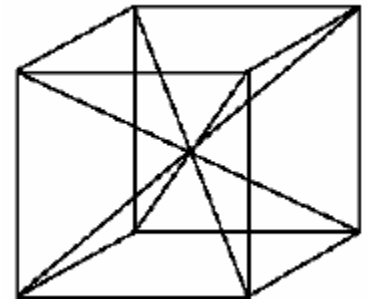
# Amortecedores:

- ◆ Cria uma força na direção oposta a força gerada pela mola e é proporcional a velocidade.



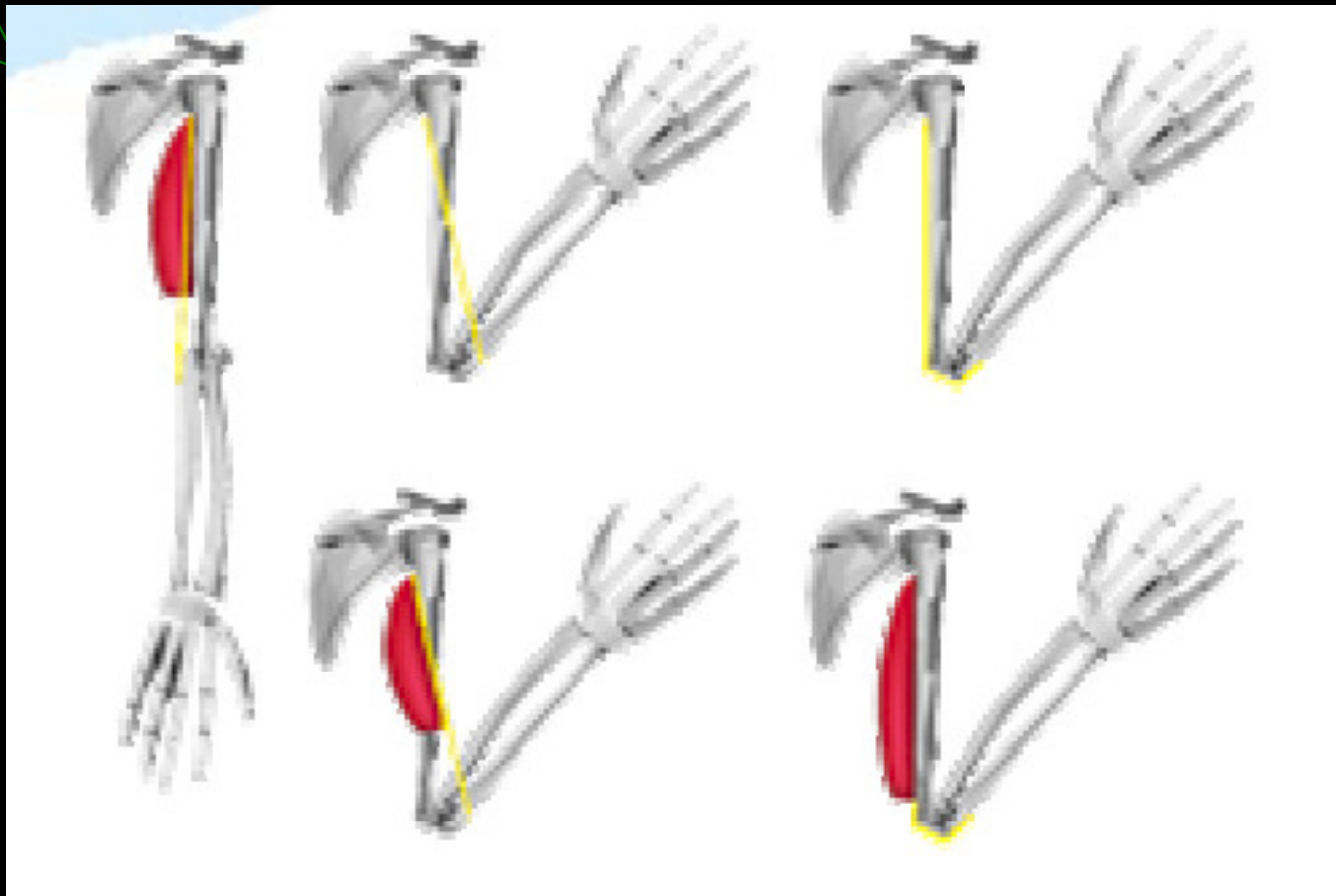
$$F_i^{\text{damper}} = -k_d \cdot v_i(t)$$

- ◆ Podem ser usados como uma forma de controlar o volume de um sólido



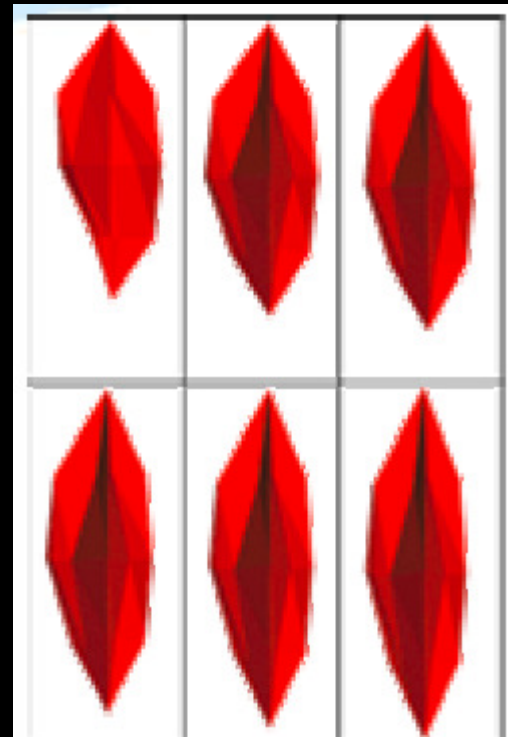
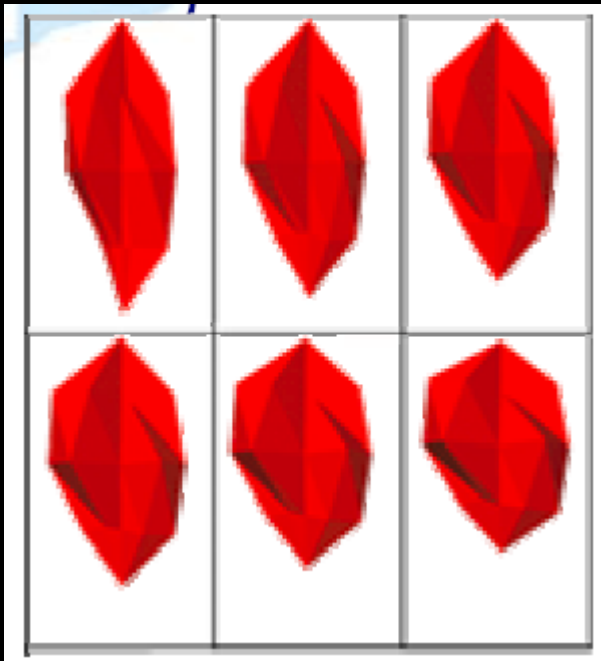


# Exemplo em Músculos:



# Exemplo de modelo completo de forças

◆  $F_{\text{result}} = F_{\text{grav}} + F_{\text{ext}} + F_{\text{elast}} + F_{\text{curvatura}} + F_{\text{colisão}} + F_{\text{restrição}} + F_{\text{atrito}}$



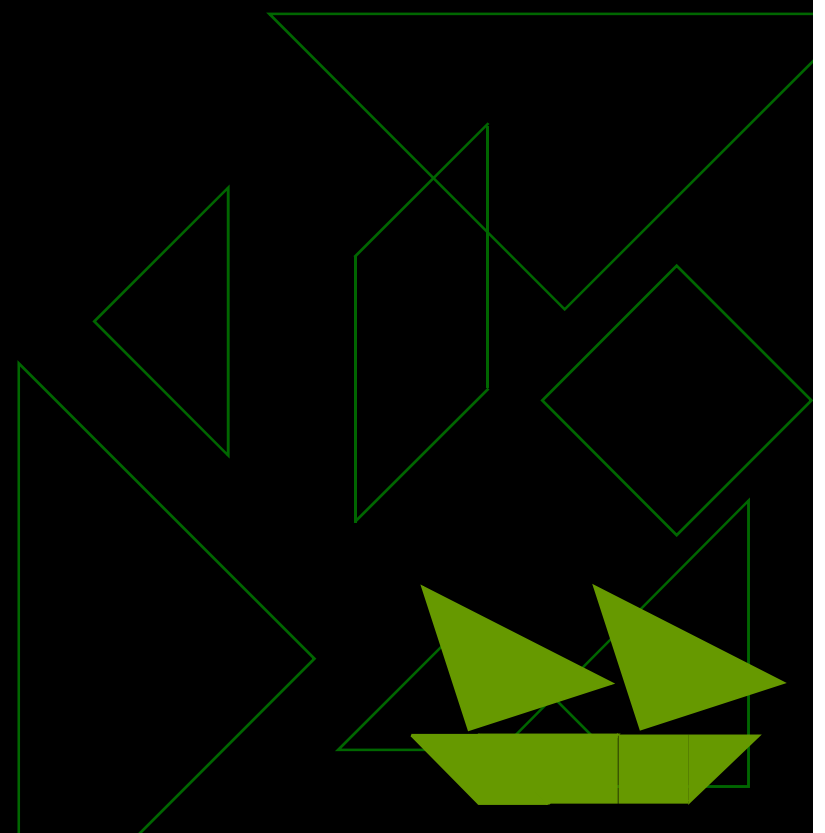
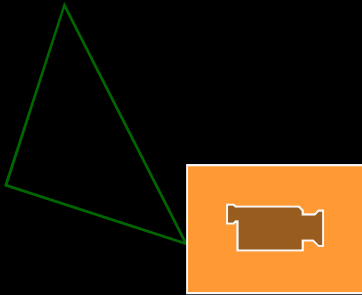
# Técnicas de Animação

## *Superfícies Flexíveis*

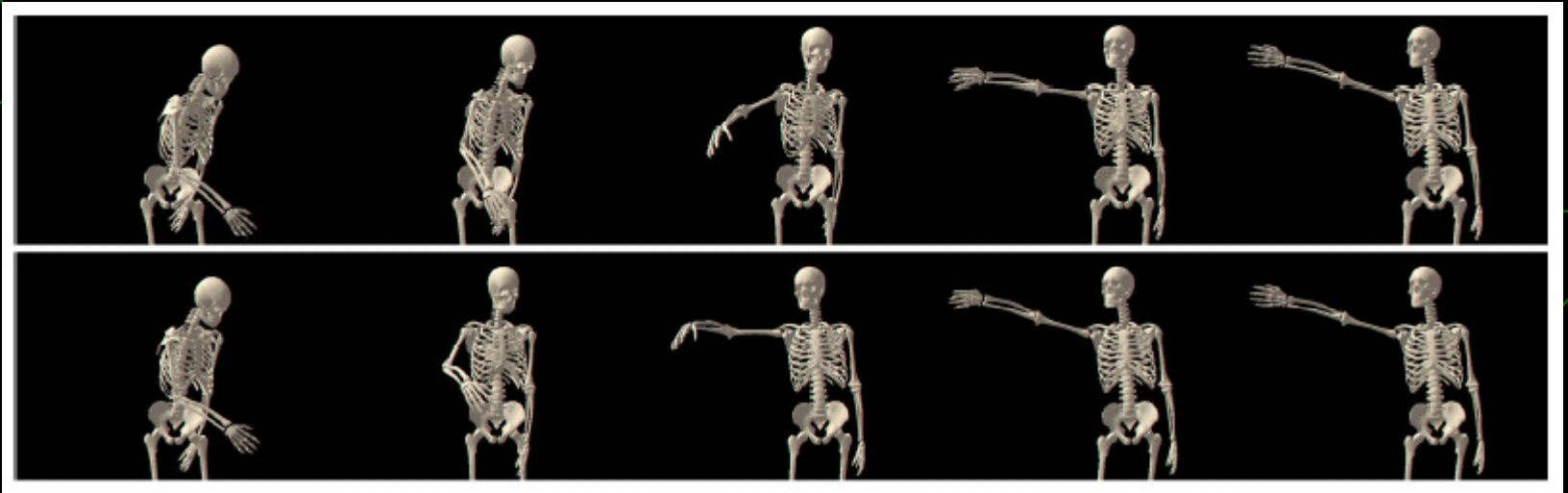


*A Bela e a Fera*

# Técnicas de Animação



# Corpos Articulados



Vídeo: TwistSequence

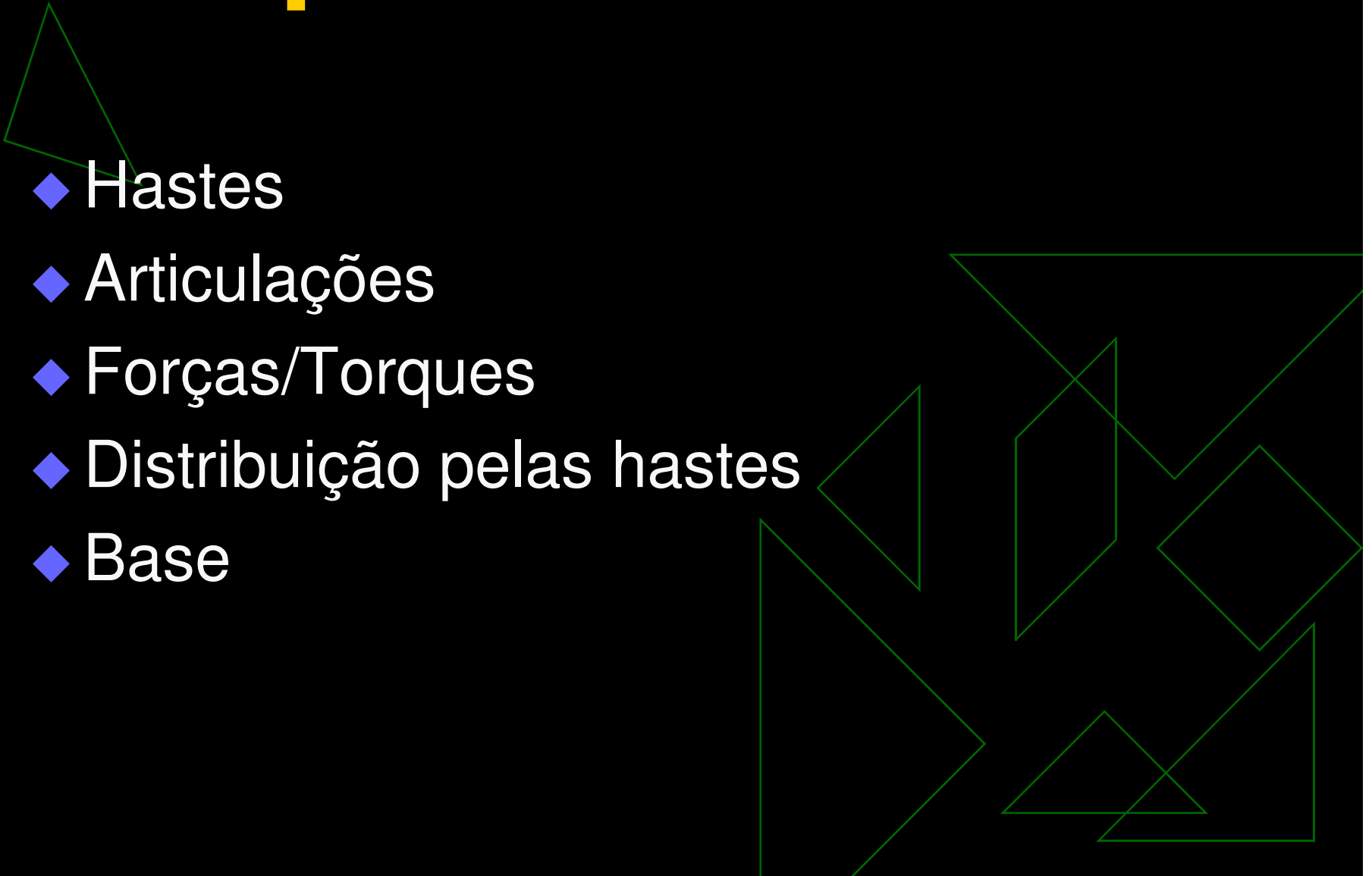
Customização de captured motion



Twist (2003)

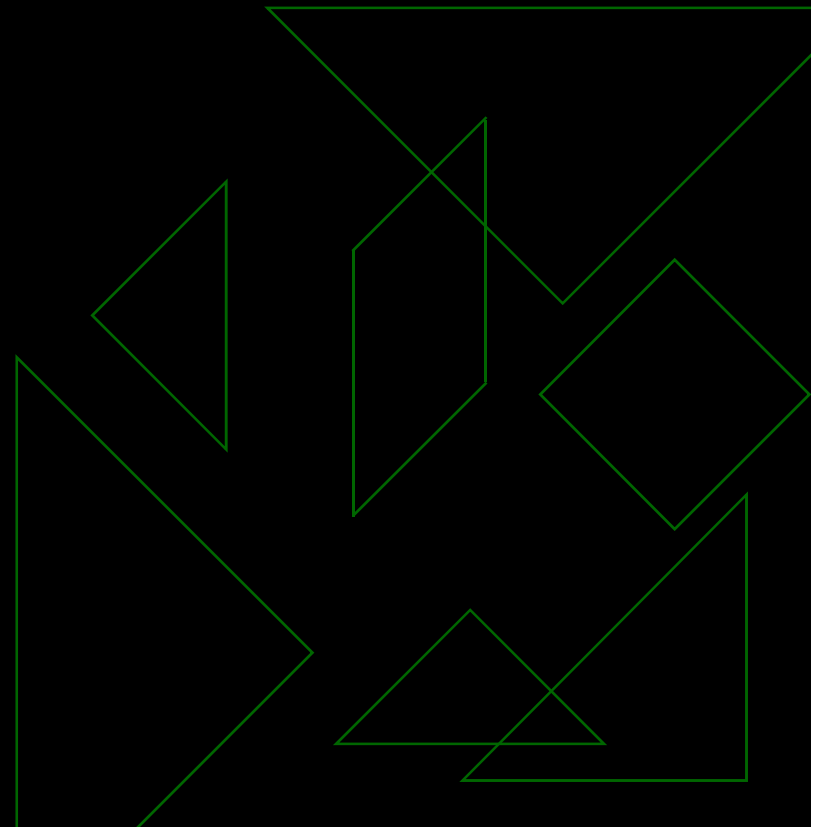
# Corpos Articulados

- ◆ Hastes
- ◆ Articulações
- ◆ Forças/Torques
- ◆ Distribuição pelas hastes
- ◆ Base



# Corpos Articulados

- ◆ Sistema livre
- ◆ Sistema Vinculado
- ◆ Vínculos:
  - Internos
  - Externos



# Corpos Articulados

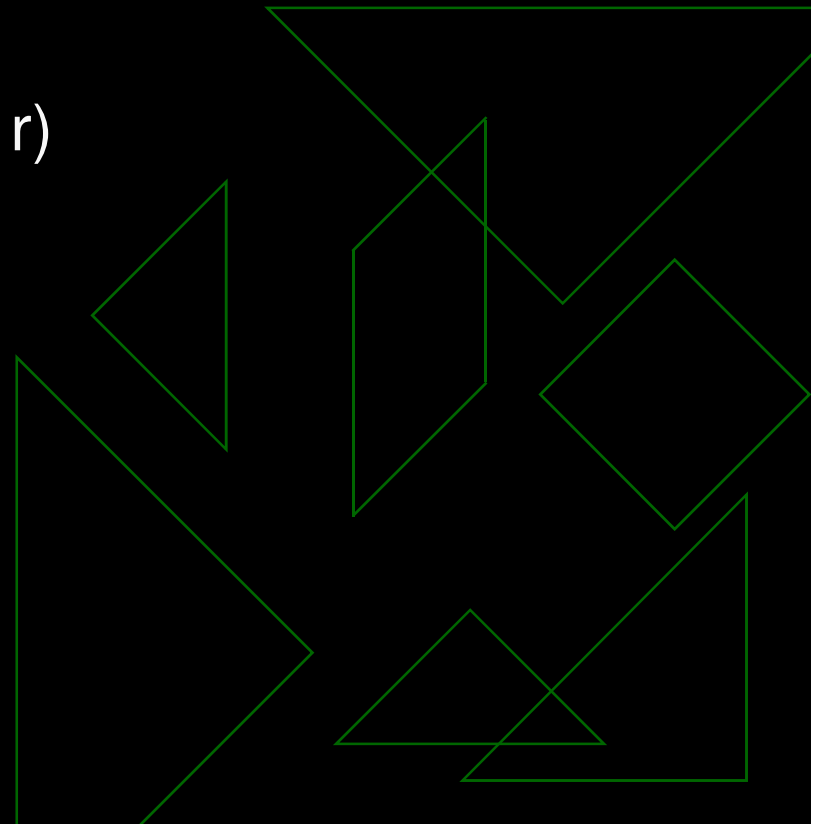
- ◆ Tipos de vínculos:
  - Vínculos translacionais
    - ◆ Apoio simples (1 DOF t)
    - ◆ Apoio duplo (2 DOF t)
    - ◆ Livre (3 DOF t)
    - ◆ Sem translação (0 DOF t)





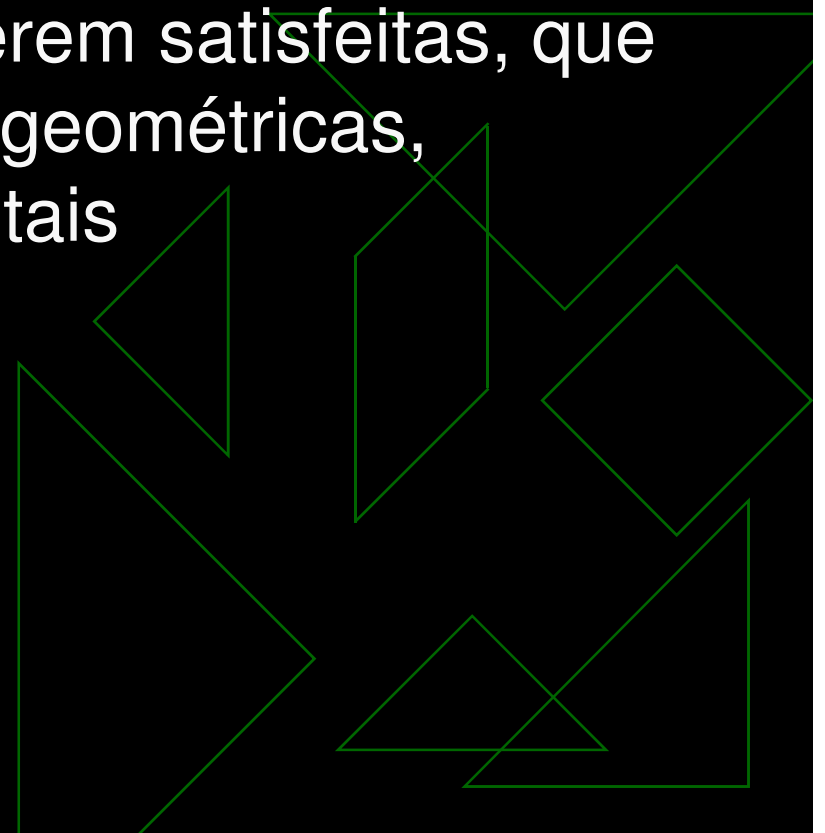
# Corpos Articulados

- ◆ Tipos de vínculos:
  - Vínculos rotacionais
    - ◆ Rotação simples (1 DOF r)
    - ◆ Rotação dupla (2 DOF r)
    - ◆ Livre (3 DOF r)
    - ◆ Sem rotação (0 DOF r)



# Corpos articulados

- ◆ Método baseado em restrição
  - Conjunto de regras a serem satisfeitas, que estabelecem restrições geométricas, físicas ou comportamentais



# Corpos articulados: Física

## ◆ Cinemática

### ■ Cinemática direta:

- ◆ Tem velocidade inicial e calcula variação de posição
- ◆ Dificuldade de controle

### ■ Cinemática inversa:

- ◆ Tem posição final e calcula velocidade inicial
- ◆ Melhor controle, mas pode possuir infinitas soluções

# Corpos articulados: Física

## ◆ Dinâmica

### ■ Dinâmica direta:

- ◆ Especifica força e calcula torque, acelerações angulares e lineares, velocidades e deslocamentos

### ■ Dinâmica inversa:

- ◆ Tem posição final e calcula força inicial

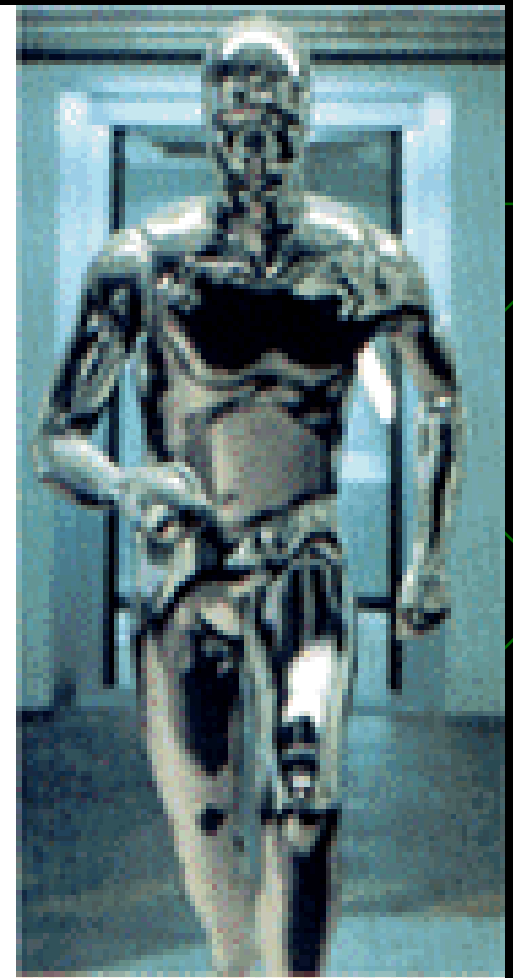
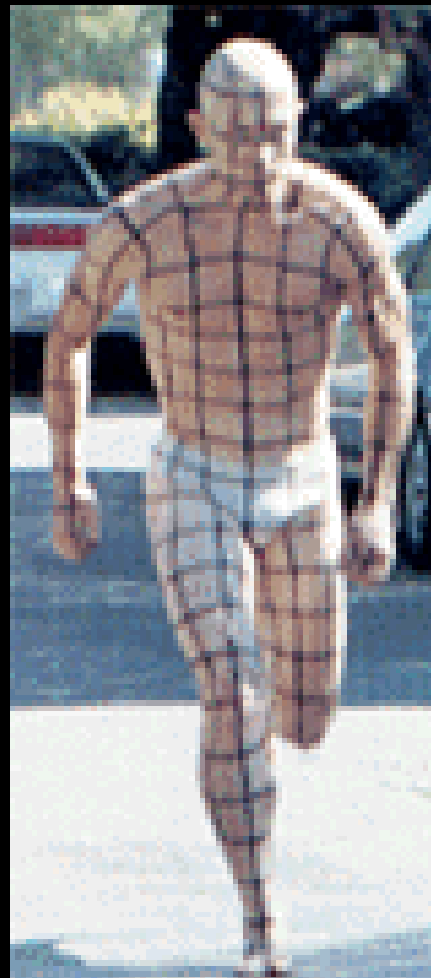
### ■ Princípio das reações vinculares:

- ◆ Para construir o vínculo, anulando os movimentos que não se deseja
  - Ex: If  $F_x \neq 0$  &&  $DOF_x = 0$ 
    - ◆ Then  $F_x = 0$

# PDA

- ◆ Rotoscopia

*Terminator II (1991)*



# PDA

- ◆ Motion Capture



# PDA in faces

- ◆ Performance-driven
  - Captura de pessoas reais
    - ◆ MOCAP
    - ◆ Visão Computacional
      - Com ou sem marcadores
      - Uma ou mais câmeras

