

# Animação por Computador - OAC 0001

Rogério Eduardo da Silva - *rogerio.silva@udesc.br*

Universidade do Estado de Santa Catarina  
Departamento de Ciência da Computação

31 de julho de 2017

# Apresentação da Disciplina

- Conteúdo Programático
- Metodologia
- Avaliações
- Bibliografia Sugerida

## Apresentação da Disciplina

### Conceitos Introdutórios

EPI #1 - Modelagem de Cenários

### Keyframing

### Modelos Hierárquicos

Personagens Articulados

DKA vs IKA

EPI #2 - Pose-to-Pose Animation

### Atores Digitais

Postura Corporal

Expressão Facial

Sincronização Labial

EPI #3 - Atores Digitais

### Modelos baseados na Física

Corpos Rígidos

Corpos Flexíveis

EPI #4 - Física

### Animação de Grupos

Sistemas de Partículas

Controle de Manobras

Fluídios

Partículas Estáticas

EPI #5 - Animação de Grupos

### Projeto Final de Animação

## Método de Ensino

---

- Aulas expositivas em sala e em laboratório
- Listas de exercícios teóricos e práticos
- Atendimento presencial (sala do professor) e/ou através da lista de emails da disciplina

[bcc-ani@googlegroups.com](mailto:bcc-ani@googlegroups.com)

# Avaliações

---

- 5 exercícios práticos individuais *EPI* (15% da média semestral cada um) :
  - Modelagem:** construção de cenários;
  - Poses:** *keyframing*, modelos hierárquicos, DKA/IKA;
  - Atores Virtuais:** *non-linear animation*, expressão corporal e facial;
  - Física:** corpos rígidos, corpos flexíveis e tecidos;
  - Grupos:** sistemas de partículas, controle de manobras;
- Projeto Final de Animação (25% da média semestral)
- Exame Final (caso média semestral < 7.0)  
Data prevista: **Última semana do semestre**

# Bibliografia Básica Sugerida (1)

---



Brito, A. (2011).

*Blender 3D - Jogos e Animações Interativas.*

Editora Novatec.



Kundert-Gibbs, J. and Kundert-Gibbs, K. (2009).

*Action! Acting Lessons for CG Animators.*

Wiley Publishing, Inc., USA.



Mullen, T. (2007).

*Introducing Character Animation with Blender.*

Sybex.



Osipa, J. (2003).

*Stop Staring: Facial Modeling and Animation Done Right.*

Sybex, 1 edition.

## Bibliografia Básica Sugerida (2)

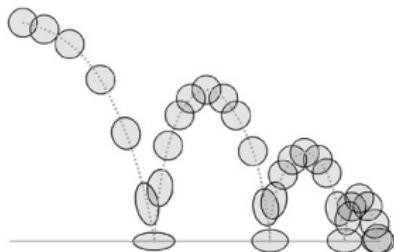
---

-  Parent, R. (2002).  
*Computer Animation - Algorithms and Techniques.*  
Morgan Kaufmann.
-  Reinolds, C. W. (1999).  
Steering behaviors for autonomous characters.  
In *Game Developers Conference*, pages 763–782, San Jose, California.
-  Roberts, S. (2007).  
*Character Animation: 2D Skills for Better 3D.*  
Focal Press, second edition.

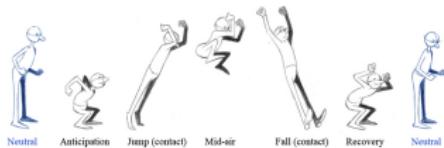
# Conceitos Introdutórios

- Apresentação de exemplos diversos (vídeos)
- Definições fundamentais
- 12 princípios da animação
- Ferramenta Blender 3D

# Animações Computacionais - Fundamentos

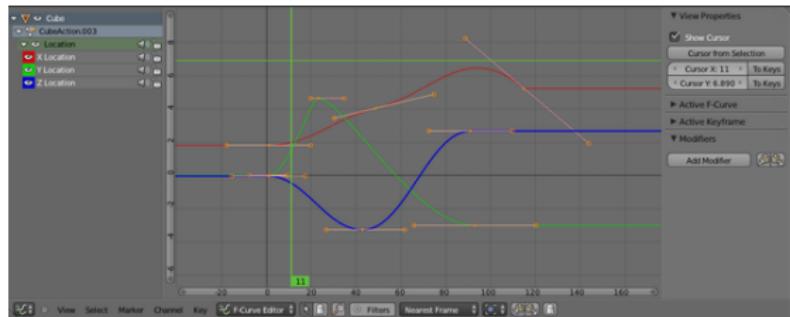
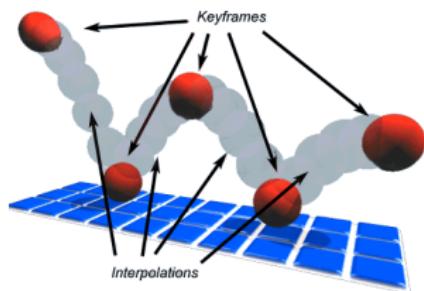


VISIT [WWW.IDLEWORM.COM/HOW/INDEX.SHTML](http://WWW.IDLEWORM.COM/HOW/INDEX.SHTML) FOR ANIMATION TUTORIALS

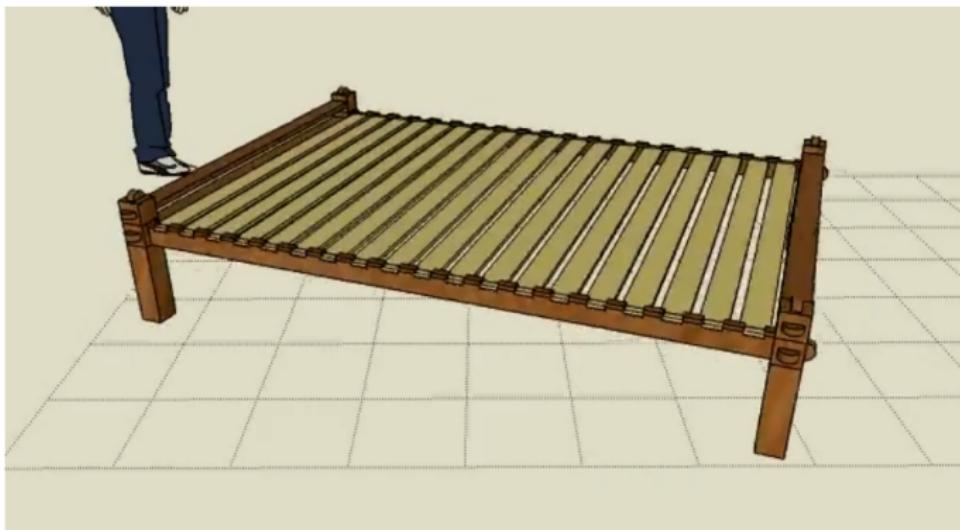


Secondary actions

# Animações Computacionais - Quadros-chaves (I)

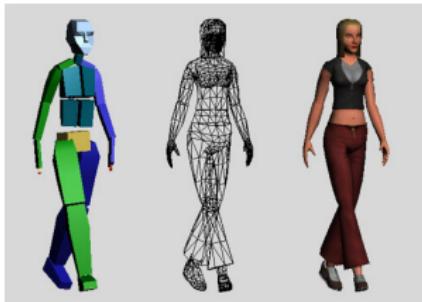


## Animações Computacionais - Quadros-chaves (II)

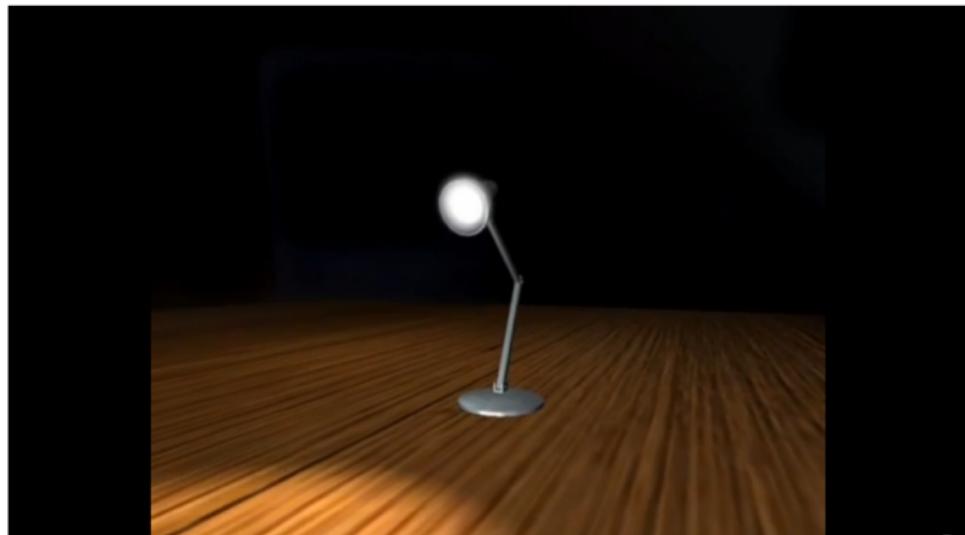


[http://www.youtube.com/watch?v=2ou8\\_SRnLjo](http://www.youtube.com/watch?v=2ou8_SRnLjo)  
( Duração: 1'20" )

# Animações Computacionais - Modelos Hierárquicos (I)



# Animações Computacionais - Modelos Hierárquicos (II)



<http://www.youtube.com/watch?v=aiZVBs1x5j0>  
( Duração: 8" )

# Animações Computacionais - Atores Digitais (I)



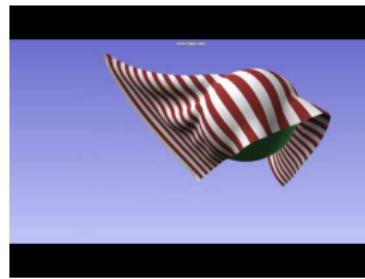
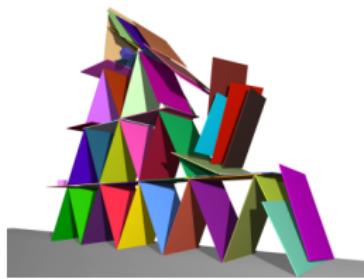
# Animações Computacionais - Atores Digitais (II)



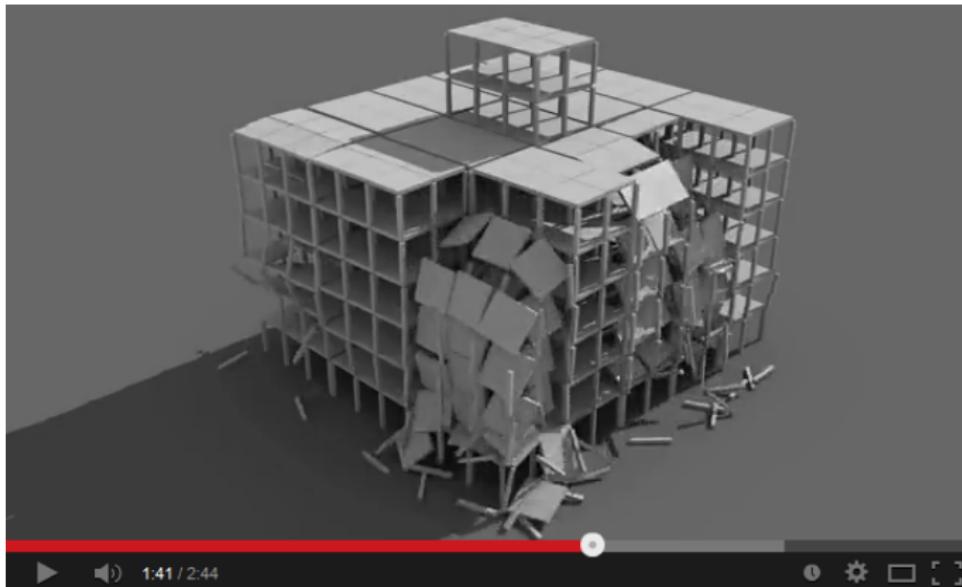
<http://www.youtube.com/watch?v=g19H3qJm-hw>  
( Duração: 1'31" )

# Animações Computacionais - Física (I)

---



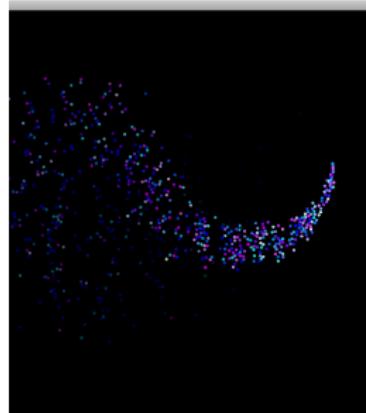
# Animações Computacionais - Física (II)



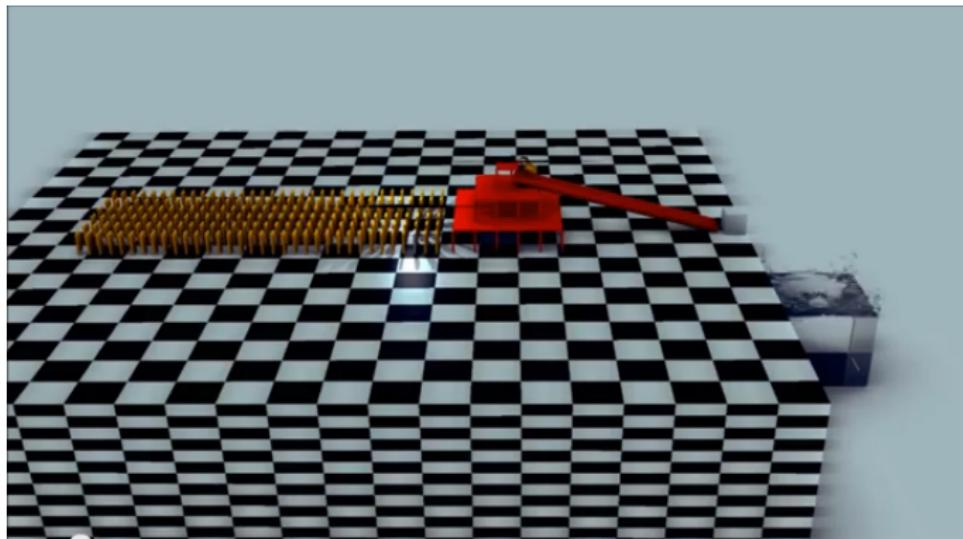
[https://www.youtube.com/watch?v=c-DQoU\\_8H4g](https://www.youtube.com/watch?v=c-DQoU_8H4g)  
( Duração: 2'44" )

# Animações Computacionais - Partículas (I)

---



## Animações Computacionais - Partículas (II)



<https://www.youtube.com/watch?v=l-gMnB3XsDM>  
( Duração: 7'33" )

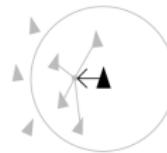
# Animações Computacionais - Bandos (I)



Separation:  
Steer to avoid crowding  
local flockmates

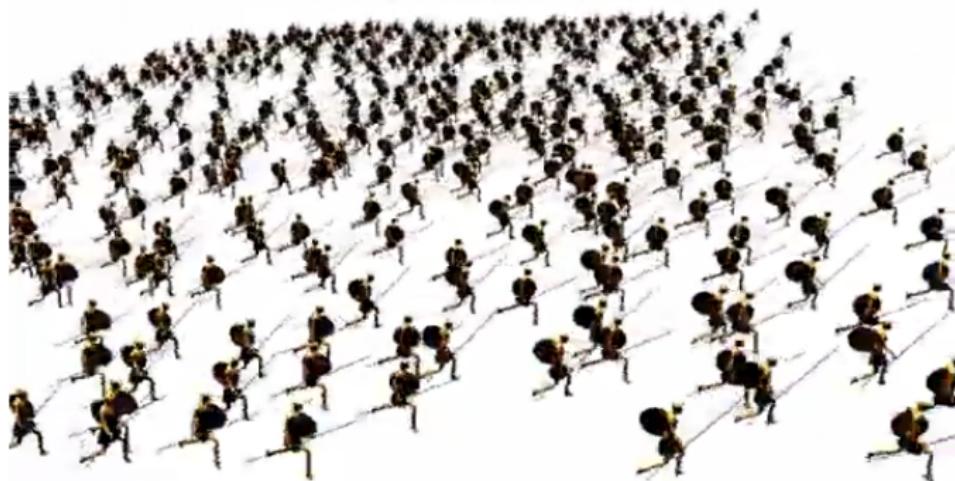


Alignment:  
Steer toward the average  
heading of local flockmates



Cohesion:  
Steer to move toward the average  
position of local flockmates

## Animações Computacionais - Bandos (II)



<https://www.youtube.com/watch?v=LDU8NcNESEg>  
( Duração: 20" )

# Definições Fundamentais

---

- O que é **Animação?**

# Definições Fundamentais

---

- O que é **Animação?**

- Do latim, a palavra *animare* significa “dar vida”, trazer à vida.

# Definições Fundamentais

---

- O que é **Animação**?

- Do latim, a palavra *animare* significa “dar vida”, trazer à vida.
  - É importante notar que animar **não necessariamente** é a mesma coisa que “produzir movimento”.

# Definições Fundamentais

---

## ■ O que é **Animação**?

- Do latim, a palavra *animare* significa “dar vida”, trazer à vida.
- É importante notar que animar **não necessariamente** é a mesma coisa que “produzir movimento”.
- Animar é alterar certos parâmetros do ator em função do tempo

$$Attr_{t+\Delta t} = func(Attr_t, t + \Delta t)$$

# Definições Fundamentais

---

## ■ O que é **Animação**?

- Do latim, a palavra *animare* significa “dar vida”, trazer à vida.
- É importante notar que animar **não necessariamente** é a mesma coisa que “produzir movimento”.
- Animar é alterar certos parâmetros do ator em função do tempo

$$Attr_{t+\Delta t} = func(Attr_t, t + \Delta t)$$

## ■ Como se representa computacionalmente uma animação?

# Definições Fundamentais

---

## ■ O que é **Animação**?

- Do latim, a palavra *animare* significa “dar vida”, trazer à vida.
- É importante notar que animar **não necessariamente** é a mesma coisa que “produzir movimento”.
- Animar é alterar certos parâmetros do ator em função do tempo

$$Attr_{t+\Delta t} = func(Attr_t, t + \Delta t)$$

## ■ Como se representa computacionalmente uma animação?

- Através de uma sequência de imagens (*frames*) que são reproduzidas, uma após a outra, a uma frequência **mais rápida** do que a capacidade do olho humano de perceber mudanças no ambiente.

# Definições Fundamentais

---

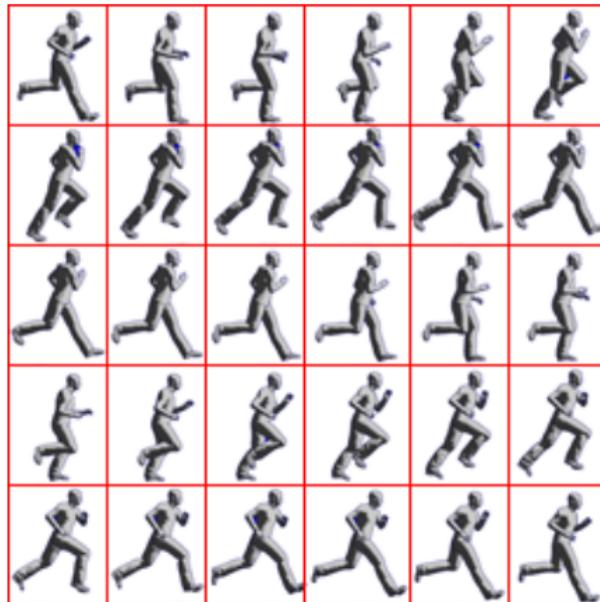


Figura 1: Sequência de frames de um trecho de uma animação

## Definições Fundamentais - Métricas

---

**Resolução Gráfica:** quantidade de *pixels* armazenada em cada quadro da animação.

$$\text{Resolução Gráfica} = N^{\circ} \text{ colunas} \times N^{\circ} \text{ linhas}$$

## Definições Fundamentais - Métricas

---

**Resolução Gráfica:** quantidade de *pixels* armazenada em cada quadro da animação.

$$\text{Resolução Gráfica} = N^{\circ} \text{ colunas} \times N^{\circ} \text{ linhas}$$

Exemplos de resoluções gráficas:

- **TV PAL** =  $720 \times 576$
- **TV NTSC** =  $720 \times 480$
- **HDV 1080p** =  $1440 \times 1080$
- **HDTV 720p** =  $1280 \times 720$
- **HDTV 1080p** =  $1920 \times 1080$

## Definições Fundamentais - Métricas

---

**Relação de Aspecto Gráfico:** é a relação entre a largura e a altura da imagem de um quadro:

$$\text{Aspect Ratio} = \frac{N^{\circ} \text{ colunas}}{N^{\circ} \text{ linhas}}$$

## Definições Fundamentais - Métricas

---

**Relação de Aspecto Gráfico:** é a relação entre a largura e a altura da imagem de um quadro:

$$\text{Aspect Ratio} = \frac{N^{\circ} \text{ colunas}}{N^{\circ} \text{ linhas}}$$

Relações de aspecto clássicas:

- **Standard Screen** = 4:3
- **Widescreen** = 16:9

## Definições Fundamentais - Métricas

---

**Frame Rate:** frequência de reprodução de quadros por unidade de tempo, usualmente medida em *frames por segundo* - fps.

## Definições Fundamentais - Métricas

---

**Frame Rate:** frequência de reprodução de quadros por unidade de tempo, usualmente medida em *frames por segundo* - fps.

- Durante muito tempo, o padrão de transmissão da televisão analógica foi de **24 fps**

## Definições Fundamentais - Métricas

---

**Frame Rate:** frequência de reprodução de quadros por unidade de tempo, usualmente medida em *frames por segundo* - fps.

- Durante muito tempo, o padrão de transmissão da televisão analógica foi de **24 fps**
- Com o advento da TV digital o padrão se tornou **30 fps**

## Definições Fundamentais - Métricas

---

**Frame Rate:** frequência de reprodução de quadros por unidade de tempo, usualmente medida em *frames por segundo* - fps.

- Durante muito tempo, o padrão de transmissão da televisão analógica foi de **24 fps**
- Com o advento da TV digital o padrão se tornou **30 fps**
- Existem consoles de video-game que utilizam padrões bem mais altos (**> 60 fps**)

## Definições Fundamentais - Métricas

---

**Frame Rate:** frequência de reprodução de quadros por unidade de tempo, usualmente medida em *frames por segundo* - fps.

- Durante muito tempo, o padrão de transmissão da televisão analógica foi de **24 fps**
- Com o advento da TV digital o padrão se tornou **30 fps**
- Existem consoles de video-game que utilizam padrões bem mais altos (**> 60 fps**)
  
- Porquê é importante conhecermos a frequência de reprodução de quadros de uma animação?

## Definições Fundamentais - Métricas

---

**Frame Rate:** frequência de reprodução de quadros por unidade de tempo, usualmente medida em *frames por segundo* - fps.

- Durante muito tempo, o padrão de transmissão da televisão analógica foi de **24 fps**
- Com o advento da TV digital o padrão se tornou **30 fps**
- Existem consoles de video-game que utilizam padrões bem mais altos (**> 60 fps**)
  
- Porquê é importante conhecermos a frequência de reprodução de quadros de uma animação?
  - O sistema visual humano também possui uma frequência de captação de imagens do mundo.

## Definições Fundamentais - Métricas

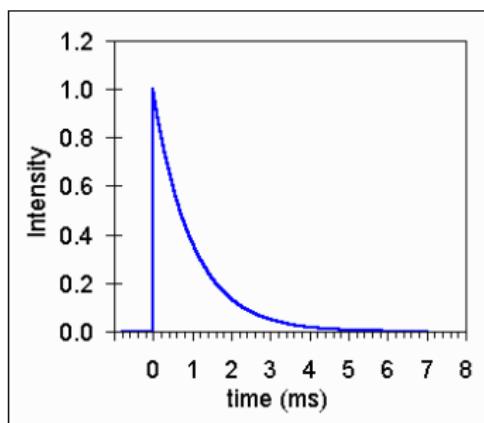
---

**Frame Rate:** frequência de reprodução de quadros por unidade de tempo, usualmente medida em *frames por segundo* - fps.

- Durante muito tempo, o padrão de transmissão da televisão analógica foi de **24 fps**
- Com o advento da TV digital o padrão se tornou **30 fps**
- Existem consoles de video-game que utilizam padrões bem mais altos (**> 60 fps**)
  
- Porquê é importante conhecermos a frequência de reprodução de quadros de uma animação?
  - O sistema visual humano também possui uma frequência de captação de imagens do mundo.
  - Se a frequência de reprodução for baixa demais, a audiência se tornaria capaz de visualizar a mudança de quadros e uma exibição de filme apareceria 'cintilante' aos nossos olhos (**efeito flickering**).

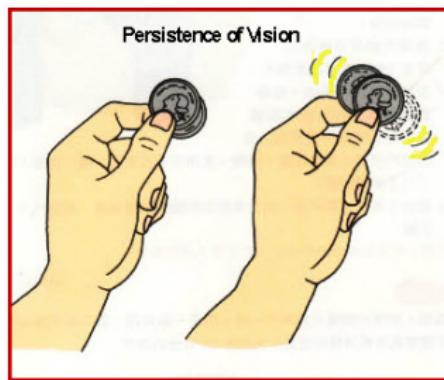
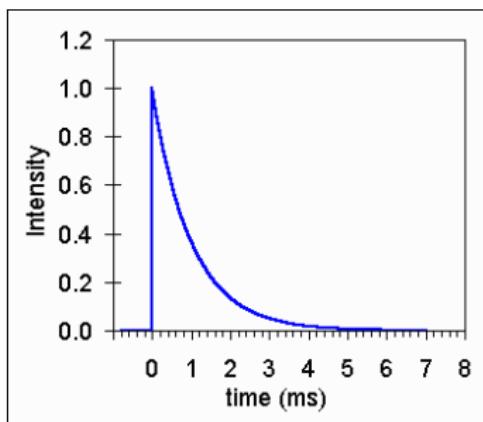
## Definições Fundamentais - Persistência da Visão

- Fenômeno fisiológico no qual uma imagem “persiste” na retina por uma fração de  $\frac{1}{25}$  de segundo, mesmo após o estímulo visual ter cessado.



## Definições Fundamentais - Persistência da Visão

- Fenômeno fisiológico no qual uma imagem “persiste” na retina por uma fração de  $\frac{1}{25}$  de segundo, mesmo após o estímulo visual ter cessado.



## 12 Princípios da Animação

---

- Na década de 1930, Walt Disney propõe uma série de regras a serem aplicadas na animação a fim de produzir resultados mais realistas.

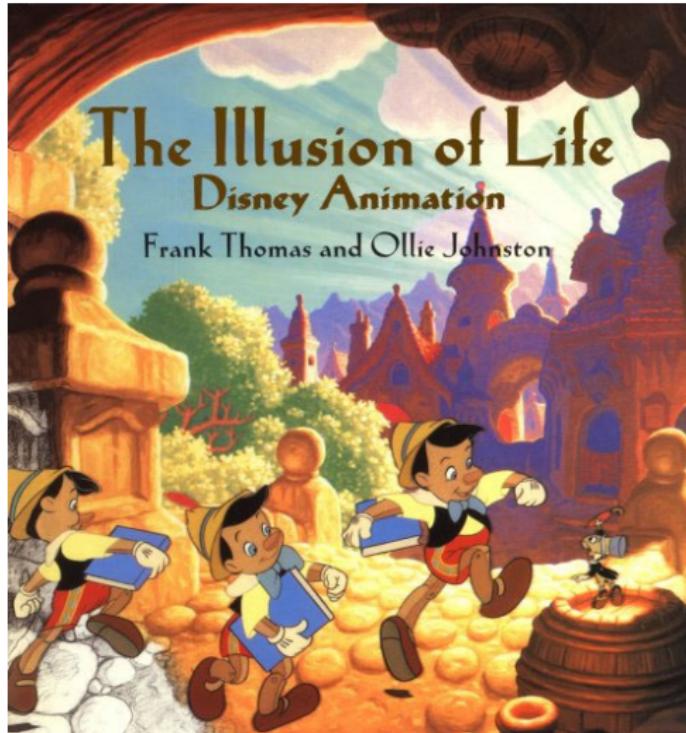
# 12 Princípios da Animação

---

- Na década de 1930, Walt Disney propõe uma série de regras a serem aplicadas na animação a fim de produzir resultados mais realistas.
- Walt Disney treinou um grupo de nove animadores que se tornaram seus principais assistentes e ficaram conhecidos como "*Disney's Nine Old Men*":
  1. **Les Clark:** Mickey Mouse e A Dama e o Vagabundo
  2. **Marc Davis:** Branca de Neve, Bambi, A Bela Adormecida, 101 Dálmatas
  3. **Ollie Johnston:** Branca de Neve e Peter Pan
  4. **Milt Kahl:** Branca de Neve, Mogli - O Menino-Lobo, Aristogatas e Robin Hood
  5. **Ward Kimball:** Pinoquio, Cinderella, Alice no País das Maravilhas
  6. **Eric Larson:** A Dama e o Vagabundo, Mogli - O Menino-Lobo e Peter Pan
  7. **John Lounsbery:** Fantasia, Peter Pan, A Dama e o Vagabundo, A Bela Adormecida, Ursinho Puff e o Tigrão
  8. **Wolfgang Reitherman:** Pinoquio, Peter Pan e A Dama e o Vagabundo
  9. **Frank Thomas:** Cinderella, ALice no País das Maravilhas e Peter Pan

# 12 Princípios da Animação

---



# 12 Princípios da Animação

---

1. Timing & Spacing
2. Squash & Stretch
3. Staging
4. Anticipation
5. Straight ahead vs Pose-to-Pose
6. Follow through & Overlapping action
7. Slow in & Slow out
8. Arcs
9. Secondary Action
10. Exaggeration
11. Solid Drawing
12. Appeal

## 12 Princípios da Animação - Timing & Spacing

---

**Timing:** refere-se a qualquer coisa que acontece em relação ao tempo.

Este princípio está diretamente relacionado à velocidade da ação no filme

## 12 Princípios da Animação - Timing & Spacing

---

**Timing:** refere-se a qualquer coisa que acontece em relação ao tempo.

Este princípio está diretamente relacionado à velocidade da ação no filme

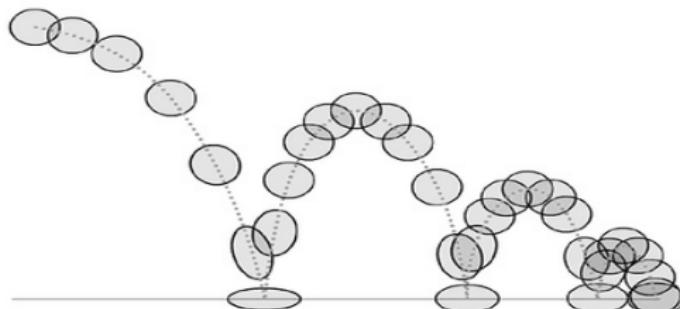
**Spacing:** refere-se à quantidade de variação que determinado atributo 'percorre' por unidade de tempo (ou seja, está relacionado com timing)

# 12 Princípios da Animação - Timing & Spacing

**Timing:** refere-se a qualquer coisa que acontece em relação ao tempo.

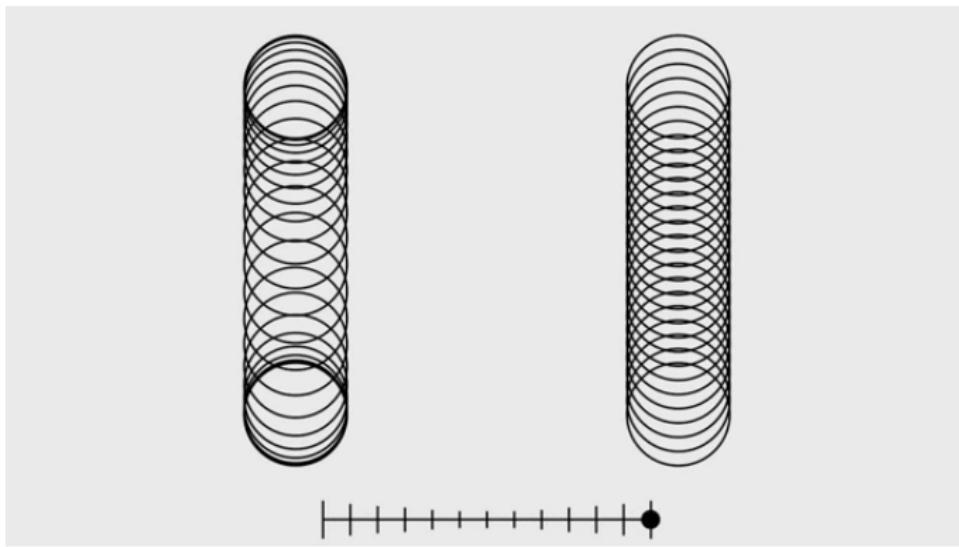
Este princípio está diretamente relacionado à velocidade da ação no filme

**Spacing:** refere-se à quantidade de variação que determinado atributo 'percorre' por unidade de tempo (ou seja, está relacionado com timing)



VISIT [WWW.IDLEWORM.COM/HOW/INDEX.SHTML](http://WWW.IDLEWORM.COM/HOW/INDEX.SHTML) FOR ANIMATION TUTORIALS

# 12 Princípios da Animação - Timing & Spacing



Filme: Animation Principle 1 - Spacing And Timing.mp4  
( Duração: 1'54" )

## 12 Princípios da Animação - Squash & Stretch

---

**Squash:** refere-se à deformação (“esmagar”) causada aquando da alteração abrupta de movimento em objetos maleáveis (p.ex.: impacto)

## 12 Princípios da Animação - Squash & Stretch

---

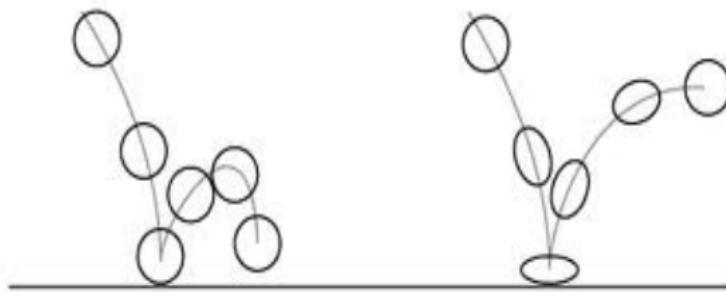
**Squash:** refere-se à deformação (“esmagar”) causada aquando da alteração abrupta de movimento em objetos maleáveis (p.ex.: impacto)

**Stretch:** refere-se à deformação inversa (“esticar”) causada devido à retomada da velocidade em objetos maleáveis (p.ex.: aceleração)

## 12 Princípios da Animação - Squash & Stretch

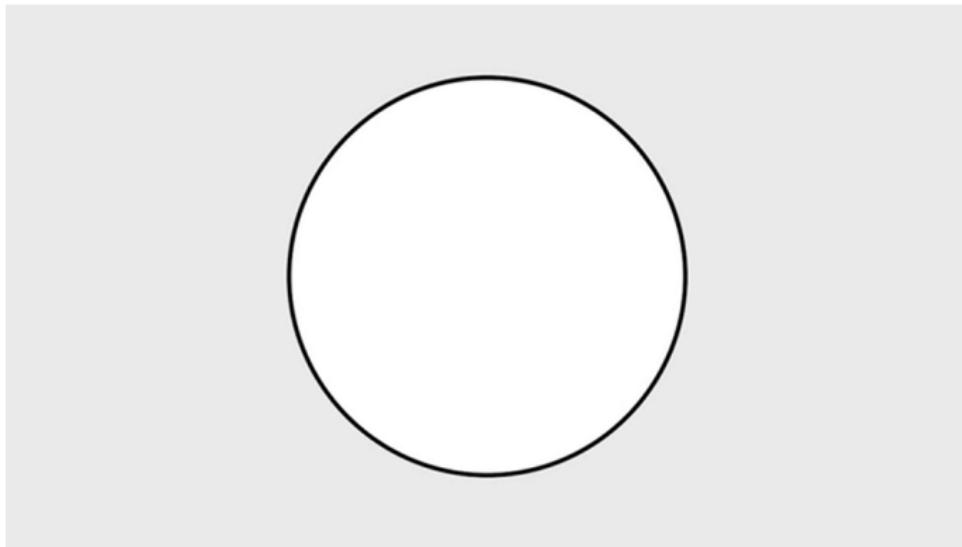
**Squash:** refere-se à deformação (“esmagar”) causada aquando da alteração abrupta de movimento em objetos maleáveis (p.ex.: impacto)

**Stretch:** refere-se à deformação inversa (“esticar”) causada devido à retomada da velocidade em objetos maleáveis (p.ex.: aceleração)



# 12 Princípios da Animação - Squash & Stretch

---



Filme: Animation Principle 2 - Squash And Stretch.mp4  
( Duração: 4'16" )

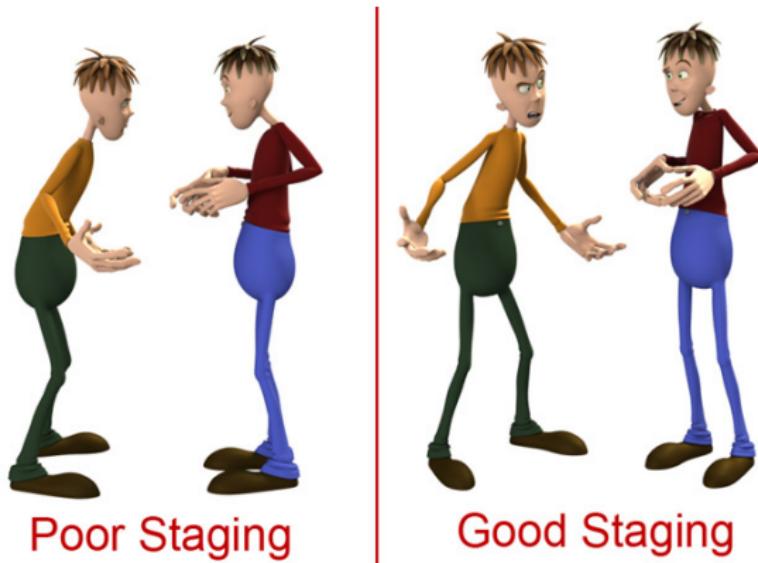
## 12 Princípios da Animação - Staging

---

**Staging:** refere-se à encenação de uma estória através da animação de um personagem. Implica em se determinar formas apropriadas de se representar emoção, personalidade, intenção, etc.

# 12 Princípios da Animação - Staging

**Staging:** refere-se à encenação de uma estória através da animação de um personagem. Implica em se determinar formas apropriadas de se representar emoção, personalidade, intenção, etc.



# 12 Princípios da Animação - Staging

---



Filme: Animation Principle 3 - Staging.mp4  
( Duração: 49" )

# 12 Princípios da Animação - Straight-Ahead vs Pose-to-Pose

---

**Straight-ahead:** refere-se ao processo de se animar uma sequência à partir do primeiro quadro, desenhando linearmente a sequência até que o último seja alcançado

# 12 Princípios da Animação - Straight-Ahead vs Pose-to-Pose

---

**Straight-ahead:** refere-se ao processo de se animar uma sequência à partir do primeiro quadro, desenhando linearmente a sequência até que o último seja alcançado

**Pose-to-Pose:** refere-se ao processo de se animar à partir de um sequência de quadros pré-planejados, preenchendo-se os quadros intermediários entre pares desses “quadros-chave”

# 12 Princípios da Animação - Straight-Ahead vs Pose-to-Pose

---

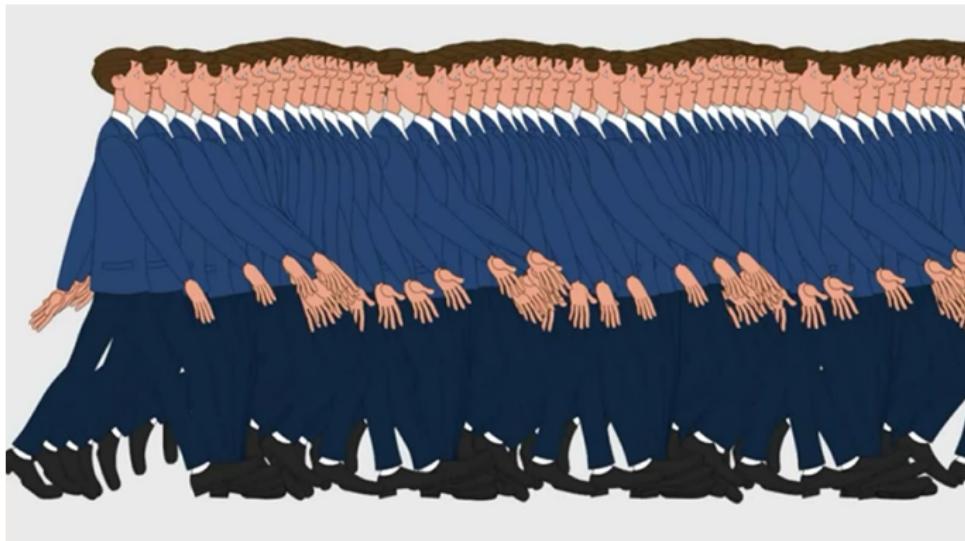
**Straight-ahead:** refere-se ao processo de se animar uma sequência à partir do primeiro quadro, desenhando linearmente a sequência até que o último seja alcançado

**Pose-to-Pose:** refere-se ao processo de se animar à partir de um sequência de quadros pré-planejados, preenchendo-se os quadros intermediários entre pares desses “quadros-chave”



# 12 Princípios da Animação - Straight-Ahead vs Pose-to-Pose

---



Filme: Animation Principle 4 - Straight Ahead And Pose To Pose Anim.mp4  
( Duração: 53" )

## 12 Princípios da Animação - Follow Through

---

**Follow Through:** refere-se ao processo de se animar sub-partes de um personagem que permanecem em movimento, mesmo após o ator ter parado de se mover (p.ex.: cabelo, braços, roupas, etc.)

## 12 Princípios da Animação - Follow Through

---

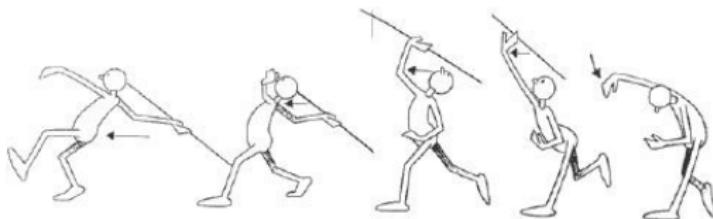
**Follow Through:** refere-se ao processo de se animar sub-partes de um personagem que permanecem em movimento, mesmo após o ator ter parado de se mover (p.ex.: cabelo, braços, roupas, etc.)

**Overlapping Action:** refere-se ao fato de que um personagem move diferentes partes do corpo simultaneamente (p.ex.: balançar dos braços ao caminhar)

## 12 Princípios da Animação - Follow Through

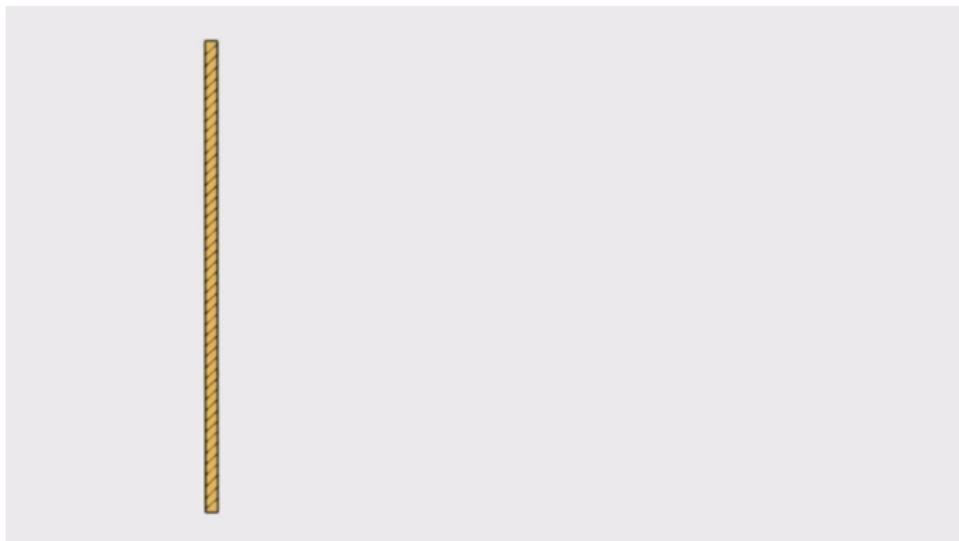
**Follow Through:** refere-se ao processo de se animar sub-partes de um personagem que permanecem em movimento, mesmo após o ator ter parado de se mover (p.ex.: cabelo, braços, roupas, etc.)

**Overlapping Action:** refere-se ao fato de que um personagem move diferentes partes do corpo simultaneamente (p.ex.: balançar dos braços ao caminhar)



# 12 Princípios da Animação - Follow Through

---



Filme: Animation Principle 5 - Follow Through.mp4  
( Duração: 37" )

## 12 Princípios da Animação - Ease In/Out

---

**Ease In:** refere-se ao fato que nenhum movimento inicia abruptamente, sendo necessário acelerar o personagem

## 12 Princípios da Animação - Ease In/Out

---

**Ease In:** refere-se ao fato que nenhum movimento inicia abruptamente, sendo necessário acelerar o personagem

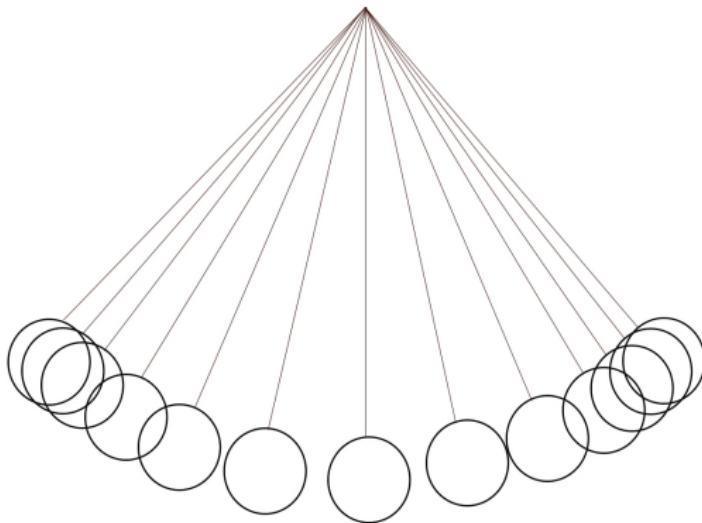
**Ease Out:** refere-se ao fato que nenhum movimento termina abruptamente, sendo necessário desacelerar o personagem

## 12 Princípios da Animação - Ease In/Out

---

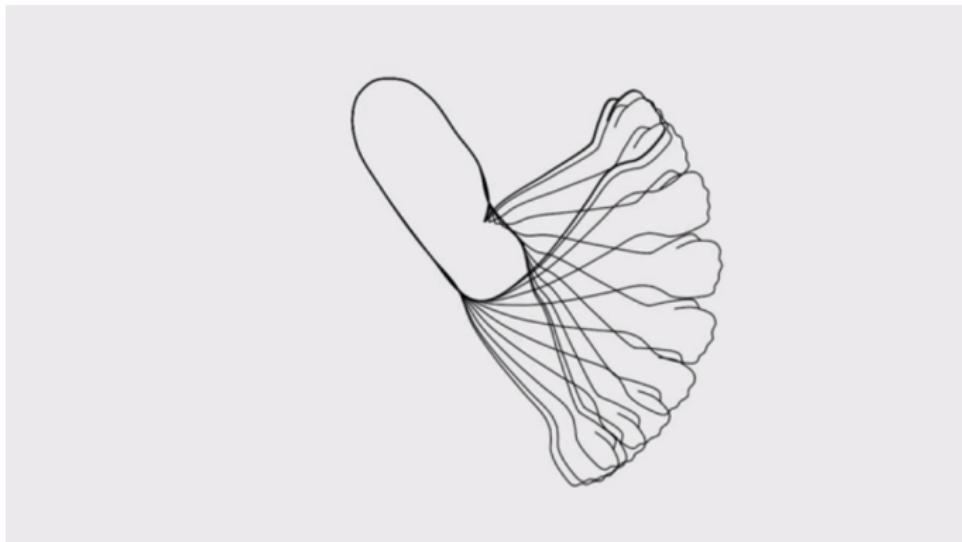
**Ease In:** refere-se ao fato que nenhum movimento inicia abruptamente, sendo necessário acelerar o personagem

**Ease Out:** refere-se ao fato que nenhum movimento termina abruptamente, sendo necessário desacelerar o personagem



# 12 Princípios da Animação - Ease In/Out

---



Filme: Animation Principle 6 - Ease In Ease Out.mp4  
( Duração: 27" )

## 12 Princípios da Animação - Arcos

---

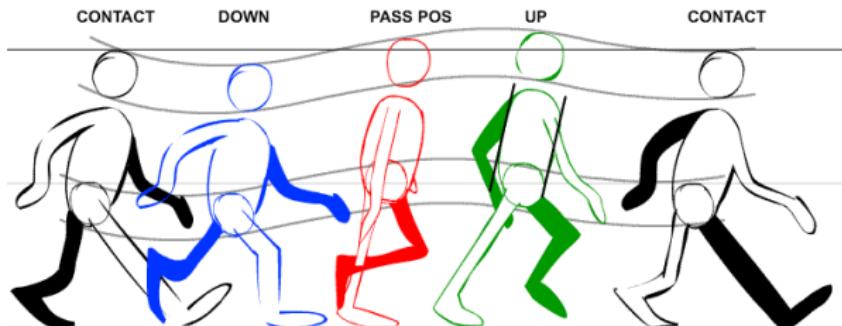
**Arcos:** refere-se ao fato que a maioria dos movimentos (especialmente os orgânicos) define um caminho curvo ao deslocar-se

# 12 Princípios da Animação - Arcos

**Arcos:** refere-se ao fato que a maioria dos movimentos (especialmente os orgânicos) define um caminho curvo ao deslocar-se

**Walks & Weight 7** (Exaggerated, more clearer)

"Watch those arcs"



## 12 Princípios da Animação - Arcos

---



Filme: Animation Principle 7 - Arcs.mp4  
( Duração: 37" )

## 12 Princípios da Animação - Ação Secundária

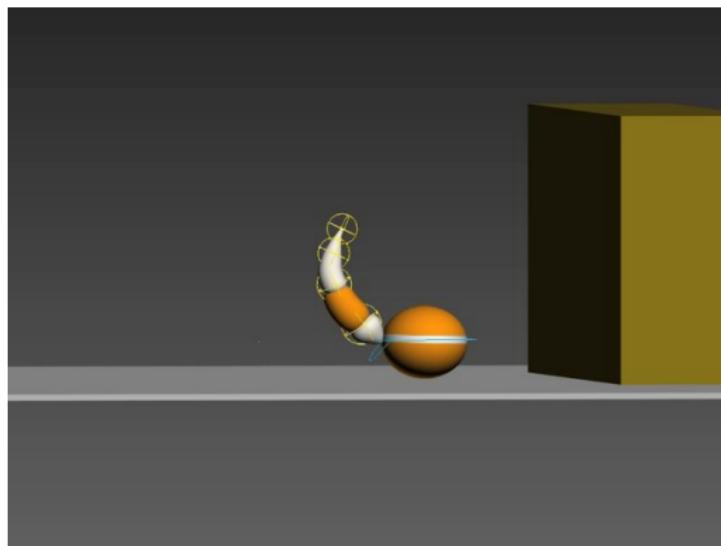
---

**Ação Secundária:** refere-se ao fato que personagens realizam múltiplas ações simultaneamente. Tais ações acrescentam uma noção maior de “vida” à animação

## 12 Princípios da Animação - Ação Secundária

---

**Ação Secundária:** refere-se ao fato que personagens realizam múltiplas ações simultaneamente. Tais ações acrescentam uma noção maior de “vida” à animação



## 12 Princípios da Animação - Ação Secundária

---



Filme: Animation Principle 8 - Secondary Motion.mp4  
( Duração: 31" )

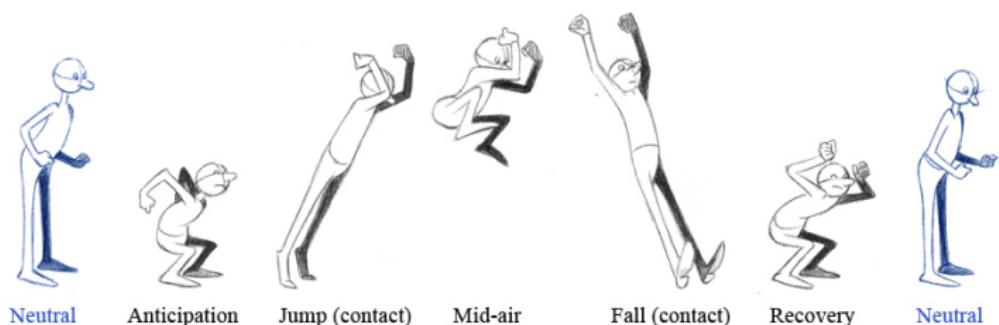
## 12 Princípios da Animação - Antecipação

---

**Antecipação:** refere-se ao fato que para personagens realizarem determinadas ações uma certa “preparação” se faz necessária (p.ex.: pegar impulso antes de iniciar um salto)

# 12 Princípios da Animação - Antecipação

**Antecipação:** refere-se ao fato que para personagens realizarem determinadas ações uma certa “preparação” se faz necessária (p.ex.: pegar impulso antes de iniciar um salto)



## 12 Princípios da Animação - Antecipação

---



Filme: Animation Principle 9 - Anticipation.mp4  
( Duração: 46" )

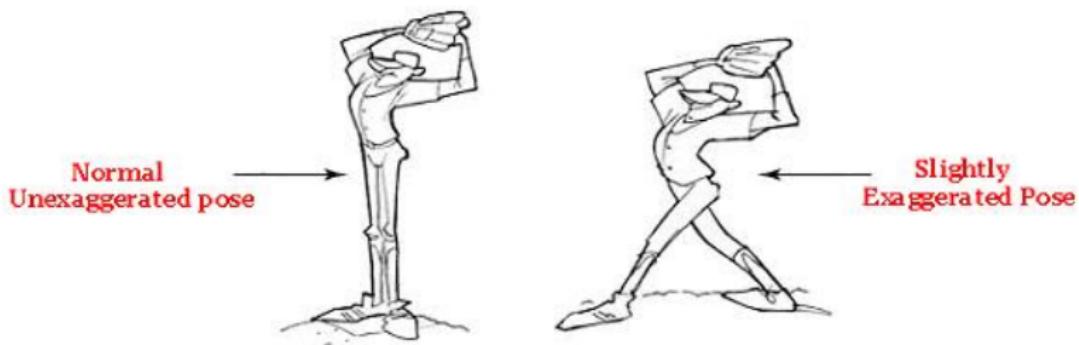
## 12 Princípios da Animação - Exagero

---

**Exagero:** refere-se ao fato de que a intenção de um personagem animado é mais facilmente percebida quando os movimentos são exagerados em termos de intensidade e duração

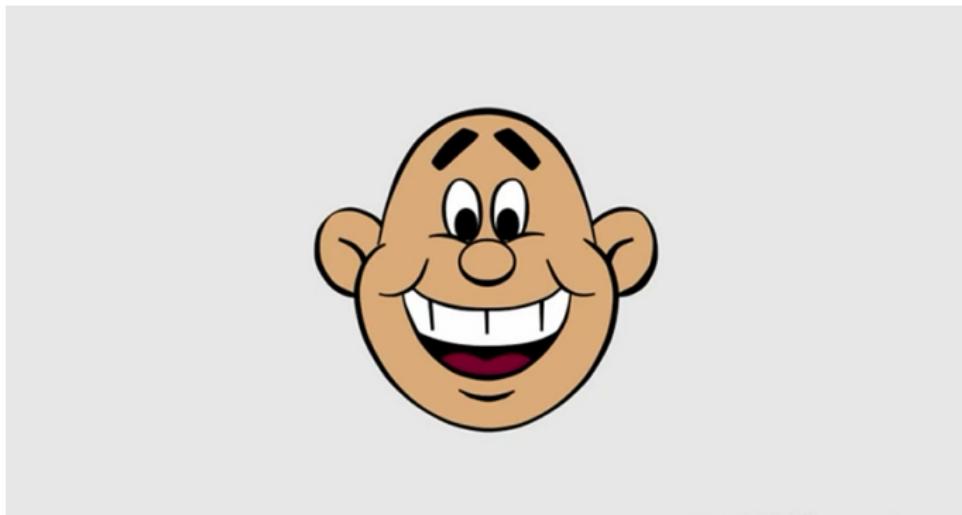
## 12 Princípios da Animação - Exagero

**Exagero:** refere-se ao fato de que a intenção de um personagem animado é mais facilmente percebida quando os movimentos são exagerados em termos de intensidade e duração



## 12 Princípios da Animação - Exagero

---



Filme: Animation Principle 10 - Exaggeration.mp4  
( Duração: 31" )

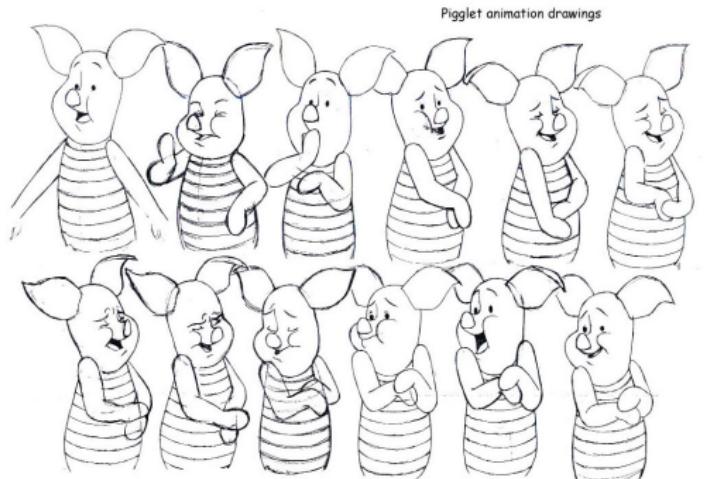
## 12 Princípios da Animação - Solid Drawing

---

**Solid Drawing:** refere-se ao fato de que um personagem deve manter as relações de proporção, cor, textura, etc. durante todos os quadros de uma animação, independente da sua posição aparente em relação ao quadro

# 12 Princípios da Animação - Solid Drawing

**Solid Drawing:** refere-se ao fato de que um personagem deve manter as relações de proporção, cor, textura, etc. durante todos os quadros de uma animação, independente da sua posição aparente em relação ao quadro



# 12 Princípios da Animação - Solid Drawing

---



Filme: Animation Principle 11 - Solid Drawing.mp4  
( Duração: 32" )

## 12 Princípios da Animação - Apelo

---

**Apelo:** refere-se ao fato de que um personagem deve ser capaz de cativar a audiência apresentando uma atuação convincente e atraente. Aspectos a serem considerados: entonação de voz, postura corporal e facial, movimentação e interação (tanto com os objetos do cenário, quanto com outros atores)

## 12 Princípios da Animação - Apelo

---

**Apelo:** refere-se ao fato de que um personagem deve ser capaz de cativar a audiência apresentando uma atuação convincente e atraente. Aspectos a serem considerados: entonação de voz, postura corporal e facial, movimentação e interação (tanto com os objetos do cenário, quanto com outros atores)



## Ferramentas Gráficas - Blender 3D

---

- Suite opensource de desenvolvimento de aplicações gráficas como imagens, animações, jogos digitais, efeitos especiais, e mais...

## Ferramentas Gráficas - Blender 3D

---

- Suite opensource de desenvolvimento de aplicações gráficas como imagens, animações, jogos digitais, efeitos especiais, e mais...
- Atualmente na versão 2.69 (Windows, Linux e Mac)

## Ferramentas Gráficas - Blender 3D

---

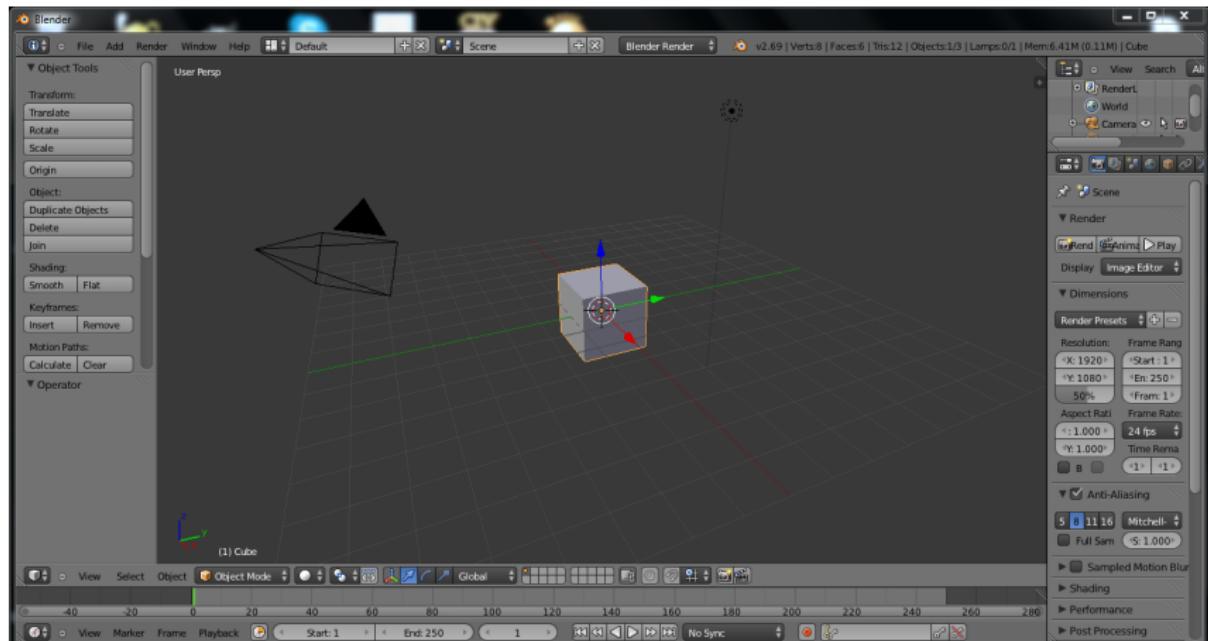
- Suite opensource de desenvolvimento de aplicações gráficas como imagens, animações, jogos digitais, efeitos especiais, e mais...
- Atualmente na versão 2.69 (Windows, Linux e Mac)
- Vasta documentação online (apostilas, livros, video-aulas e cursos)

# Ferramentas Gráficas - Blender 3D

---

- Suite opensource de desenvolvimento de aplicações gráficas como imagens, animações, jogos digitais, efeitos especiais, e mais...
- Atualmente na versão 2.69 (Windows, Linux e Mac)
- Vasta documentação online (apostilas, livros, video-aulas e cursos)
- Características (segundo o desenvolvedor):
  - Rendering fotorealístico
  - Fácil modelagem de objetos
  - Especificação de materiais realísticos
  - Fácil construção de modelos hierárquicos
  - Conjunto de ferramentas para animação
  - Ferramentas para escultura digital
  - Recursos para construção de mapas de textura UV
  - Composição digital
  - Simulação digital
  - Criação de jogos
  - Rastreio de câmera e de objetos no espaço
  - Extensa biblioteca de extensões (*third-party add-ons*)
  - Interface customizável
  - Suporte a múltiplos formatos de arquivo

# Ferramentas Gráficas - Blender 3D



# Ferramentas Gráficas - Blender 3D

---

Tipos de engines gráficas suportadas pelo Blender:

**Blender Render:** engine clássica baseada nas técnicas tradicionais de iluminação direta

# Ferramentas Gráficas - Blender 3D

---

Tipos de engines gráficas suportadas pelo Blender:

**Blender Render:** engine clássica baseada nas técnicas tradicionais de iluminação direta

**Blender Game:** engine para renderização em tempo real de aplicações interativas (jogos)

## Ferramentas Gráficas - Blender 3D

---

Tipos de engines gráficas suportadas pelo Blender:

**Blender Render:** engine clássica baseada nas técnicas tradicionais de iluminação direta

**Blender Game:** engine para renderização em tempo real de aplicações interativas (jogos)

**Blender Cycles:** engine baseada em técnicas de iluminação global cíclica

# Ferramentas Gráficas - Blender 3D

---

Tipos de engines gráficas suportadas pelo Blender:

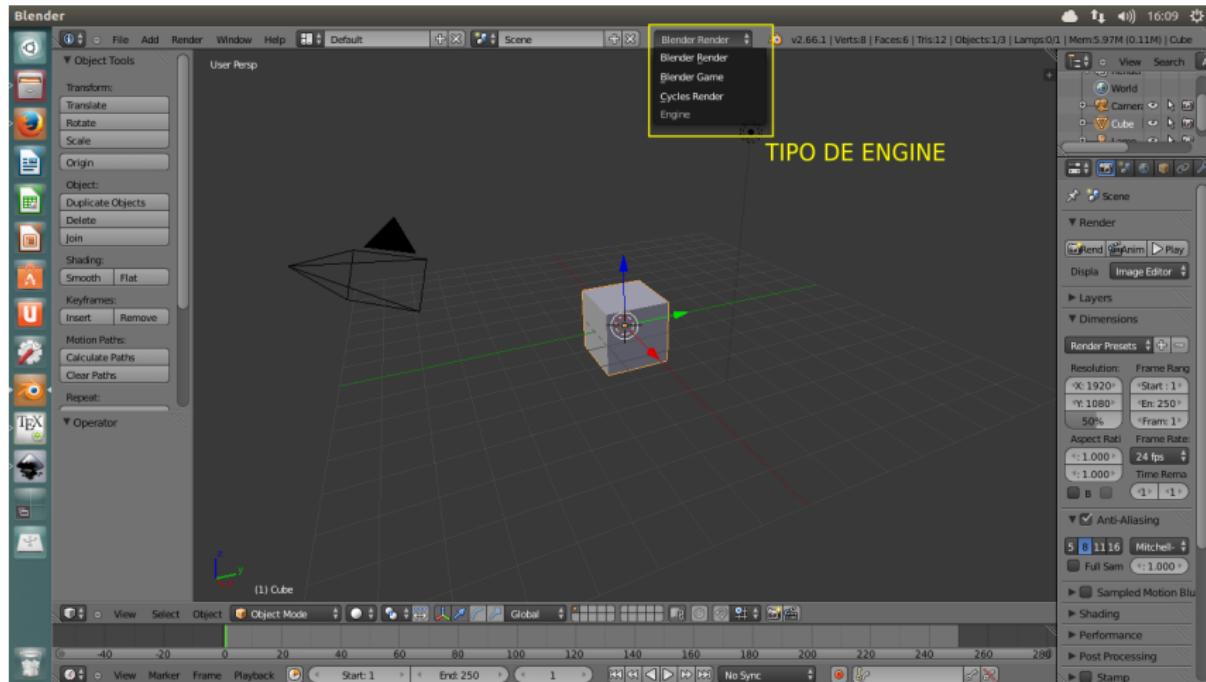
**Blender Render:** engine clássica baseada nas técnicas tradicionais de iluminação direta

**Blender Game:** engine para renderização em tempo real de aplicações interativas (jogos)

**Blender Cycles:** engine baseada em técnicas de iluminação global cíclica

**Blender Eevee:** nova engine para renderização em tempo real = **EM BREVE...**

# Ferramentas Gráficas - Blender 3D



# Ferramentas Gráficas - Blender 3D

---

Contextos gráficos suportadas pelo Blender:

**Video Sequence Editor:** ambiente de construção de vídeos multi-trilhas  
(áudio + vídeo + imagens + etc.)

# Ferramentas Gráficas - Blender 3D

---

Contextos gráficos suportadas pelo Blender:

**Video Sequence Editor:** ambiente de construção de vídeos multi-trilhas  
(áudio + vídeo + imagens + etc.)

**UV/Image Editor:** ambiente para construção de mapeamento de texturas  
UV

# Ferramentas Gráficas - Blender 3D

---

Contextos gráficos suportadas pelo Blender:

**Video Sequence Editor:** ambiente de construção de vídeos multi-trilhas  
(áudio + vídeo + imagens + etc.)

**UV/Image Editor:** ambiente para construção de mapeamento de texturas  
UV

**NLA Editor:** ambiente para construção de animações não-lineares  
(*non-linear animation*)

# Ferramentas Gráficas - Blender 3D

---

Contextos gráficos suportadas pelo Blender:

**Video Sequence Editor:** ambiente de construção de vídeos multi-trilhas  
(áudio + vídeo + imagens + etc.)

**UV/Image Editor:** ambiente para construção de mapeamento de texturas  
UV

**NLA Editor:** ambiente para construção de animações não-lineares  
(*non-linear animation*)

**Dope Sheet:** ambiente para edição de quadros-chave

# Ferramentas Gráficas - Blender 3D

---

Contextos gráficos suportadas pelo Blender:

**Video Sequence Editor:** ambiente de construção de vídeos multi-trilhas  
(áudio + vídeo + imagens + etc.)

**UV/Image Editor:** ambiente para construção de mapeamento de texturas  
UV

**NLA Editor:** ambiente para construção de animações não-lineares  
(*non-linear animation*)

**Dope Sheet:** ambiente para edição de quadros-chave

**Graph Editor:** ambiente para edição do comportamento dos atributos  
animados via linha do tempo

# Ferramentas Gráficas - Blender 3D

---

Contextos gráficos suportadas pelo Blender:

**Video Sequence Editor:** ambiente de construção de vídeos multi-trilhas  
(áudio + vídeo + imagens + etc.)

**UV/Image Editor:** ambiente para construção de mapeamento de texturas  
UV

**NLA Editor:** ambiente para construção de animações não-lineares  
(*non-linear animation*)

**Dope Sheet:** ambiente para edição de quadros-chave

**Graph Editor:** ambiente para edição do comportamento dos atributos  
animados via linha do tempo

**Timeline:** ambiente de apresentação da linha do tempo sendo animada

# Ferramentas Gráficas - Blender 3D

---

Contextos gráficos suportadas pelo Blender:

**Video Sequence Editor:** ambiente de construção de vídeos multi-trilhas  
(áudio + vídeo + imagens + etc.)

**UV/Image Editor:** ambiente para construção de mapeamento de texturas  
UV

**NLA Editor:** ambiente para construção de animações não-lineares  
(*non-linear animation*)

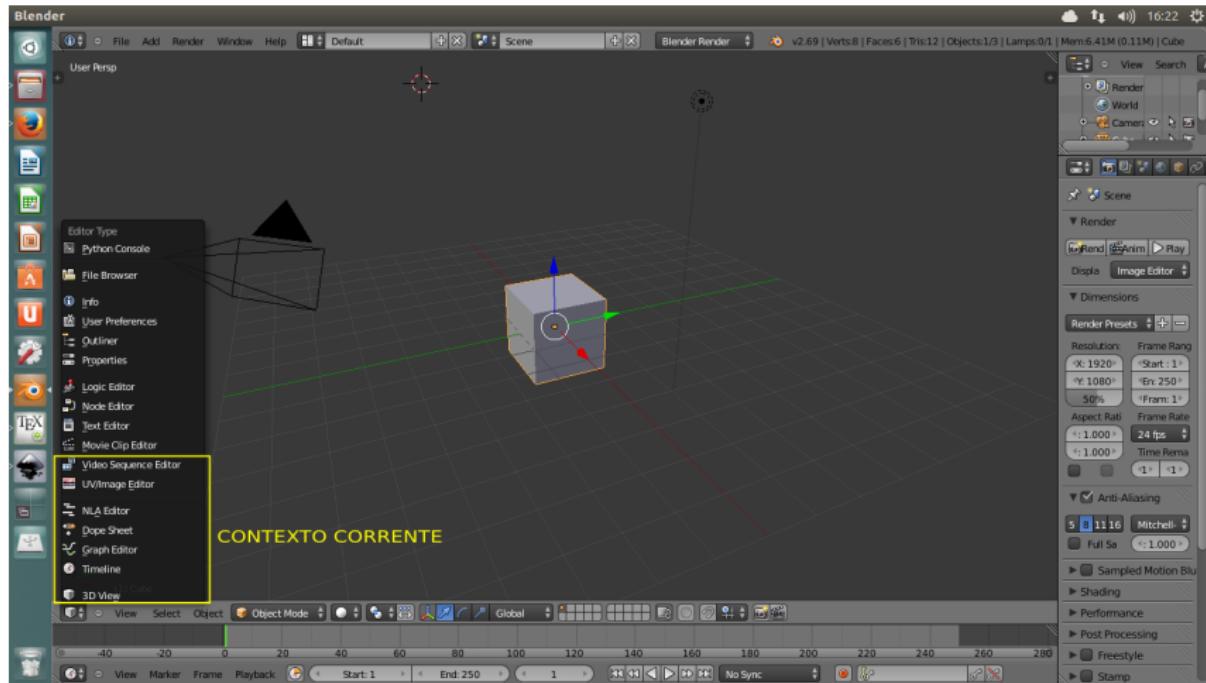
**Dope Sheet:** ambiente para edição de quadros-chave

**Graph Editor:** ambiente para edição do comportamento dos atributos  
animados via linha do tempo

**Timeline:** ambiente de apresentação da linha do tempo sendo animada

**3D View:** ambiente editor da cena sendo animada.

# Ferramentas Gráficas - Blender 3D



CONTEXTO CORRENTE

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

**Scene:** configuração dos parâmetros de edição de cena (p.ex.: unidade de medida)

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

**Scene:** configuração dos parâmetros de edição de cena (p.ex.: unidade de medida)

**World:** configuração do ambiente no qual a cena está inserida

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

**Scene:** configuração dos parâmetros de edição de cena (p.ex.: unidade de medida)

**World:** configuração do ambiente no qual a cena está inserida

**Object:** configuração de parâmetros de edição do objeto atualmente selecionado na cena

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

**Scene:** configuração dos parâmetros de edição de cena (p.ex.: unidade de medida)

**World:** configuração do ambiente no qual a cena está inserida

**Object:** configuração de parâmetros de edição do objeto atualmente selecionado na cena

**Constraints:** regras de restrição (para modelagem e/ou animação) para o objeto selecionado

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

**Scene:** configuração dos parâmetros de edição de cena (p.ex.: unidade de medida)

**World:** configuração do ambiente no qual a cena está inserida

**Object:** configuração de parâmetros de edição do objeto atualmente selecionado na cena

**Constraints:** regras de restrição (para modelagem e/ou animação) para o objeto selecionado

**Modifiers:** operadores de modificação de modelagem de objetos

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

**Scene:** configuração dos parâmetros de edição de cena (p.ex.: unidade de medida)

**World:** configuração do ambiente no qual a cena está inserida

**Object:** configuração de parâmetros de edição do objeto atualmente selecionado na cena

**Constraints:** regras de restrição (para modelagem e/ou animação) para o objeto selecionado

**Modifiers:** operadores de modificação de modelagem de objetos

**Data:** manipuladores dos dados da geometria do objeto selecionado

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

**Scene:** configuração dos parâmetros de edição de cena (p.ex.: unidade de medida)

**World:** configuração do ambiente no qual a cena está inserida

**Object:** configuração de parâmetros de edição do objeto atualmente selecionado na cena

**Constraints:** regras de restrição (para modelagem e/ou animação) para o objeto selecionado

**Modifiers:** operadores de modificação de modelagem de objetos

**Data:** manipuladores dos dados da geometria do objeto selecionado

**Material:** configuração do(s) material(is) para o objeto selecionado

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

**Scene:** configuração dos parâmetros de edição de cena (p.ex.: unidade de medida)

**World:** configuração do ambiente no qual a cena está inserida

**Object:** configuração de parâmetros de edição do objeto atualmente selecionado na cena

**Constraints:** regras de restrição (para modelagem e/ou animação) para o objeto selecionado

**Modifiers:** operadores de modificação de modelagem de objetos

**Data:** manipuladores dos dados da geometria do objeto selecionado

**Material:** configuração do(s) material(is) para o objeto selecionado

**Texture:** configuração da(s) textura(s) associadas ao material editado na aba anterior

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

**Scene:** configuração dos parâmetros de edição de cena (p.ex.: unidade de medida)

**World:** configuração do ambiente no qual a cena está inserida

**Object:** configuração de parâmetros de edição do objeto atualmente selecionado na cena

**Constraints:** regras de restrição (para modelagem e/ou animação) para o objeto selecionado

**Modifiers:** operadores de modificação de modelagem de objetos

**Data:** manipuladores dos dados da geometria do objeto selecionado

**Material:** configuração do(s) material(is) para o objeto selecionado

**Texture:** configuração da(s) textura(s) associadas ao material editado na aba anterior

**Particles:** configuração do modelo de partículas

# Ferramentas Gráficas - Blender 3D

---

Outros contextos gráficos suportadas pelo Blender:

**Outliner:** ambiente para edição da hierarquia de objetos sendo editados na cena

**Properties:** ambiente de configuração da diversos aspectos relativos à cena atual

**Render:** configuração da cena/frame sendo editado

**RenderLayer:** subdivisão da cena em sub-camadas (para efeito de edição)

**Scene:** configuração dos parâmetros de edição de cena (p.ex.: unidade de medida)

**World:** configuração do ambiente no qual a cena está inserida

**Object:** configuração de parâmetros de edição do objeto atualmente selecionado na cena

**Constraints:** regras de restrição (para modelagem e/ou animação) para o objeto selecionado

**Modifiers:** operadores de modificação de modelagem de objetos

**Data:** manipuladores dos dados da geometria do objeto selecionado

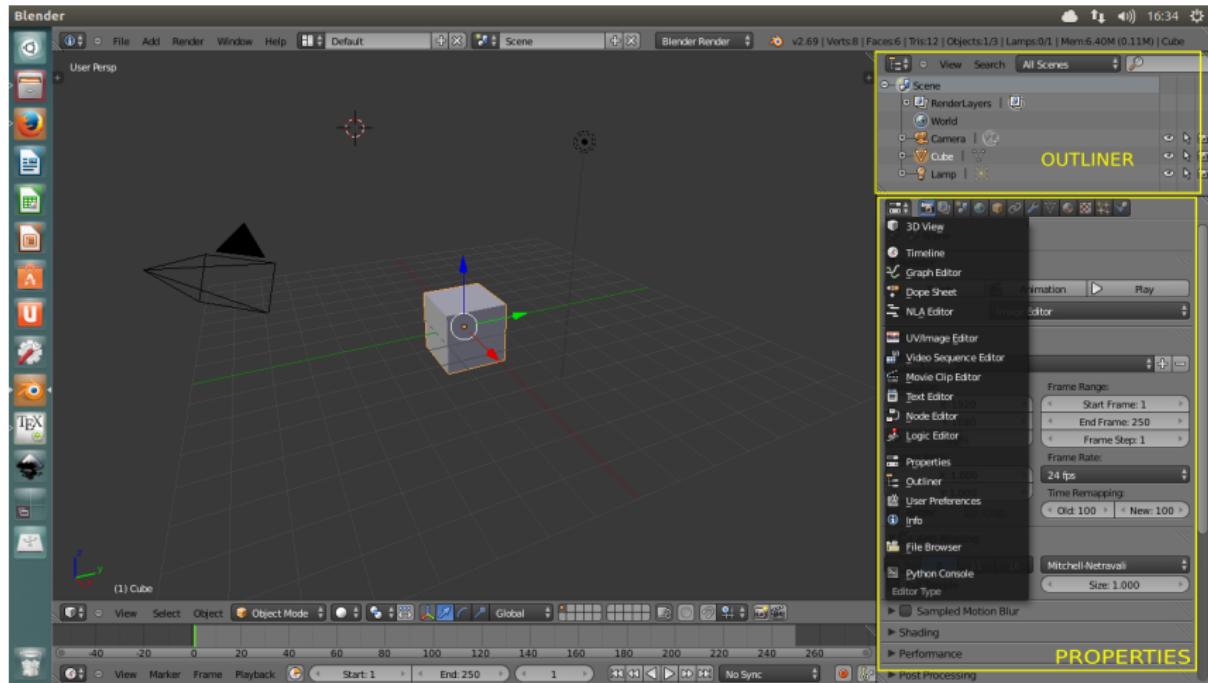
**Material:** configuração do(s) material(is) para o objeto selecionado

**Texture:** configuração da(s) textura(s) associadas ao material editado na aba anterior

**Particles:** configuração do modelo de partículas

**Physics:** configuração dos parâmetros de simulação física

# Ferramentas Gráficas - Blender 3D



# Ferramentas Gráficas - Blender 3D

---

Processo de criação de uma animação não-linear

**Linear:** apresenta cada ação<sup>1</sup> linearmente, uma após a outra

---

<sup>1</sup>Uma ação representa uma mídia específica: sequência de frames, arquivo de áudio, texto de uma legenda, etc.

# Ferramentas Gráficas - Blender 3D

---

Processo de criação de uma animação não-linear

**Linear:** apresenta cada ação<sup>1</sup> linearmente, uma após a outra

**Não-linear:** apresentar diversas ações em paralelo, combinando-as para gerar cada frame da animação resultante.

---

<sup>1</sup>Uma ação representa uma mídia específica: sequência de frames, arquivo de áudio, texto de uma legenda, etc.

# Ferramentas Gráficas - Blender 3D

Processo de criação de uma animação não-linear

**Linear:** apresenta cada ação<sup>1</sup> linearmente, uma após a outra

**Não-linear:** apresentar diversas ações em paralelo, combinando-as para gerar cada frame da animação resultante.



Linear



Não-Linear

---

<sup>1</sup>Uma ação representa uma mídia específica: sequência de frames, arquivo de áudio, texto de uma legenda, etc.

# Blender 3D - Modelagem Geométrica

---

- Modelagem geométrica é um conjunto de modelos para descrição matemática de primitivas gráficas (superfícies e volumes)

# Blender 3D - Modelagem Geométrica

---

- Modelagem geométrica é um conjunto de modelos para descrição matemática de primitivas gráficas (superfícies e volumes)
- Domínio contínuo vs discreto

# Blender 3D - Modelagem Geométrica

---

- Modelagem geométrica é um conjunto de modelos para descrição matemática de primitivas gráficas (superfícies e volumes)
- Domínio contínuo vs discreto
- Diversas abordagens:

# Blender 3D - Modelagem Geométrica

---

- Modelagem geométrica é um conjunto de modelos para descrição matemática de primitivas gráficas (superfícies e volumes)
- Domínio contínuo vs discreto
- Diversas abordagens:

**Procedural:** utiliza-se de equações matemáticas para calcular a aparência contínua das primitivas

# Blender 3D - Modelagem Geométrica

---

- Modelagem geométrica é um conjunto de modelos para descrição matemática de primitivas gráficas (superfícies e volumes)
- Domínio contínuo vs discreto
- Diversas abordagens:

**Procedural:** utiliza-se de equações matemáticas para calcular a aparência contínua das primitivas

**B-Rep:** (ou *boundary representation*) significa aproximar uma superfície contínua por uma malha poligonal discreta (2D ou 3D). No caso 3D, apenas os “limites” externos (casca) do objeto é aproximado

# Blender 3D - Modelagem Geométrica

---

- Modelagem geométrica é um conjunto de modelos para descrição matemática de primitivas gráficas (superfícies e volumes)
- Domínio contínuo vs discreto
- Diversas abordagens:

**Procedural:** utiliza-se de equações matemáticas para calcular a aparência contínua das primitivas

**B-Rep:** (ou *boundary representation*) significa aproximar uma superfície contínua por uma malha poligonal discreta (2D ou 3D). No caso 3D, apenas os “limites” externos (casca) do objeto é aproximado

**CSG:** (ou *constructive solid geometry*) significa aplicar operações booleanas (união, interseção ou diferença) a um conjunto de primitivas gráficas básicas, em uma estrutura em árvore, a fim de se criar objetos mais complexos.

# Blender 3D - Modelagem Geométrica

---

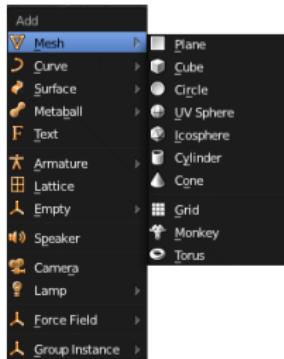
Modelagem Procedural por primitivas gráficas:

- Blender possui uma série de primitivas gráficas pré-implementadas:  
plano, esfera, cilindro, cone, grid, torus, cabeça de macaco (Suzanne)

# Blender 3D - Modelagem Geométrica

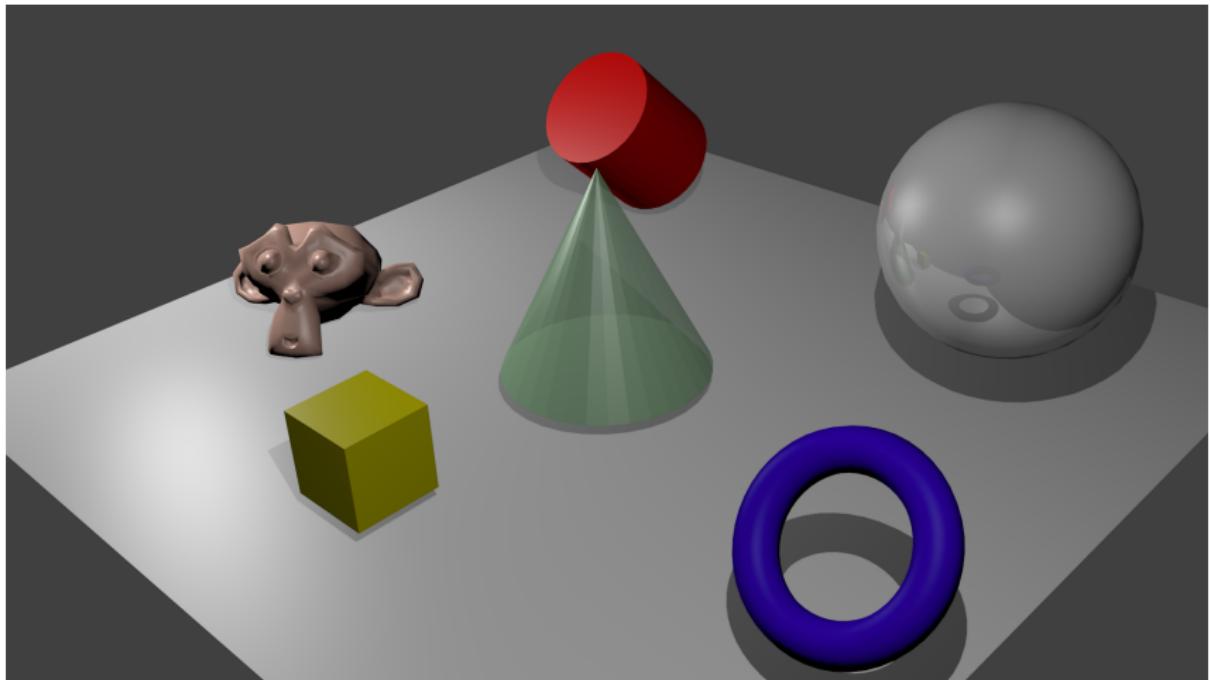
Modelagem Procedural por primitivas gráficas:

- Blender possui uma série de primitivas gráficas pré-implementadas: plano, esfera, cilindro, cone, grid, torus, cabeça de macaco (Suzanne)
- Para adicionar uma primitiva na 3D View:
  - Acesse menu Add / Mesh / <Tipo da Primitiva> ou
  - Tecla de atalho: Shift + A



# Blender 3D - Modelagem Geométrica

---



## Blender 3D - Edição de B-Rep

---

**Extrusão** transformar uma *mediatriz* (curva base) em um sólido 3D através de uma *diretriz*. Tipos: translacional e rotacional

## Blender 3D - Edição de B-Rep

---

**Extrusão** transformar uma *mediatriz* (curva base) em um sólido 3D através de uma *diretriz*. Tipos: translacional e rotacional

**Subdivisão** permite criar novas faces subdividindo as existentes

# Blender 3D - Edição de B-Rep

---

**Extrusão** transformar uma *mediatriz* (curva base) em um sólido 3D através de uma *diretriz*. Tipos: translacional e rotacional

**Subdivisão** permite criar novas faces subdividindo as existentes

**Recorte** permite criar novas faces, recortando as arestas existentes

# Blender 3D - Edição de B-Rep

---

**Extrusão** transformar uma *mediatriz* (curva base) em um sólido 3D através de uma *diretriz*. Tipos: translacional e rotacional

**Subdivisão** permite criar novas faces subdividindo as existentes

**Recorte** permite criar novas faces, recortando as arestas existentes

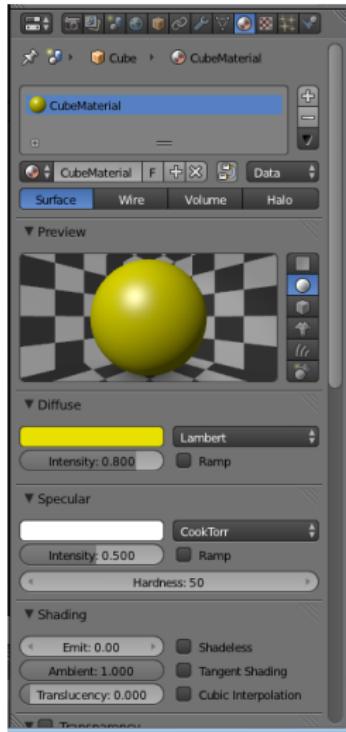
**Modificadores** permite alterar as superfícies através de funções de modificação:

**Array** cria uma série de cópias do objeto

**Subsurf** suaviza o volume poligonal através da criação de uma sub-malha em resolução mais alta

**Bevel** arredonda os cantos retos da malha

# Blender 3D - Modelagem Geométrica



# Blender 3D - Modelagem Geométrica

---

**Diffuse:** especifica a cor da superfície do material

# Blender 3D - Modelagem Geométrica

---

**Diffuse:** especifica a cor da superfície do material

**Specular:** especifica a quantidade de luz refletida na superfície (brilho)

# Blender 3D - Modelagem Geométrica

---

**Diffuse:** especifica a cor da superfície do material

**Specular:** especifica a quantidade de luz refletida na superfície (brilho)

**Hardness:** determina o grau de espalhamento da especularidade

# Blender 3D - Modelagem Geométrica

---

**Diffuse:** especifica a cor da superfície do material

**Specular:** especifica a quantidade de luz refletida na superfície (brilho)

**Hardness:** determina o grau de espalhamento da especularidade

**Transparency:** especifica que o material é transparente

**Z Transparent:** simplesmente combina a cor com o cenário de fundo (*blending*)

# Blender 3D - Modelagem Geométrica

---

**Diffuse:** especifica a cor da superfície do material

**Specular:** especifica a quantidade de luz refletida na superfície (brilho)

**Hardness:** determina o grau de espalhamento da especularidade

**Transparency:** especifica que o material é transparente

**Z Transparent:** simplesmente combina a cor com o cenário de fundo (*blending*)

**Raytrace:** calcula a trajetória dos raios de luz ao atravessar a superfície (refração da luz)

# Blender 3D - Modelagem Geométrica

---

**Diffuse:** especifica a cor da superfície do material

**Specular:** especifica a quantidade de luz refletida na superfície (brilho)

**Hardness:** determina o grau de espalhamento da especularidade

**Transparency:** especifica que o material é transparente

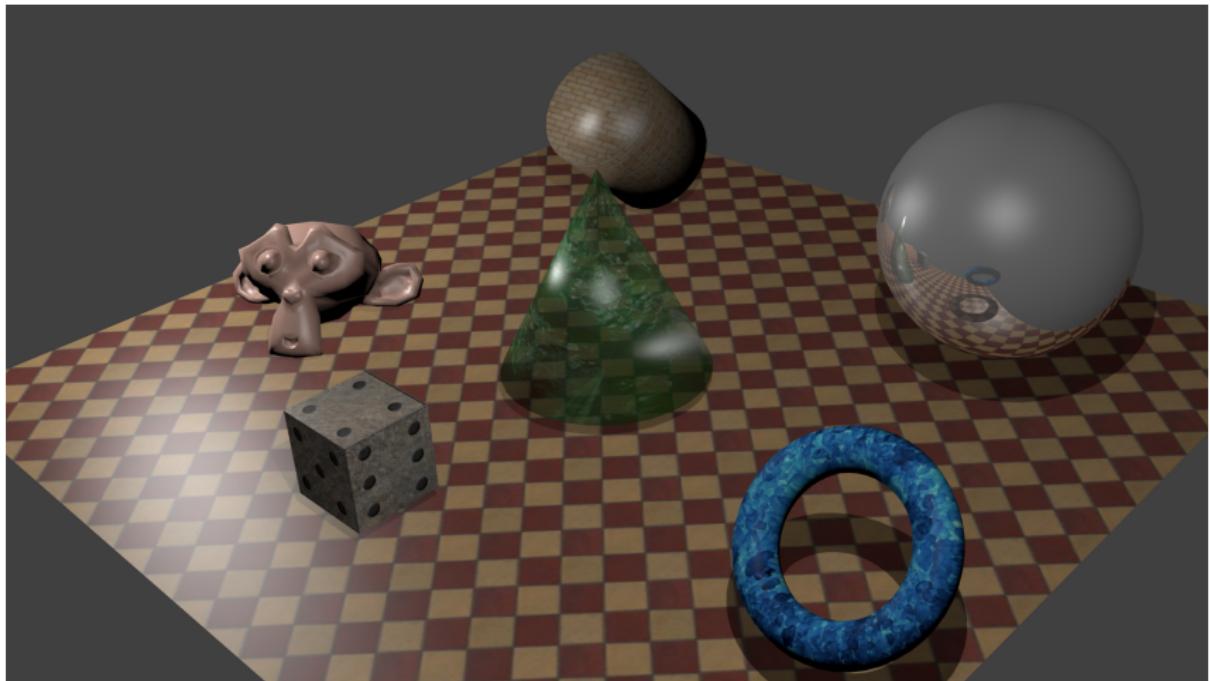
**Z Transparent:** simplesmente combina a cor com o cenário de fundo (*blending*)

**Raytrace:** calcula a trajetória dos raios de luz ao atravessar a superfície (refração da luz)

**Mirror:** calcula o espelhamento da luz sobre a superfície

# Blender 3D - Modelagem Geométrica

---



# Blender 3D - Modelagem Geométrica

---

- Mapeamento de texturas representa uma função responsável por substituir o modelo de iluminação difusa (cor sólida) por um mapa de *pixels* coloridos (imagem *bitmap*)

# Blender 3D - Modelagem Geométrica

---

- Mapeamento de texturas representa uma função responsável por substituir o modelo de iluminação difusa (cor sólida) por um mapa de *pixels* coloridos (imagem *bitmap*)
- Para malhas poligonais (B-Reps) duas técnicas são bastante populares:  
**Mapeamento quadrático** que se baseia por equações que determinam as coordenadas dos pixels na superfície do objeto poligonal

# Blender 3D - Modelagem Geométrica

---

- Mapeamento de texturas representa uma função responsável por substituir o modelo de iluminação difusa (cor sólida) por um mapa de *pixels* coloridos (imagem *bitmap*)
- Para malhas poligonais (B-Reps) duas técnicas são bastante populares:  
**Mapeamento quadrático** que se baseia por equações que determinam as coordenadas dos pixels na superfície do objeto poligonal  
**Mapeamento UV** que se baseia na definição manual das coordenadas de textura para cada vértice da malha poligonal do objeto

# Blender 3D - Modelagem Geométrica



# Blender 3D - Rendering

---

- Processo de geração de imagens à partir do modelo 3D produzido

# Blender 3D - Rendering

---

- Processo de geração de imagens à partir do modelo 3D produzido
- Elementos envolvidos: objetos 3D, materiais, fontes de iluminação e câmera virtual

# Blender 3D - Rendering

---

- Processo de geração de imagens à partir do modelo 3D produzido
- Elementos envolvidos: objetos 3D, materiais, fontes de iluminação e câmera virtual
- Modelo de Iluminação: determinação da intensidade de cor e brilho de cada pixel na imagem resultante. Tipos: local vs global

# Blender 3D - Rendering

---

- Processo de geração de imagens à partir do modelo 3D produzido
- Elementos envolvidos: objetos 3D, materiais, fontes de iluminação e câmera virtual
- Modelo de Iluminação: determinação da intensidade de cor e brilho de cada pixel na imagem resultante. Tipos: local vs global
  - Local:** considera apenas os pontos que recebem luz provenientes diretamente da fonte de luz (uso: Blender Render)

# Blender 3D - Rendering

---

- Processo de geração de imagens à partir do modelo 3D produzido
- Elementos envolvidos: objetos 3D, materiais, fontes de iluminação e câmera virtual
- Modelo de Iluminação: determinação da intensidade de cor e brilho de cada pixel na imagem resultante. Tipos: local vs global
  - Local:** considera apenas os pontos que recebem luz provenientes diretamente da fonte de luz (uso: Blender Render)
  - Global:** considera também reflexões internas no ambiente = mais preciso (uso: Blender Cycles)

# Blender 3D - Rendering

---

Tipos de fontes de luz suportadas pelo Blender:

# Blender 3D - Rendering

---

Tipos de fontes de luz suportadas pelo Blender:

**Point:** emite luz em todas as direções no espaço à partir de um ponto

# Blender 3D - Rendering

---

Tipos de fontes de luz suportadas pelo Blender:

**Point:** emite luz em todas as direções no espaço à partir de um ponto

**Sun:** emite raios de luz paralelos sem origem definida

# Blender 3D - Rendering

---

Tipos de fontes de luz suportadas pelo Blender:

**Point:** emite luz em todas as direções no espaço à partir de um ponto

**Sun:** emite raios de luz paralelos sem origem definida

**Spot:** emite luz à partir de um ponto em uma direção específica com um determinado ângulo de abertura (feixe de luz)

# Blender 3D - Rendering

---

Tipos de fontes de luz suportadas pelo Blender:

**Point:** emite luz em todas as direções no espaço à partir de um ponto

**Sun:** emite raios de luz paralelos sem origem definida

**Spot:** emite luz à partir de um ponto em uma direção específica com um determinado ângulo de abertura (feixe de luz)

**Hemi:** emite luz partindo de um ‘hemisfério’ de uma esfera. Serve para simulação espalhamento da luz como um dia nublado

# Blender 3D - Rendering

---

Tipos de fontes de luz suportadas pelo Blender:

**Point:** emite luz em todas as direções no espaço à partir de um ponto

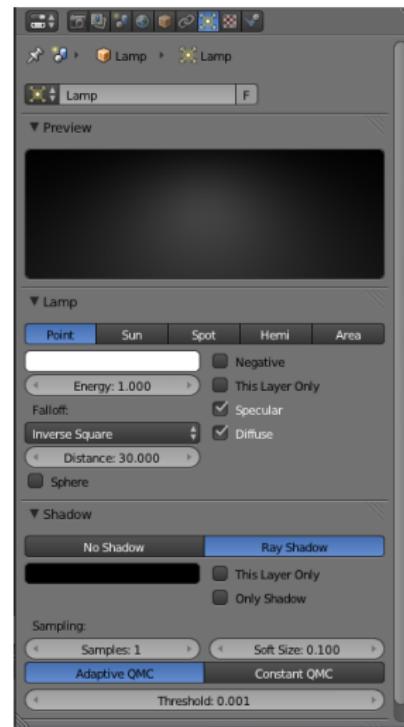
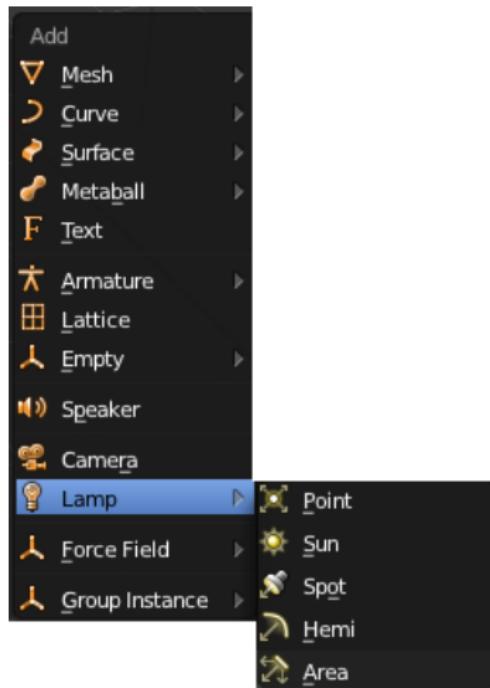
**Sun:** emite raios de luz paralelos sem origem definida

**Spot:** emite luz à partir de um ponto em uma direção específica com um determinado ângulo de abertura (feixe de luz)

**Hemi:** emite luz partindo de um ‘hemisfério’ de uma esfera. Serve para simulação espalhamento da luz como um dia nublado

**Area:** emite luz partindo de uma ‘área’ pré-determinada (superfície) como, por exemplo: um *outdoor* ou uma janela

# Blender 3D - Rendering



# Blender 3D - Rendering

---

Tipos de câmeras virtuais:

# Blender 3D - Rendering

---

Tipos de câmeras virtuais:

**Ortográfica:** projeta o mundo 3D para um plano através de um paralelepípedo (raios projetores parelelos)

# Blender 3D - Rendering

---

Tipos de câmeras virtuais:

**Ortográfica:** projeta o mundo 3D para um plano através de um paralelepípedo (*raios projetores parelelos*)

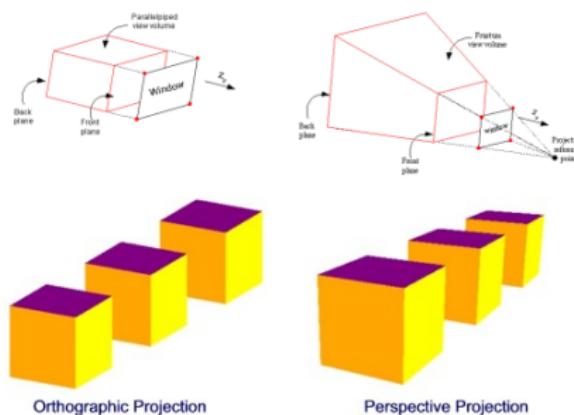
**Perspectiva:** projeta o mundo 3D para um plano através de raios projetores convergentes (*frustum*)

# Blender 3D - Rendering

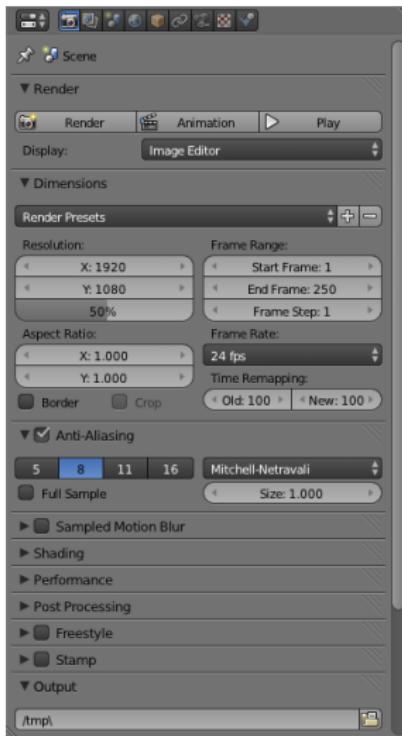
Tipos de câmeras virtuais:

**Ortográfica:** projeta o mundo 3D para um plano através de um paralelepípedo (*raios projetores parelelos*)

**Perspectiva:** projeta o mundo 3D para um plano através de raios projetores convergentes (*frustum*)



# Blender 3D - Rendering



# Blender 3D - Rendering Fotorealístico

---

Novos conceitos envolvidos:

# Blender 3D - Rendering Fotorealístico

---

Novos conceitos envolvidos:

**Materiais:** são modelos matemáticos compostos por um grafo de modificadores padrão (*shaders*)

# Blender 3D - Rendering Fotorealístico

---

Novos conceitos envolvidos:

**Materiais:** são modelos matemáticos compostos por um grafo de modificadores padrão (*shaders*)

**Fonte de Luz:** é um objeto associado a um material especial denominado **emission**

# Blender 3D - Rendering Fotorealístico

---

Novos conceitos envolvidos:

**Materiais:** são modelos matemáticos compostos por um grafo de modificadores padrão (*shaders*)

**Fonte de Luz:** é um objeto associado a um material especial denominado **emission**

**Sampling:** como o rendering é realizado de forma iterativa (em ciclos), este parâmetro determinada a precisão do processo (quantidade de ciclos de render)

# Blender 3D - Rendering Fotorealístico

---

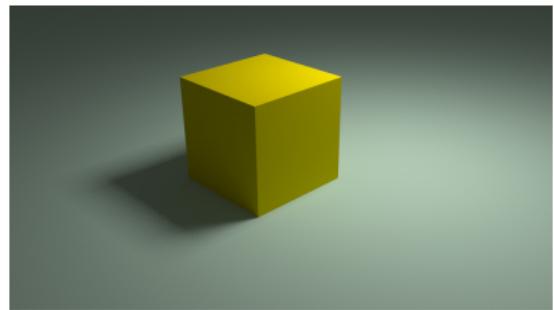
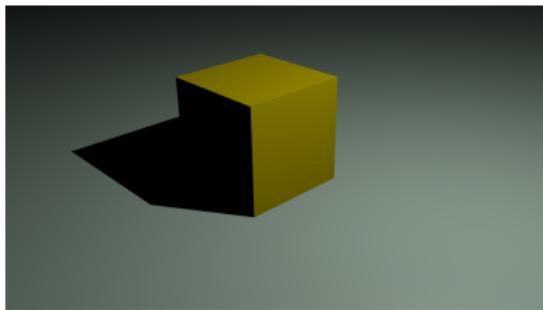


Figura 2: Tempo de Rendering: Internal = 64ms e Cycles<sup>2</sup> = 36s78ms

---

<sup>2</sup>150 ciclos  
78 of 264

# Blender 3D - Rendering Fotorealístico

---

Tipos de *shaders* atualmente suportados pelo Blender:

# Blender 3D - Rendering Fotorealístico

---

Tipos de *shaders* atualmente suportados pelo Blender:

**Diffuse BSDF:** material para definir superfícies difusas (cores sólidas)

# Blender 3D - Rendering Fotorealístico

---

Tipos de *shaders* atualmente suportados pelo Blender:

**Diffuse BSDF:** material para definir superfícies difusas (cores sólidas)

**Transparent BSDF:** define superfícies transparentes (sem refração da luz)

# Blender 3D - Rendering Fotorealístico

---

Tipos de *shaders* atualmente suportados pelo Blender:

**Diffuse BSDF:** material para definir superfícies difusas (cores sólidas)

**Transparent BSDF:** define superfícies transparentes (sem refração da luz)

**Velvet BSDF:** define a aparência do tecido veludo

# Blender 3D - Rendering Fotorealístico

---

Tipos de *shaders* atualmente suportados pelo Blender:

**Diffuse BSDF:** material para definir superfícies difusas (cores sólidas)

**Transparent BSDF:** define superfícies transparentes (sem refração da luz)

**Velvet BSDF:** define a aparência do tecido veludo

**Subsurface Scattering:** define superfícies parcialmente difusas, com uma componente transparente

# Blender 3D - Rendering Fotorealístico

---

Tipos de *shaders* atualmente suportados pelo Blender:

**Diffuse BSDF:** material para definir superfícies difusas (cores sólidas)

**Transparent BSDF:** define superfícies transparentes (sem refração da luz)

**Velvet BSDF:** define a aparência do tecido veludo

**Subsurface Scattering:** define superfícies parcialmente difusas, com uma componente transparente

**Glossy BSDF:** define superfícies polidas que refletem luz direcionalmente como metal ou espelhos

# Blender 3D - Rendering Fotorealístico

---

Tipos de *shaders* atualmente suportados pelo Blender:

**Diffuse BSDF:** material para definir superfícies difusas (cores sólidas)

**Transparent BSDF:** define superfícies transparentes (sem refração da luz)

**Velvet BSDF:** define a aparência do tecido veludo

**Subsurface Scattering:** define superfícies parcialmente difusas, com uma componente transparente

**Glossy BSDF:** define superfícies polidas que refletem luz direcionalmente como metal ou espelhos

**Hair BSDF:** define o material para objetos partículas estáticas do tipo 'cabelos'

# Blender 3D - Rendering Fotorealístico

---

Tipos de *shaders* atualmente suportados pelo Blender:

**Diffuse BSDF:** material para definir superfícies difusas (cores sólidas)

**Transparent BSDF:** define superfícies transparentes (sem refração da luz)

**Velvet BSDF:** define a aparência do tecido veludo

**Subsurface Scattering:** define superfícies parcialmente difusas, com uma componente transparente

**Glossy BSDF:** define superfícies polidas que refletem luz direcionalmente como metal ou espelhos

**Hair BSDF:** define o material para objetos partículas estáticas do tipo 'cabelos'

**Background:** só é válido se associado ao componente *world* e define a cor da luz de fundo do ambiente

# Blender 3D - Rendering Fotorealístico

---

Tipos de *shaders* atualmente suportados pelo Blender:

**Diffuse BSDF:** material para definir superfícies difusas (cores sólidas)

**Transparent BSDF:** define superfícies transparentes (sem refração da luz)

**Velvet BSDF:** define a aparência do tecido veludo

**Subsurface Scattering:** define superfícies parcialmente difusas, com uma componente transparente

**Glossy BSDF:** define superfícies polidas que refletem luz direcionalmente como metal ou espelhos

**Hair BSDF:** define o material para objetos partículas estáticas do tipo 'cabelos'

**Background:** só é válido se associado ao componente *world* e define a cor da luz de fundo do ambiente

**Refraction BSDF:** define materiais capazes de transmitir luz (refração da luz)

# Blender 3D - Rendering Fotorealístico

---

Tipos de *shaders* atualmente suportados pelo Blender:

**Diffuse BSDF:** material para definir superfícies difusas (cores sólidas)

**Transparent BSDF:** define superfícies transparentes (sem refração da luz)

**Velvet BSDF:** define a aparência do tecido veludo

**Subsurface Scattering:** define superfícies parcialmente difusas, com uma componente transparente

**Glossy BSDF:** define superfícies polidas que refletem luz direcionalmente como metal ou espelhos

**Hair BSDF:** define o material para objetos partículas estáticas do tipo 'cabelos'

**Background:** só é válido se associado ao componente *world* e define a cor da luz de fundo do ambiente

**Refraction BSDF:** define materiais capazes de transmitir luz (refração da luz)

**Toon BSDF:** define materiais do tipo desenho animado (*cartoon*), ou seja, restrição da paleta de cores (como tintas)

# Blender 3D - Rendering Fotorealístico

---

**Ambient Occlusion:** define o quanto uma superfície é afetada por reflexões internas da luz no ambiente

# Blender 3D - Rendering Fotorealístico

---

**Ambient Occlusion:** define o quanto uma superfície é afetada por reflexões internas da luz no ambiente

**Holdout:** permite criar “furos” no material como se apenas parte da superfície fosse transparente

# Blender 3D - Rendering Fotorealístico

---

**Ambient Occlusion:** define o quanto uma superfície é afetada por reflexões internas da luz no ambiente

**Holdout:** permite criar “furos” no material como se apenas parte da superfície fosse transparente

**Glass BSDF:** define a aparência de vidro à superfície. Mais eficiente quando combinado com uma transparência

# Blender 3D - Rendering Fotorealístico

---

**Ambient Occlusion:** define o quanto uma superfície é afetada por reflexões internas da luz no ambiente

**Holdout:** permite criar “furos” no material como se apenas parte da superfície fosse transparente

**Glass BSDF:** define a aparência de vidro à superfície. Mais eficiente quando combinado com uma transparência

**Translucent BSDF:** define materiais translúcidos

# Blender 3D - Rendering Fotorealístico

---

**Ambient Occlusion:** define o quanto uma superfície é afetada por reflexões internas da luz no ambiente

**Holdout:** permite criar “furos” no material como se apenas parte da superfície fosse transparente

**Glass BSDF:** define a aparência de vidro à superfície. Mais eficiente quando combinado com uma transparência

**Translucent BSDF:** define materiais translúcidos

**Mix Shader:** permite combinar dois ou mais materiais

# Blender 3D - Rendering Fotorealístico

---

**Ambient Occlusion:** define o quanto uma superfície é afetada por reflexões internas da luz no ambiente

**Holdout:** permite criar “furos” no material como se apenas parte da superfície fosse transparente

**Glass BSDF:** define a aparência de vidro à superfície. Mais eficiente quando combinado com uma transparência

**Translucent BSDF:** define materiais translúcidos

**Mix Shader:** permite combinar dois ou mais materiais

**Emission:** define uma fonte emissora de luz (energia luminosa)

# Blender 3D - Rendering Fotorealístico

---

**Ambient Occlusion:** define o quanto uma superfície é afetada por reflexões internas da luz no ambiente

**Holdout:** permite criar “furos” no material como se apenas parte da superfície fosse transparente

**Glass BSDF:** define a aparência de vidro à superfície. Mais eficiente quando combinado com uma transparência

**Translucent BSDF:** define materiais translúcidos

**Mix Shader:** permite combinar dois ou mais materiais

**Emission:** define uma fonte emissora de luz (energia luminosa)

**Anisotropic BSDF:** reflexão da luz com base no mapeamento UV ativo para a superfície

# Blender 3D - Rendering Fotorealístico

---

**Ambient Occlusion:** define o quanto uma superfície é afetada por reflexões internas da luz no ambiente

**Holdout:** permite criar “furos” no material como se apenas parte da superfície fosse transparente

**Glass BSDF:** define a aparência de vidro à superfície. Mais eficiente quando combinado com uma transparência

**Translucent BSDF:** define materiais translúcidos

**Mix Shader:** permite combinar dois ou mais materiais

**Emission:** define uma fonte emissora de luz (energia luminosa)

**Anisotropic BSDF:** reflexão da luz com base no mapeamento UV ativo para a superfície

<http://wiki.blender.org/index.php/Doc:2.6/Manual/Render/Cycles/Nodes/Shaders>

# Exercício Prático Individual - EPI #1

---

**Nome:** *Modelagem de Cenários*

**Objetivo:** treinamento de uso da ferramenta Blender, projeto de uma animação, construção dos cenários

**Enunciado:**

1. escolher/confeccionar um roteiro a ser animado (estória)
2. modelar os elementos e personagens

**Requisitos:**

- Estória deve conter, no mínimo, dois personagens
- Descrição detalhada do cenário
- Não é obrigatório conter diálogos

**Prazo para Entrega:** 7 dias (filme + estória + slides ilustrando o processo)

**Peso:** 15% da média semestral

# Keyframing

- Animação Pose-to-Pose (Quadros-chave)
- Linha do tempo
- Curvas de Animação

# Keyframing

---

- Animar significa criar comportamentos variáveis para um determinado atributo em relação ao tempo

# Keyframing

---

- Animar significa criar comportamentos variáveis para um determinado atributo em relação ao tempo

$$\text{Atributo}_{t+1} = \text{Atributo}_t + \Delta$$

# Keyframing

---

- Animar significa criar comportamentos variáveis para um determinado atributo em relação ao tempo

$$\text{Atributo}_{t+1} = \text{Atributo}_t + \Delta$$

- O conjunto de valores dos atributos animáveis a cada instante de tempo  $t_i$  é denominado **pose**

# Keyframing

---

- Animar significa criar comportamentos variáveis para um determinado atributo em relação ao tempo

$$\text{Atributo}_{t+1} = \text{Atributo}_t + \Delta$$

- O conjunto de valores dos atributos animáveis a cada instante de tempo  $t_i$  é denominado **pose**

$$pose = \left\{ \begin{array}{l} \text{atributo}_1, \\ \text{atributo}_2, \\ \dots, \\ \text{atributo}_N \end{array} \right.$$

# Keyframing

---

- Animar significa criar comportamentos variáveis para um determinado atributo em relação ao tempo

$$\text{Atributo}_{t+1} = \text{Atributo}_t + \Delta$$

- O conjunto de valores dos atributos animáveis a cada instante de tempo  $t_i$  é denominado **pose**

$$pose = \left\{ \begin{array}{l} \text{atributo}_1, \\ \text{atributo}_2, \\ \dots, \\ \text{atributo}_N \end{array} \right.$$

- O processo de animação por ‘quadros-chave’ consiste na criação de uma sequência de poses no tempo para posterior preenchimento dos espaços vazios intermediários (*Pose-to-Pose Animation*)

# Keyframing

---

- O processo de animação consiste de uma sequência em três etapas:

# Keyframing

---

- O processo de animação consiste de uma sequência em três etapas:  
**Blocking:** etapa responsável pela definição da *pose* do personagem em instantes de tempos conhecidos (*Timing*) e no posicionamento específico (*Spacing*) de cada pose

# Keyframing

---

- O processo de animação consiste de uma sequência em três etapas:
  - Blocking:** etapa responsável pela definição da *pose* do personagem em instantes de tempos conhecidos (*Timing*) e no posicionamento específico (*Spacing*) de cada pose
  - Splining:** etapa responsável pela definição do método interpolador (curvas) para cada par consecutivo de poses definidas pela etapa anterior

# Keyframing

---

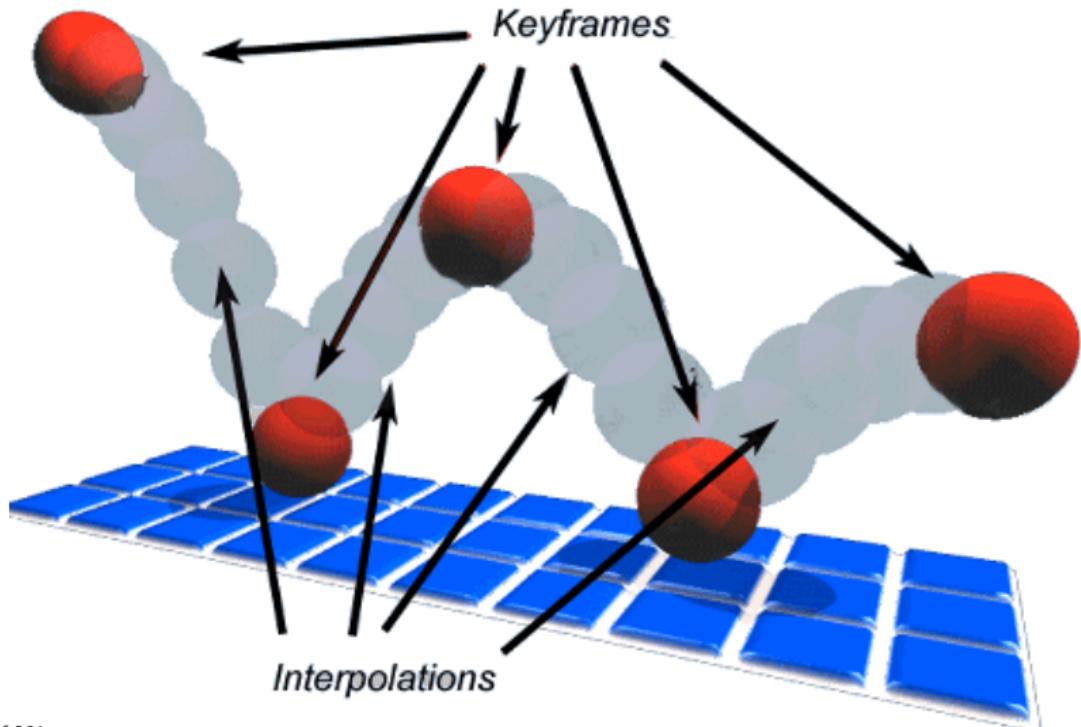
- O processo de animação consiste de uma sequência em três etapas:

**Blocking:** etapa responsável pela definição da *pose* do personagem em instantes de tempos conhecidos (*Timing*) e no posicionamento específico (*Spacing*) de cada pose

**Splining:** etapa responsável pela definição do método interpolador (curvas) para cada par consecutivo de poses definidas pela etapa anterior

**Polishing:** etapa responsável pela suavização das curvas de interpolação, *timing* e *spacing* para cada personagem durante a animação

# Keyframing



# Keyframing em Blender

---

**3D View:** ambiente para construção (modelagem) da cena e descrição interativa das poses (*spacing*)

# Keyframing em Blender

---

**3D View:** ambiente para construção (modelagem) da cena e descrição interativa das poses (*spacing*)

**Properties:** ambiente para edição de valores para atributos específicos da animação

# Keyframing em Blender

---

**3D View:** ambiente para construção (modelagem) da cena e descrição interativa das poses (*spacing*)

**Properties:** ambiente para edição de valores para atributos específicos da animação

**Timeline:** ambiente para edição de quadros-chave e parâmetros da animação resultante

# Keyframing em Blender

---

**3D View:** ambiente para construção (modelagem) da cena e descrição interativa das poses (*spacing*)

**Properties:** ambiente para edição de valores para atributos específicos da animação

**Timeline:** ambiente para edição de quadros-chave e parâmetros da animação resultante

**Dope Sheet Editor:** ambiente para editoração de quadros-chave em termos de *timing*

# Keyframing em Blender

---

**3D View:** ambiente para construção (modelagem) da cena e descrição interativa das poses (*spacing*)

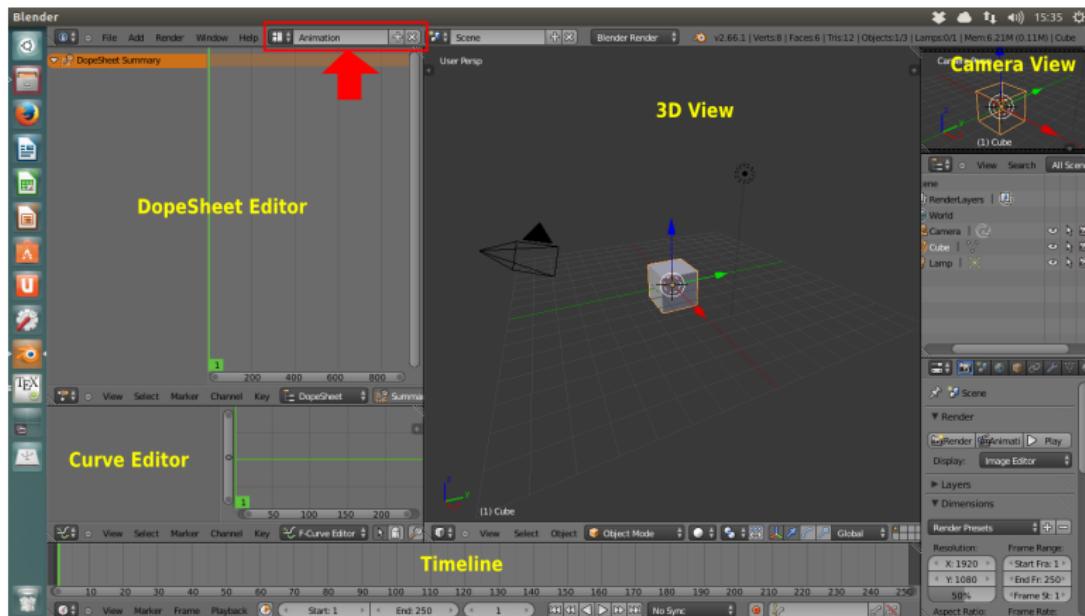
**Properties:** ambiente para edição de valores para atributos específicos da animação

**Timeline:** ambiente para edição de quadros-chave e parâmetros da animação resultante

**Dope Sheet Editor:** ambiente para editoração de quadros-chave em termos de *timing*

**F-Curve Editor:** ambiente para ajustes detalhados no comportamento animável de cada atributo (*splining* e *polishing*)

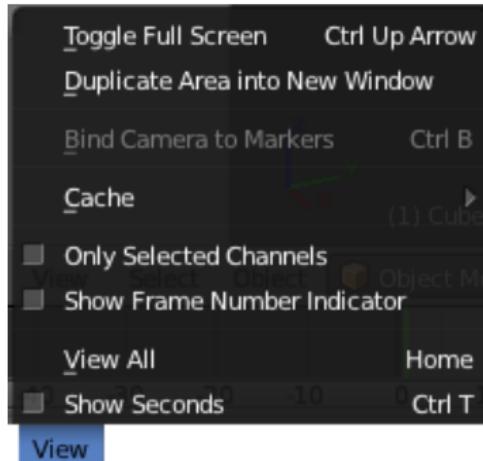
# Keyframing em Blender



# Keyframing em Blender - Timeline



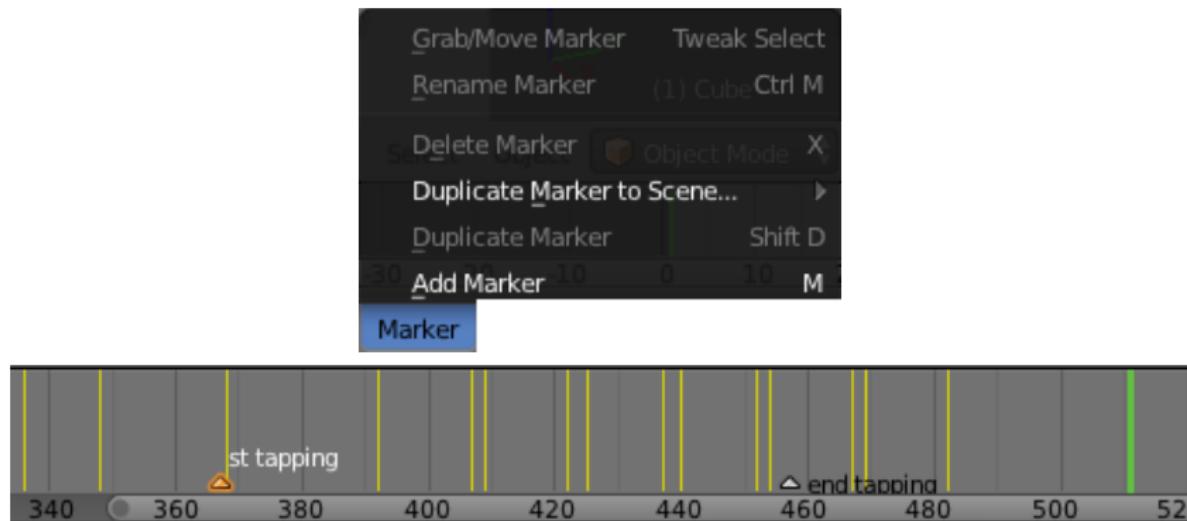
# Keyframing em Blender - Timeline



- Conjunto de opções responsáveis pela forma de visualização das informações da timeline como:
- Exibir apenas alguns canais selecionados da animação
- Exibir indicador de tempo transcorrido da animação
- Exibir lista de frames e/ou tempo em segundos

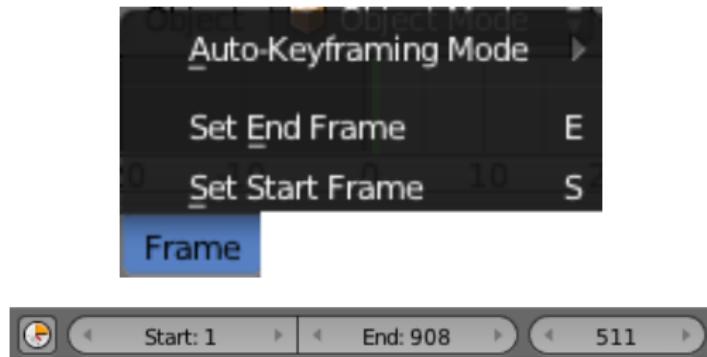
# Keyframing em Blender - Timeline

- *Markers* são anotações que se faz na timeline para indicar rótulos a momentos específicos da ação



# Keyframing em Blender - Timeline

- No menu *Frame* editam-se os parâmetros do tamanho da timeline: frame de início e fim e o modo de adição de novos frames



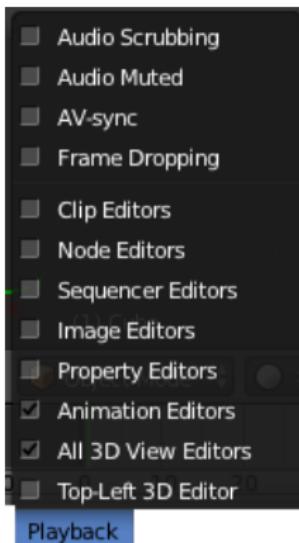
## Keyframing em Blender - Timeline

---

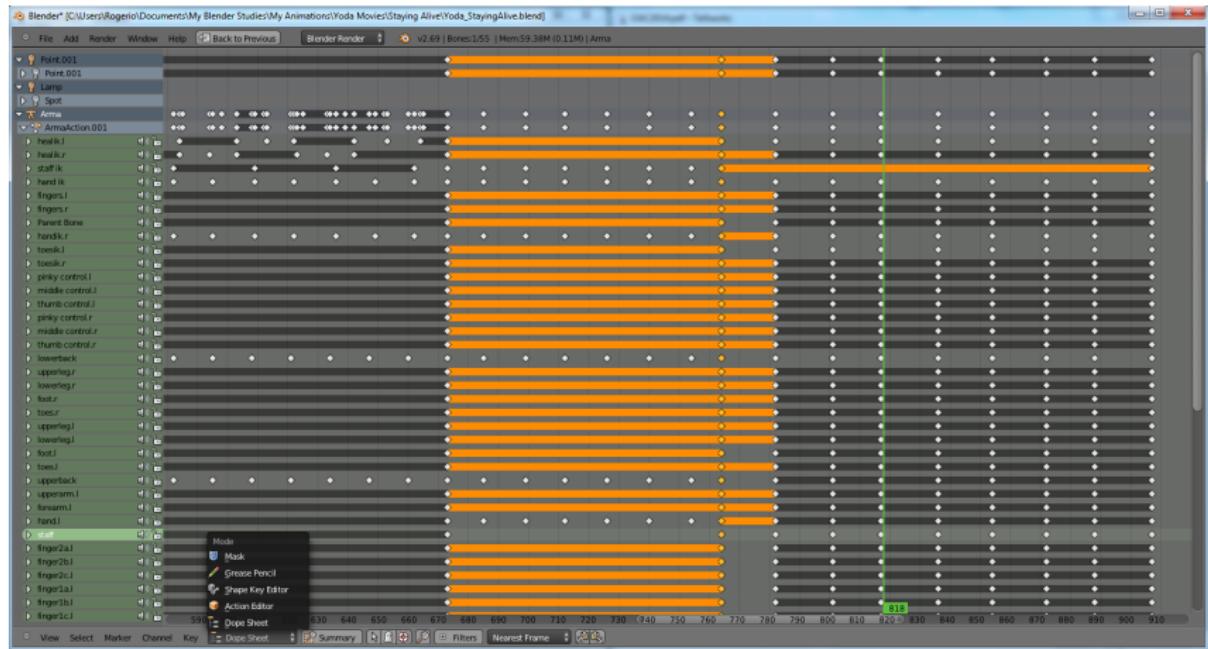
- O menu *playback* permite configurar as opções de reprodução *inline* (em tempo real) da timeline sendo editada

# Keyframing em Blender - Timeline

- O menu *playback* permite configurar as opções de reprodução *inline* (em tempo real) da timeline sendo editada
- Parâmetros como: arrasto de som (*scratch*), sincronização áudio/vídeo; bem como, quais editores são afetados pela reprodução da timeline



# Keyframing em Blender - Dope Sheet



# Keyframing em Blender - Dope Sheet

---

**Dope Sheet** Responsável pela edição de grupos de quadros-chave em relação ao tempo (*timing*)

# Keyframing em Blender - Dope Sheet

---

**Dope Sheet** Responsável pela edição de grupos de quadros-chave em relação ao tempo (*timing*)

**Action Editor** Responsável pela edição de grupos de quadros-chave para canais específicos de um personagem específico. Permite a criação de ações básicas reutilizáveis na timeline (ciclo de caminhada, ciclo de corrida, decolagem/pouso de aeronaves, etc.)

# Keyframing em Blender - Dope Sheet

---

**Dope Sheet** Responsável pela edição de grupos de quadros-chave em relação ao tempo (*timing*)

**Action Editor** Responsável pela edição de grupos de quadros-chave para canais específicos de um personagem específico. Permite a criação de ações básicas reutilizáveis na timeline (ciclo de caminhada, ciclo de corrida, decolagem/pouso de aeronaves, etc.)

**Shape Keys** Permite a modelagem e animação de deformações de malhas poligonais. Utilizado para expressões faciais e objetos flexíveis em geral

# Keyframing em Blender - Dope Sheet

---

**Dope Sheet** Responsável pela edição de grupos de quadros-chave em relação ao tempo (*timing*)

**Action Editor** Responsável pela edição de grupos de quadros-chave para canais específicos de um personagem específico. Permite a criação de ações básicas reutilizáveis na timeline (ciclo de caminhada, ciclo de corrida, decolagem/pouso de aeronaves, etc.)

**Shape Keys** Permite a modelagem e animação de deformações de malhas poligonais. Utilizado para expressões faciais e objetos flexíveis em geral

**Grease Pencil** ferramenta de anotações gráficas no ambiente de animação (funciona em vários editores diferentes)

# Keyframing em Blender - Dope Sheet - Action Editor

---

- Associa-se uma timeline e seus respectivos quadros-chave (para um dado personagem) com um rótulo

# Keyframing em Blender - Dope Sheet - Action Editor

---

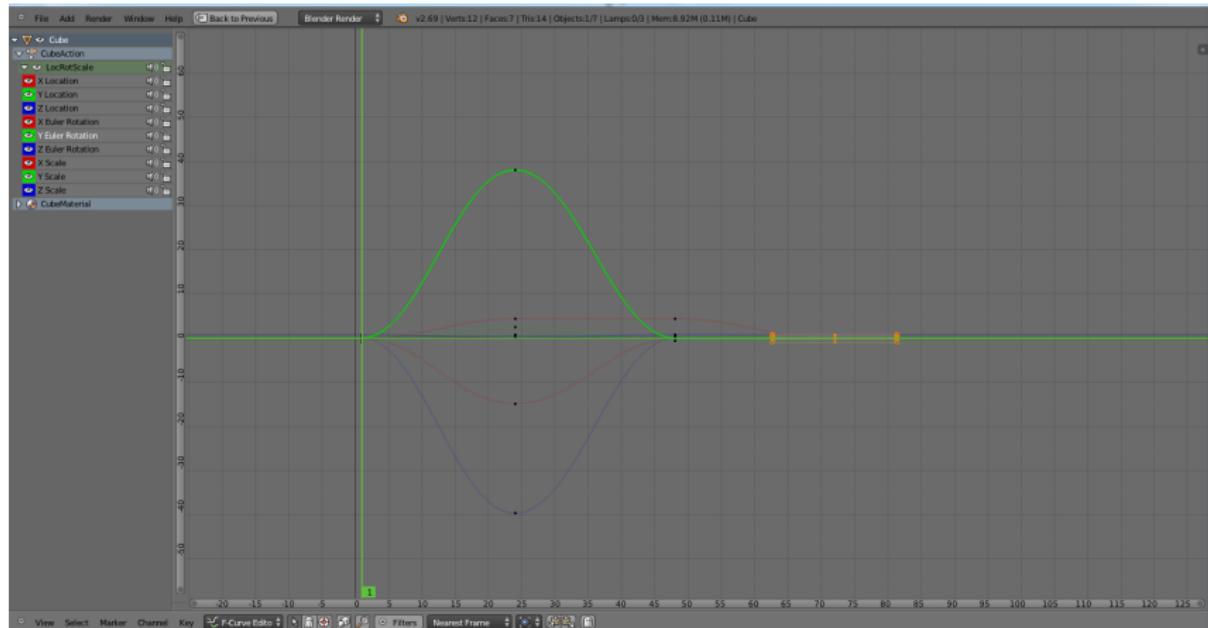
- Associa-se uma timeline e seus respectivos quadros-chave (para um dado personagem) com um rótulo
- Posteriormente, através desse rótulo, é possível se **instanciar** objetos de uma dada ação no **NLA editor** para se evitar o retrabalho de animar a mesma ação várias vezes, a instantes de tempos diferentes da timeline.  
Exemplo: ciclo de caminhada

# Keyframing em Blender - Dope Sheet - Action Editor

- Associa-se uma timeline e seus respectivos quadros-chave (para um dado personagem) com um rótulo
- Posteriormente, através desse rótulo, é possível se **instanciar** objetos de uma dada ação no **NLA editor** para se evitar o retrabalho de animar a mesma ação várias vezes, a instantes de tempos diferentes da timeline.  
Exemplo: ciclo de caminhada



# Keyframing em Blender - Graph Editor



## Keyframing em Blender - Graph Editor

---

- Mostra cada canal animado como uma curva que descreve o comportamento do valor em relação ao tempo

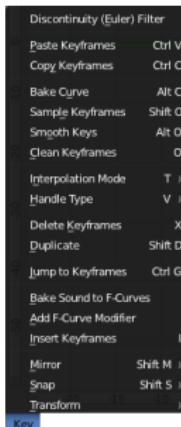
## Keyframing em Blender - Graph Editor

---

- Mostra cada canal animado como uma curva que descreve o comportamento do valor em relação ao tempo
- Permite edição das curvas a fim de suavizar/detalhar os movimentos dos personagens (*Splining* e *Polishing*)

# Keyframing em Blender - Graph Editor

- Mostra cada canal animado como uma curva que descreve o comportamento do valor em relação ao tempo
- Permite edição das curvas a fim de suavizar/detalhar os movimentos dos personagens (*Splining* e *Polishing*)



<http://wiki.blender.org/index.php/Doc:2.6/Manual/Animation/Editors/Graph/FCurves>

# Keyframing em Blender - Graph Editor

---

Modos de Interpolação:

# Keyframing em Blender - Graph Editor

---

Modos de Interpolação:

**Constant** nenhum tipo de interpolação. Os atributos permanecem constantes de um quadro-chave para o próximo

# Keyframing em Blender - Graph Editor

---

Modos de Interpolação:

**Constant** nenhum tipo de interpolação. Os atributos permanecem constantes de um quadro-chave para o próximo

**Linear** interpolação linear entre quadros-chave vizinhos

# Keyframing em Blender - Graph Editor

---

Modos de Interpolação:

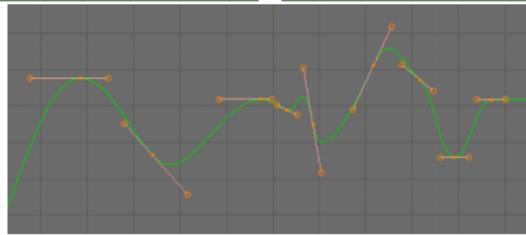
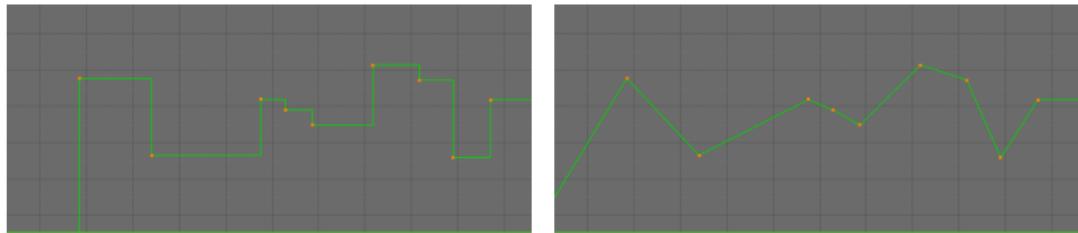
**Constant** nenhum tipo de interpolação. Os atributos permanecem constantes de um quadro-chave para o próximo

**Linear** interpolação linear entre quadros-chave vizinhos

**Bezier** interpolação quadrática por curvas Bézier (suaves) entre agrupamentos de quadros-chaves. Permite recursos como *ease-in ease-out*

**Tecla de Atalho: T**

# Keyframing em Blender - Graph Editor



# Keyframing em Blender - Graph Editor

---

Tipos de Manipuladores:

# Keyframing em Blender - Graph Editor

---

Tipos de Manipuladores:

**Free** pode-se mover cada lado do manipulador de forma independente

# Keyframing em Blender - Graph Editor

---

Tipos de Manipuladores:

**Free** pode-se mover cada lado do manipulador de forma independente

**Vector** cada lado do manipulador aponta para o quadro-chave vizinho (interpolação linear)

# Keyframing em Blender - Graph Editor

---

Tipos de Manipuladores:

**Free** pode-se mover cada lado do manipulador de forma independente

**Vector** cada lado do manipulador aponta para o quadro-chave vizinho (interpolação linear)

**Aligned** o manipulador mantém orientação e posição tangente à curva

# Keyframing em Blender - Graph Editor

---

Tipos de Manipuladores:

**Free** pode-se mover cada lado do manipulador de forma independente

**Vector** cada lado do manipulador aponta para o quadro-chave vizinho (interpolação linear)

**Aligned** o manipulador mantém orientação e posição tangente à curva

**Automatic** os quadros-chave são interpolados automaticamente

# Keyframing em Blender - Graph Editor

---

Tipos de Manipuladores:

**Free** pode-se mover cada lado do manipulador de forma independente

**Vector** cada lado do manipulador aponta para o quadro-chave vizinho (interpolação linear)

**Aligned** o manipulador mantém orientação e posição tangente à curva

**Automatic** os quadros-chave são interpolados automaticamente

**Auto Clamped** o manipulador se alinha à curva de forma que a curvatura não ultrapasse a tangente (*overshooting*)

**Tecla de Atalho: V**

## Keyframing em Blender - Graph Editor

---

Extrapolação significa o critério a ser adotado para os frames posicionados fora dos limites definidos para a timeline, ou seja, aqueles **antes** do frame inicial e **depois** do frame final

# Keyframing em Blender - Graph Editor

---

Extrapolação significa o critério a ser adotado para os frames posicionados fora dos limites definidos para a timeline, ou seja, aqueles **antes** do frame inicial e **depois** do frame final

Modos de Extrapolação:

**Constant** nenhum tipo de extrapolação. Os atributos permanecem constantes

# Keyframing em Blender - Graph Editor

---

Extrapolação significa o critério a ser adotado para os frames posicionados fora dos limites definidos para a timeline, ou seja, aqueles **antes** do frame inicial e **depois** do frame final

Modos de Extrapolação:

**Constant** nenhum tipo de extrapolação. Os atributos permanecem constantes

**Linear** interpolação linear entre quadros-chave extrapolados

# Keyframing em Blender - Graph Editor

---

Extrapolação significa o critério a ser adotado para os frames posicionados fora dos limites definidos para a timeline, ou seja, aqueles **antes** do frame inicial e **depois** do frame final

Modos de Extrapolação:

**Constant** nenhum tipo de extrapolação. Os atributos permanecem constantes

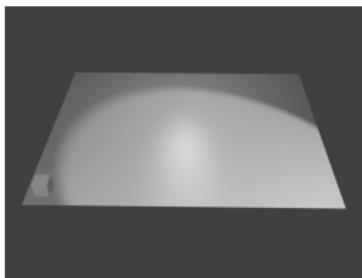
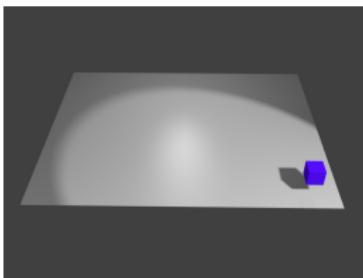
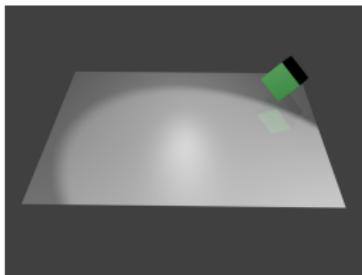
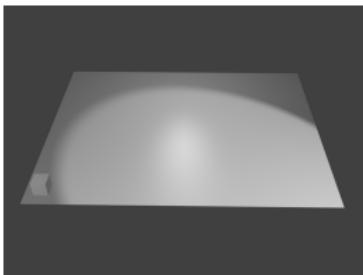
**Linear** interpolação linear entre quadros-chave extrapolados

**Make Cyclic** repete a curva definida na timeline de maneira cíclica

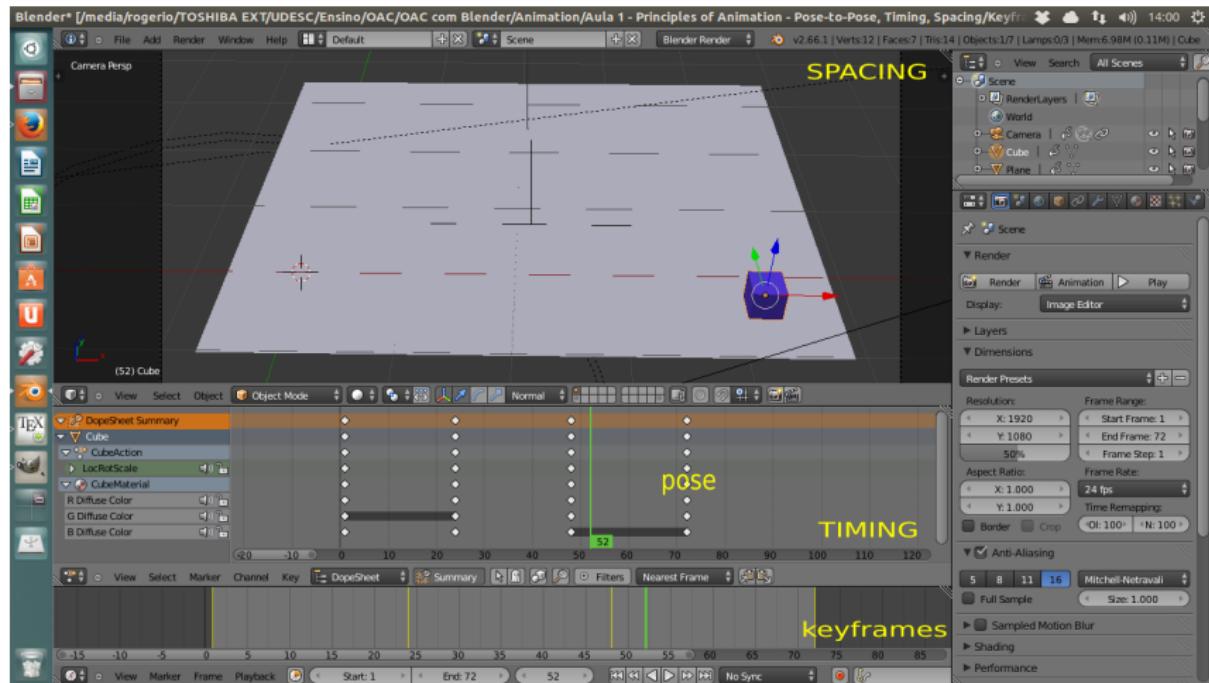
**Menu: Channel | Extrapolation Mode**

# Aula #1 - Timing & Spacing

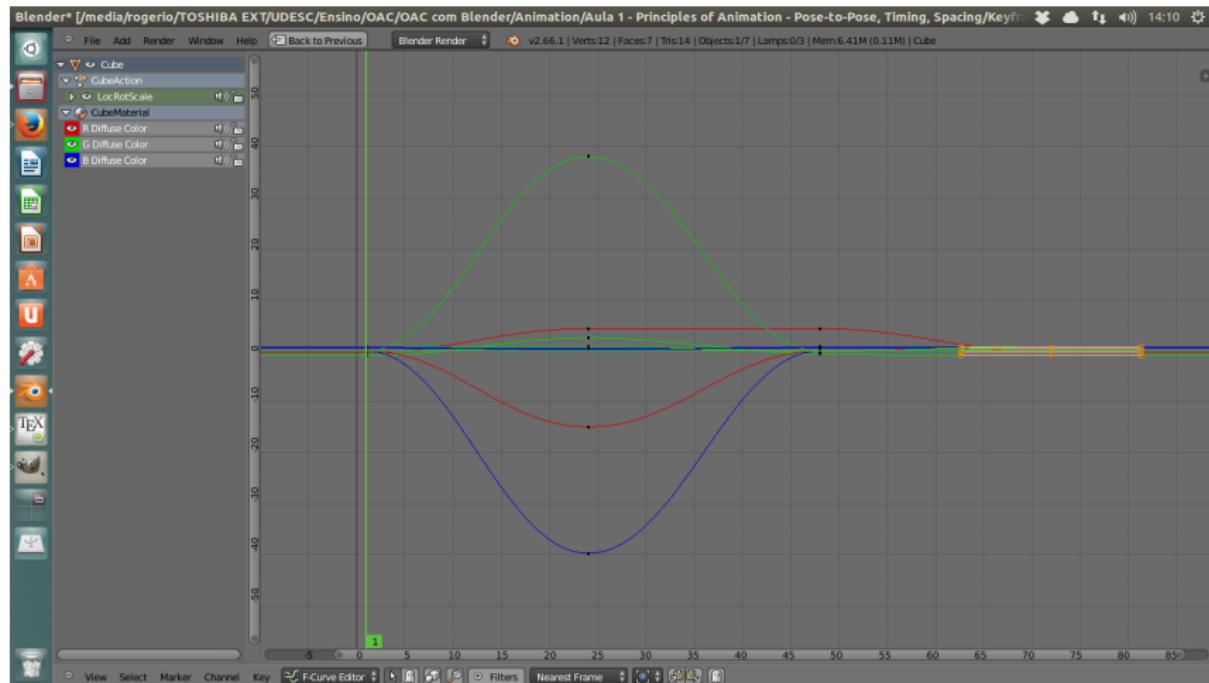
---



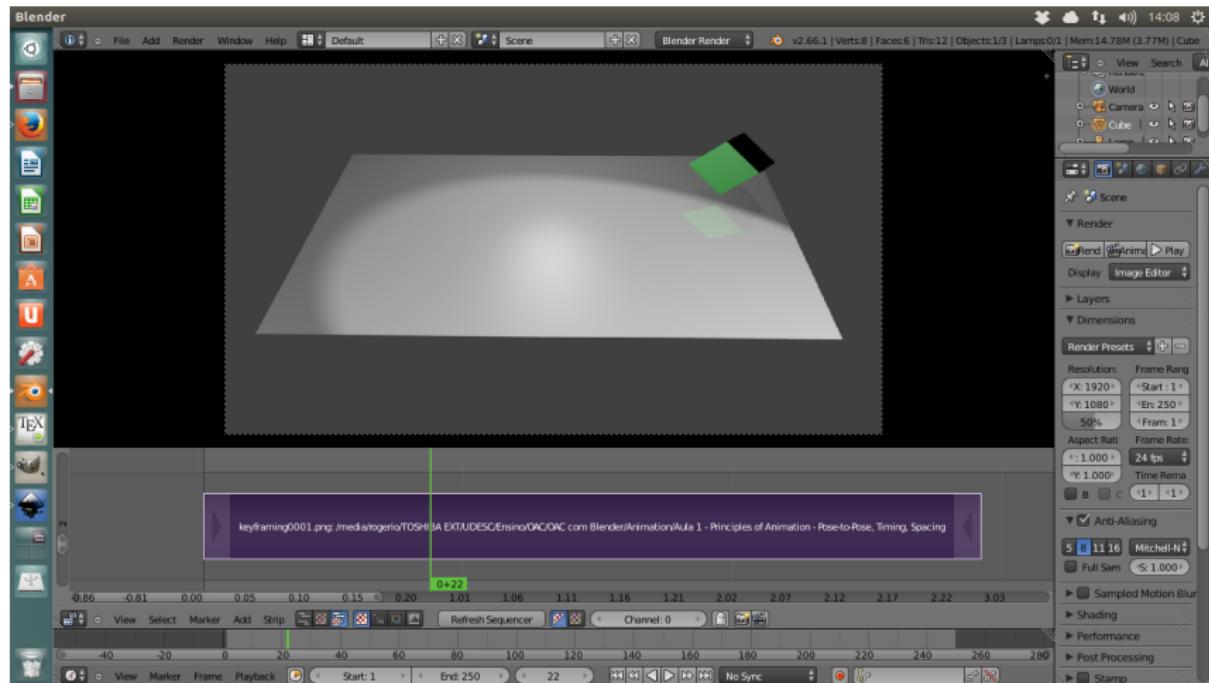
# Aula #1 - Timing & Spacing



# Aula #1 - Timing & Spacing



# Aula #1 - Timing & Spacing



## Aula #2 - Ease In/Out + Squash & Stretch

---

**Script** Definir suscintamente a história a ser animada: “*uma bola de borracha atravessa a tela quicando sobre um piso. Ela demora 5 segundos (aprox.) para percorrer a tela toda. Durante esse tempo ela quica 9 vezes. Ouve-se o som do contato da bola com o chão a cada salto.*”

## Aula #2 - Ease In/Out + Squash & Stretch

---

**Script** Definir suscintamente a história a ser animada: “*uma bola de borracha atravessa a tela quicando sobre um piso. Ela demora 5 segundos (aprox.) para percorrer a tela toda. Durante esse tempo ela quica 9 vezes. Ouve-se o som do contato da bola com o chão a cada salto.*”

**Storyboard** Traçar o projeto da animação através do *grease pencil*

## Aula #2 - Ease In/Out + Squash & Stretch

---

**Script** Definir suscintamente a história a ser animada: “*uma bola de borracha atravessa a tela quicando sobre um piso. Ela demora 5 segundos (aprox.) para percorrer a tela toda. Durante esse tempo ela quica 9 vezes. Ouve-se o som do contato da bola com o chão a cada salto.*”

**Storyboard** Traçar o projeto da animação através do *grease pencil*

**Blocking** Criar os quadros-chave nos pontos extremos da curva definida na fase anterior (*spacing*). Ajustar o tempo em que cada quadro ocorre (*timing*)

## Aula #2 - Ease In/Out + Squash & Stretch

---

**Script** Definir suscintamente a história a ser animada: “*uma bola de borracha atravessa a tela quicando sobre um piso. Ela demora 5 segundos (aprox.) para percorrer a tela toda. Durante esse tempo ela quica 9 vezes. Ouve-se o som do contato da bola com o chão a cada salto.*”

**Storyboard** Traçar o projeto da animação através do *grease pencil*

**Blocking** Criar os quadros-chave nos pontos extremos da curva definida na fase anterior (*spacing*). Ajustar o tempo em que cada quadro ocorre (*timing*)

**Splining** Ajustar as curvas de animação para cada atributo da bola

## Aula #2 - Ease In/Out + Squash & Stretch

---

**Script** Definir suscintamente a história a ser animada: “*uma bola de borracha atravessa a tela quicando sobre um piso. Ela demora 5 segundos (aprox.) para percorrer a tela toda. Durante esse tempo ela quica 9 vezes. Ouve-se o som do contato da bola com o chão a cada salto.*”

**Storyboard** Traçar o projeto da animação através do *grease pencil*

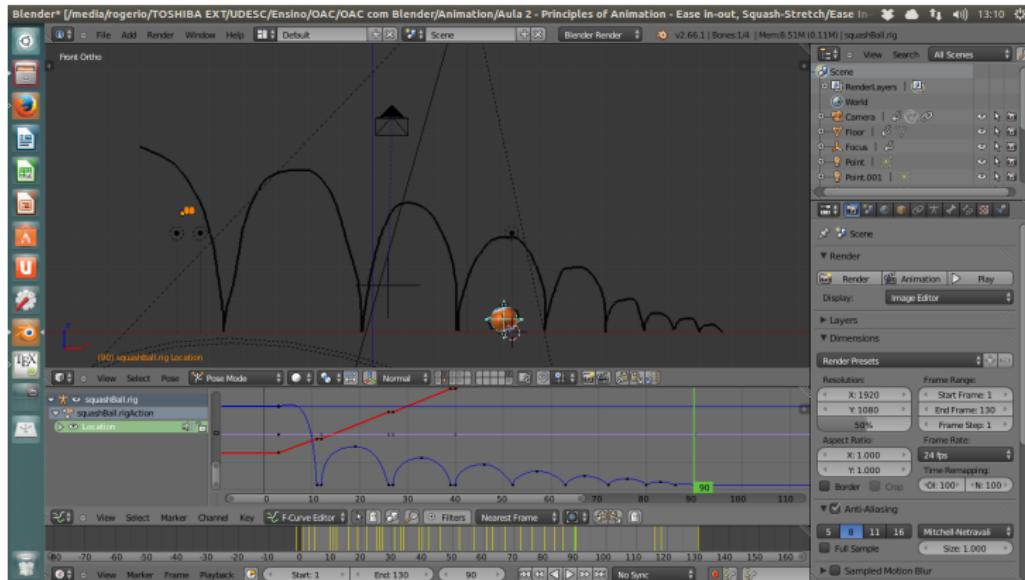
**Blocking** Criar os quadros-chave nos pontos extremos da curva definida na fase anterior (*spacing*). Ajustar o tempo em que cada quadro ocorre (*timing*)

**Splining** Ajustar as curvas de animação para cada atributo da bola

**Polishing** Melhorar as curvas de interpolação a fim de coincidir com os princípios da animação de Disney (*timing, spacing, pose-to-pose, squash-stretch, exaggeration, staging, arcs, solid drawing*)

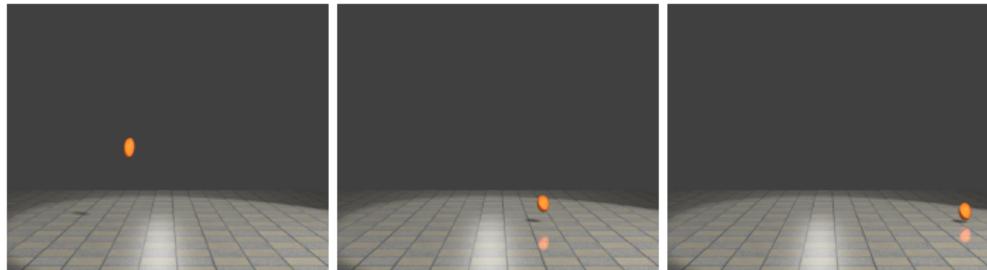
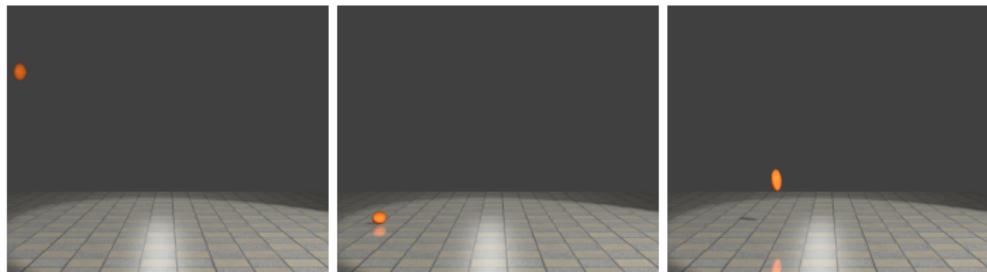
# Aula #2 - Ease In/Out + Squash & Stretch

Criando o traçado para uma animação de 'bola quicando' (*bouncing ball*) com *grease pencil* (D+LMB: risca D+RMB: apaga)

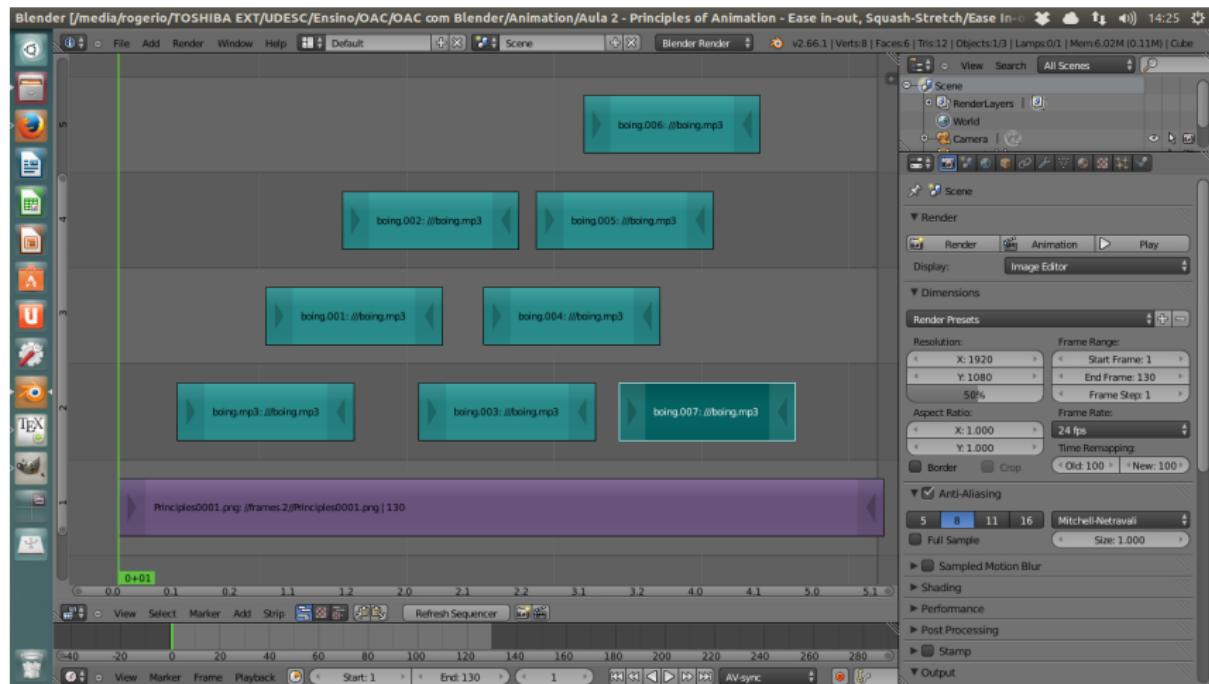


## Aula #2 - Ease In/Out + Squash & Stretch

---



# Aula #2 - Ease In/Out + Squash & Stretch



# Modelos Hierárquicos

- Sistemas de Coordenadas Locais vs. Globais
- Relação Pai-Filho
- Hierarquias típicas & Personagens Articulados
- DKA vs. IKA

# Sistemas de Coordenadas

---

- É definido como um conjunto de distância (em relação a eixos coordenados) à partir de uma origem pré-definida

# Sistemas de Coordenadas

---

- É definido como um conjunto de distância (em relação a eixos coordenados) à partir de uma origem pré-definida
- Caso clássico:  $\mathbb{R}^3$  (coordenadas espaciais), onde três eixos ortogonais definem o espaço tridimensional com base nas distâncias: ordenada, absissa e cota

# Sistemas de Coordenadas

---

- É definido como um conjunto de distância (em relação a eixos coordenados) à partir de uma origem pré-definida
- Caso clássico:  $\mathbb{R}^3$  (coordenadas espaciais), onde três eixos ortogonais definem o espaço tridimensional com base nas distâncias: ordenada, absissa e cota
- Sistemas de coordenadas podem ser hierárquicos, dado que dependem de um único ponto de referência (a origem dos eixos). Tal ponto de origem pode ser representado como um ponto em outro sistema de coordenadas

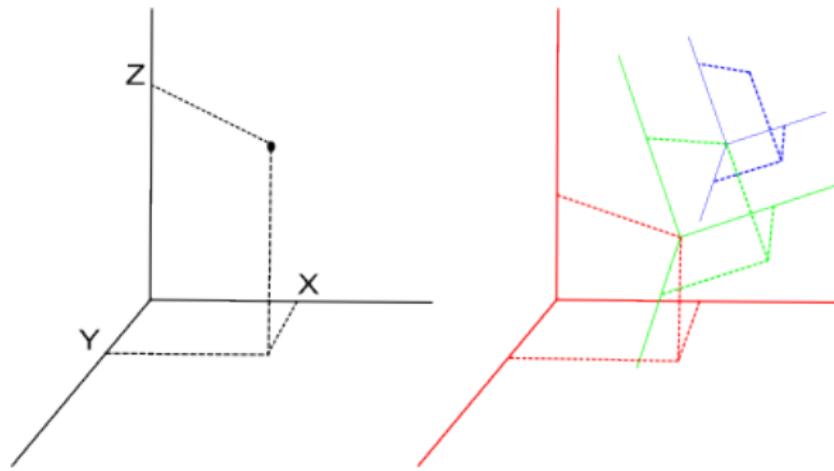
# Sistemas de Coordenadas

---

- É definido como um conjunto de distância (em relação a eixos coordenados) à partir de uma origem pré-definida
- Caso clássico:  $\mathbb{R}^3$  (coordenadas espaciais), onde três eixos ortogonais definem o espaço tridimensional com base nas distâncias: ordenada, absissa e cota
- Sistemas de coordenadas podem ser hierárquicos, dado que dependem de um único ponto de referência (a origem dos eixos). Tal ponto de origem pode ser representado como um ponto em outro sistema de coordenadas
- O sistema de coordenadas que contém outros sistemas é dito ser as coordenadas **globais**, todos os demais sistemas (dependentes deste) são ditos ser representações em coordenadas **locais**

# Sistemas de Coordenadas

---



# Sistemas de Coordenadas

---

- Esta noção de coordenadas globais vs locais permite a modelagem de objetos complexos, compostos de sub-partes relacionadas porém independentes (**relação pai-filho**)

# Sistemas de Coordenadas

---

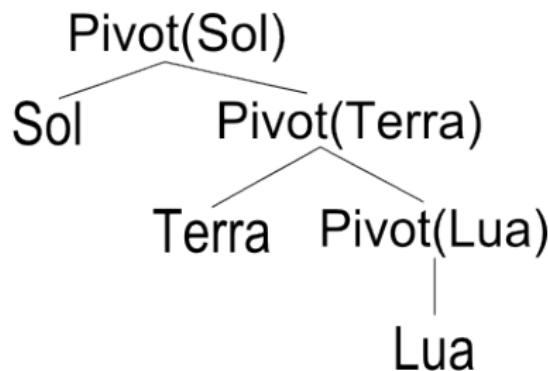
- Esta noção de coordenadas globais vs locais permite a modelagem de objetos complexos, compostos de sub-partes relacionadas porém independentes (**relação pai-filho**)
- Uma estrutura pai-filho (*Parenting*) indica que determinado objeto (filho) é modelado em um sistema de coordenadas local relacionado ao sistema de coordenadas do objeto pai. Desta forma, qualquer transformação geométrica aplicada ao sistema global refletirá da mesma forma sobre o sistema de coordenadas local. O inverso não se aplica

# Sistemas de Coordenadas

---

- Esta noção de coordenadas globais vs locais permite a modelagem de objetos complexos, compostos de sub-partes relacionadas porém independentes (**relação pai-filho**)
- Uma estrutura pai-filho (*Parenting*) indica que determinado objeto (filho) é modelado em um sistema de coordenadas local relacionado ao sistema de coordenadas do objeto pai. Desta forma, qualquer transformação geométrica aplicada ao sistema global refletirá da mesma forma sobre o sistema de coordenadas local. O inverso não se aplica
- Um exemplo de relação pai-filho clássico: sistema solar. Onde planetas apresentam seus próprios movimentos de rotação (giram ao redor de seus eixos), porém transladam (rotacionam) ao redor do sol no centro do sistema. O mesmo pode ser dito das luas em relação aos planetas e do próprio sol em relação ao centro da galáxia

# Sistemas de Coordenadas



# Parenting in Blender

---

Processo de construção de uma relação pai-filho em Blender:

# Parenting in Blender

---

Processo de construção de uma relação pai-filho em Blender:

1. Construa cada objeto (pai e filho) separadamente, posicionando-os corretamente no espaço

# Parenting in Blender

---

Processo de construção de uma relação pai-filho em Blender:

1. Construa cada objeto (pai e filho) separadamente, posicionando-os corretamente no espaço
2. Selecione primeiro o objeto filho, e então **Shift+RMB** o objeto pai

# Parenting in Blender

---

Processo de construção de uma relação pai-filho em Blender:

1. Construa cada objeto (pai e filho) separadamente, posicionando-os corretamente no espaço
2. Selecione primeiro o objeto filho, e então **Shift+RMB** o objeto pai
3. Use as teclas **Alt+P** para criar a relação pai-filho. Duas opções possíveis:

# Parenting in Blender

---

Processo de construção de uma relação pai-filho em Blender:

1. Construa cada objeto (pai e filho) separadamente, posicionando-os corretamente no espaço
2. Selecione primeiro o objeto filho, e então **Shift+RMB** o objeto pai
3. Use as teclas **Alt+P** para criar a relação pai-filho. Duas opções possíveis:  
**Object** relação pai-filho é criada sem levar em consideração as transformações geométricas correntes dos objetos

# Parenting in Blender

---

Processo de construção de uma relação pai-filho em Blender:

1. Construa cada objeto (pai e filho) separadamente, posicionando-os corretamente no espaço
2. Selecione primeiro o objeto filho, e então **Shift+RMB** o objeto pai
3. Use as teclas **Alt+P** para criar a relação pai-filho. Duas opções possíveis:

**Object** relação pai-filho é criada sem levar em consideração as transformações geométricas correntes dos objetos

**Object (Keep Transform)** relação pai-filho é criada considerando-se as transformações geométricas correntes dos objetos

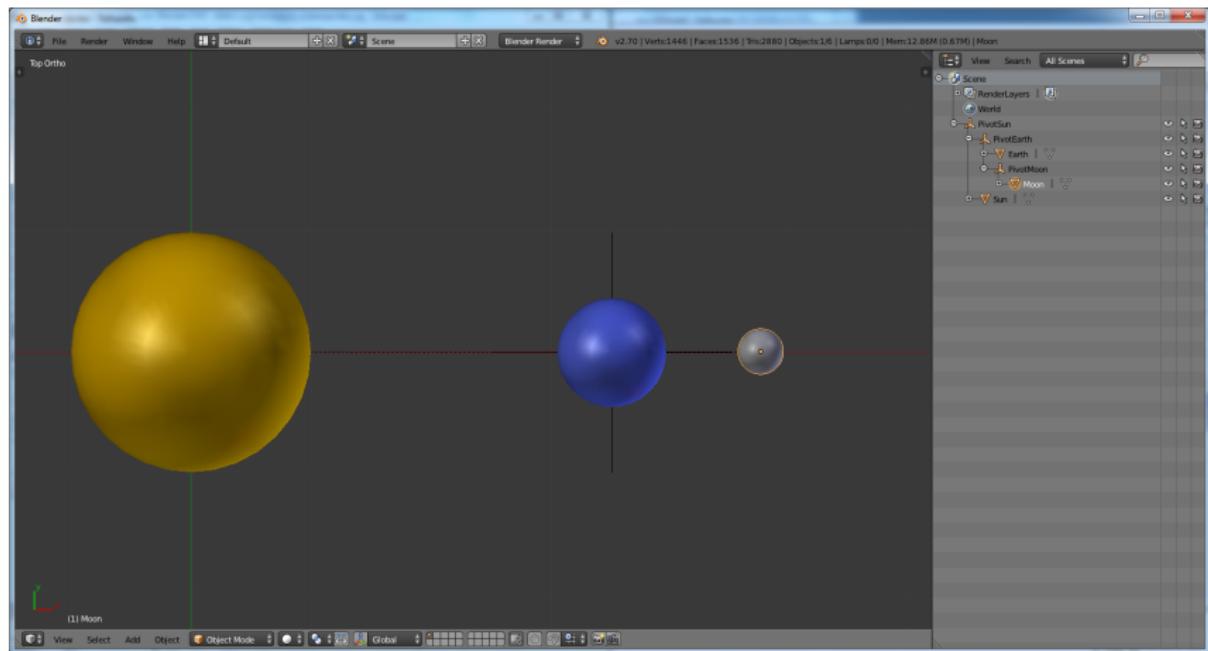
# Parenting in Blender

---

Processo de construção de uma relação pai-filho em Blender:

1. Construa cada objeto (pai e filho) separadamente, posicionando-os corretamente no espaço
2. Selecione primeiro o objeto filho, e então **Shift+RMB** o objeto pai
3. Use as teclas **Alt+P** para criar a relação pai-filho. Duas opções possíveis:
  - Object** relação pai-filho é criada sem levar em consideração as transformações geométricas correntes dos objetos
  - Object (Keep Transform)** relação pai-filho é criada considerando-se as transformações geométricas correntes dos objetos
4. Os objetos estão prontos para serem animados hierarquicamente!

# Parenting in Blender



# Personagens Articulados

---

- Dá-se o nome de *personagem articulado* para os modelos hierárquicos onde a relação pai-filho representa **juntas** (*Joints*) conexas

# Personagens Articulados

---

- Dá-se o nome de *personagem articulado* para os modelos hierárquicos onde a relação pai-filho representa **juntas** (*Joints*) conexas
- Tipos de juntas:  
**Hinge** junta do tipo 'dobradiça' → 1 grau de liberdade (*DOF*). Exemplo:  
cotovelo

# Personagens Articulados

---

- Dá-se o nome de *personagem articulado* para os modelos hierárquicos onde a relação pai-filho representa **juntas** (*Joints*) conexas
- Tipos de juntas:
  - Hinge** junta do tipo 'dobradiça' → 1 grau de liberdade (*DOF*). Exemplo: cotovelo
  - Pivot** junta do tipo 'alavanca' → 2 graus de liberdade. Exemplo: tibia

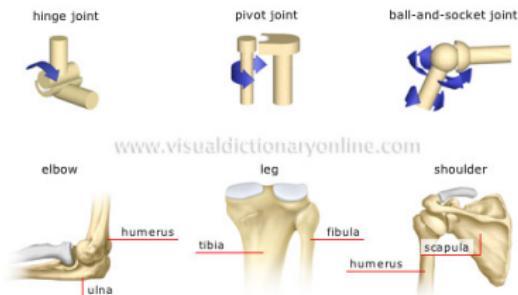
# Personagens Articulados

---

- Dá-se o nome de *personagem articulado* para os modelos hierárquicos onde a relação pai-filho representa **juntas** (*Joints*) conexas
- Tipos de juntas:
  - Hinge** junta do tipo 'dobradiça' → 1 grau de liberdade (*DOF*). Exemplo: cotovelo
  - Pivot** junta do tipo 'alavanca' → 2 graus de liberdade. Exemplo: tíbia
  - Ball-and-Socket** junta do tipo 'soquete' → 3 graus de liberdade. Exemplo: ombro

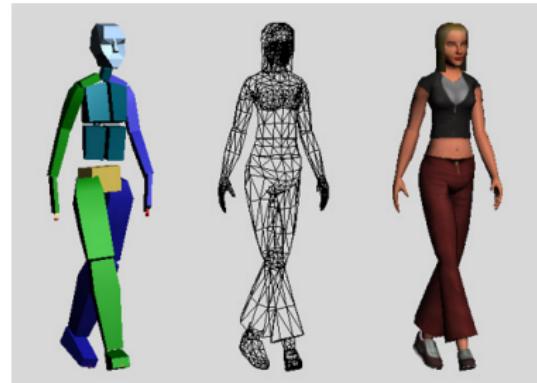
# Personagens Articulados

- Dá-se o nome de *personagem articulado* para os modelos hierárquicos onde a relação pai-filho representa **juntas** (*Joints*) conexas
- Tipos de juntas:
  - Hinge** junta do tipo 'dobradiça' → 1 grau de liberdade (*DOF*). Exemplo: cotovelo
  - Pivot** junta do tipo 'alavanca' → 2 graus de liberdade. Exemplo: tíbia
  - Ball-and-Socket** junta do tipo 'soquete' → 3 graus de liberdade. Exemplo: ombro



# Personagens Articulados

---



## Personagens Articulados - Esqueletos

---

- **Esqueletos** são uma técnica de modelagem de personagem articulados que permitem um controle simplificado de personagens complexos

# Personagens Articulados - Esqueletos

---

- **Esqueletos** são uma técnica de modelagem de personagem articulados que permitem um controle simplificado de personagens complexos
- O processo de construção de um personagem articulado com esqueleto é denominado *rigging*

## Personagens Articulados - Esqueletos

---

- **Esqueletos** são uma técnica de modelagem de personagem articulados que permitem um controle simplificado de personagens complexos
- O processo de construção de um personagem articulado com esqueleto é denominado *rigging*
- A ideia central é construir o modelo em camadas:

# Personagens Articulados - Esqueletos

---

- **Esqueletos** são uma técnica de modelagem de personagem articulados que permitem um controle simplificado de personagens complexos
- O processo de construção de um personagem articulado com esqueleto é denominado *rigging*
- A ideia central é construir o modelo em camadas:  
**Esqueleto** representação hierárquica das juntas do personagem = árvore dos pontos de articulação do modelo

# Personagens Articulados - Esqueletos

---

- **Esqueletos** são uma técnica de modelagem de personagem articulados que permitem um controle simplificado de personagens complexos
- O processo de construção de um personagem articulado com esqueleto é denominado *rigging*
- A ideia central é construir o modelo em camadas:
  - Esqueleto** representação hierárquica das juntas do personagem = árvore dos pontos de articulação do modelo

**Malha Deformável** representa o modelo do personagem em si como uma malha de polígonos, onde os vértices da cada polígono são criados hierarquicamente em relação a um ou mais pontos de articulação do esqueleto, e são alterados em função de um *modelo de deformação*

# Personagens Articulados - Esqueletos

---

- **Esqueletos** são uma técnica de modelagem de personagem articulados que permitem um controle simplificado de personagens complexos
- O processo de construção de um personagem articulado com esqueleto é denominado *rigging*
- A ideia central é construir o modelo em camadas:

**Esqueleto** representação hierárquica das juntas do personagem = árvore dos pontos de articulação do modelo

**Malha Deformável** representa o modelo do personagem em si como uma malha de polígonos, onde os vértices da cada polígono são criados hierarquicamente em relação a um ou mais pontos de articulação do esqueleto, e são alterados em função de um *modelo de deformação*

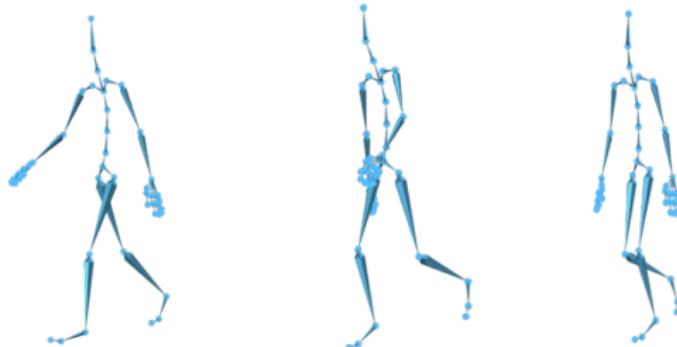
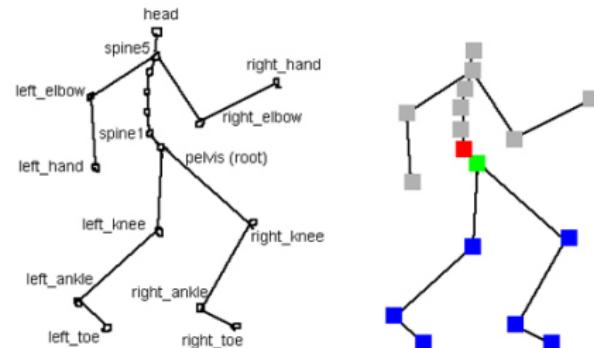
**Pele** aplica-se sobre a malha poligonal deformável um mapa de textura UV (*skinning*) para representar a aparência física do personagem

# Personagens Articulados Deformáveis

---

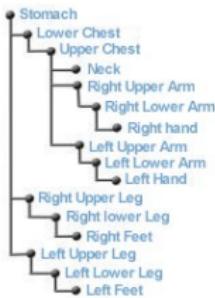


# Animando Personagens Articulados com Pose-to-Pose



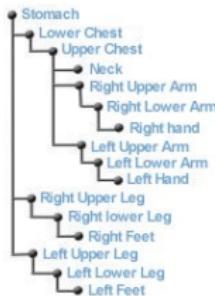
# Cinemática Direta vs Inversa

- O ‘esqueleto’ de um personagem articulado é, na verdade, uma árvore



# Cinemática Direta vs Inversa

- O ‘esqueleto’ de um personagem articulado é, na verdade, uma árvore



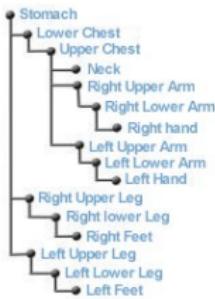
- A disciplina de Estruturas de Dados nos ensina que há duas formas de se construir uma árvore:

Top down no sentido raiz → folhas

Bottom up no sentido folhas → raiz

# Cinemática Direta vs Inversa

- O ‘esqueleto’ de um personagem articulado é, na verdade, uma árvore



- A disciplina de Estruturas de Dados nos ensina que há duas formas de se construir uma árvore:
  - Top down no sentido raiz → folhas
  - Bottom up no sentido folhas → raiz
- O processo de animação esquelética também segue essa convenção...

# Cinemática Direta vs Inversa

---

**Cinemática Direta** (ou *Direct Kinematics Animation - DKA*) é o processo de se definir uma pose, primeiramente, para o elemento na raiz da árvore, e então, prosseguir sequencialmente nível a nível até posicionarmos os elementos-folha chamados de **atuadores**

# Cinemática Direta vs Inversa

---

**Cinemática Direta** (ou *Direct Kinematics Animation - DKA*) é o processo de se definir uma pose, primeiramente, para o elemento na raiz da árvore, e então, prosseguir sequencialmente nível a nível até posicionarmos os elementos-folha chamados de **atuadores**

A DKA permite um posicionamento mais preciso do modelo dado que o animador especifica manualmente cada nó, porém é mais complicado de se atingir uma pose específica (*Reaching problem*)

# Cinemática Direta vs Inversa

---

**Cinemática Direta** (ou *Direct Kinematics Animation - DKA*) é o processo de se definir uma pose, primeiramente, para o elemento na raiz da árvore, e então, prosseguir sequencialmente nível a nível até posicionarmos os elementos-folha chamados de **atuadores**

A DKA permite um posicionamento mais preciso do modelo dado que o animador especifica manualmente cada nó, porém é mais complicado de se atingir uma pose específica (*Reaching problem*)

**Cinemática Inversa** (ou *Inverse Kinematics Animation - IKA*) é o processo inverso: primeiramente se posicionam os atuadores da árvore (extremidades), e então por extração matemática, se definem as configurações para os demais elementos internos da árvore

# Cinemática Direta vs Inversa

---

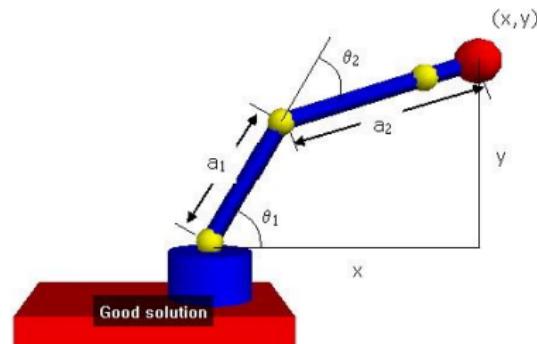
**Cinemática Direta** (ou *Direct Kinematics Animation - DKA*) é o processo de se definir uma pose, primeiramente, para o elemento na raiz da árvore, e então, prosseguir sequencialmente nível a nível até posicionarmos os elementos-folha chamados de **atuadores**

A DKA permite um posicionamento mais preciso do modelo dado que o animador especifica manualmente cada nó, porém é mais complicado de se atingir uma pose específica (*Reaching problem*)

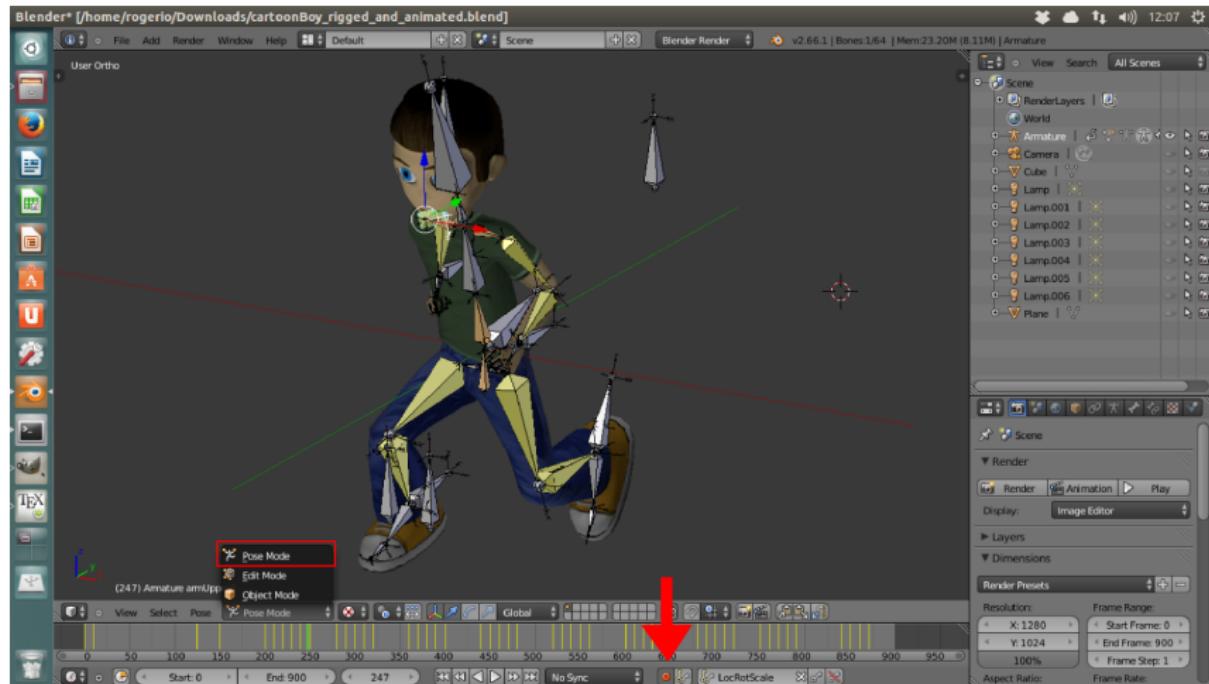
**Cinemática Inversa** (ou *Inverse Kinematics Animation - IKA*) é o processo inverso: primeiramente se posicionam os atuadores da árvore (extremidades), e então por extração matemática, se definem as configurações para os demais elementos internos da árvore

A IKA resolve o problema do *reaching problem*, dado que o atuador é o primeiro elemento a ser animado; porém, dado que os demais nós são gerados por extração, pode-se obter poses indesejadas, além disso, a etapa de *splining* também fica comprometida, pois não há garantias de que as curvas de animação produzidas sejam as esperadas

# Cinemática Direta vs Inversa



# Animando Personagens Articulados com Pose-to-Pose



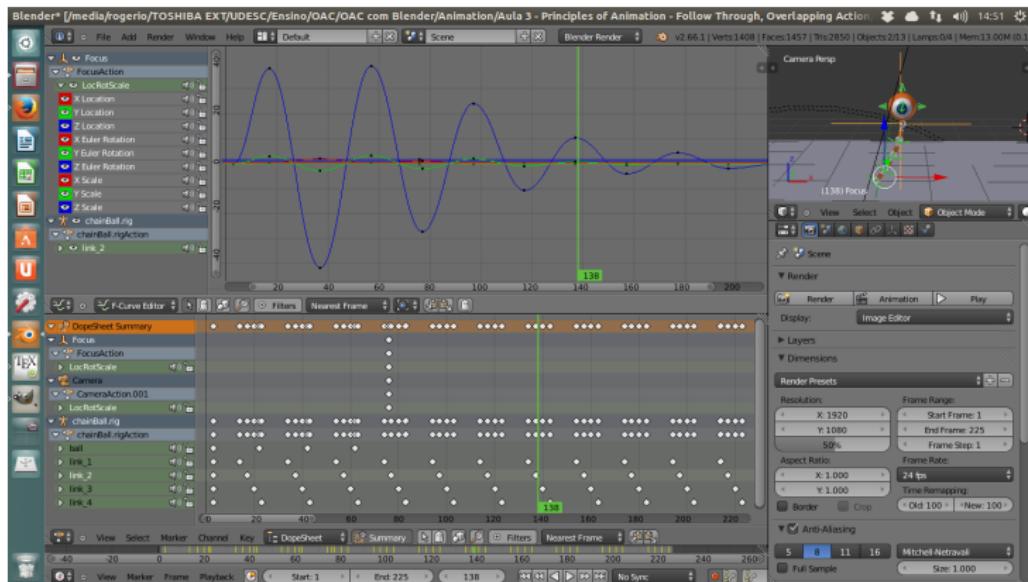
# Animando Personagens Articulados com Pose-to-Pose

---



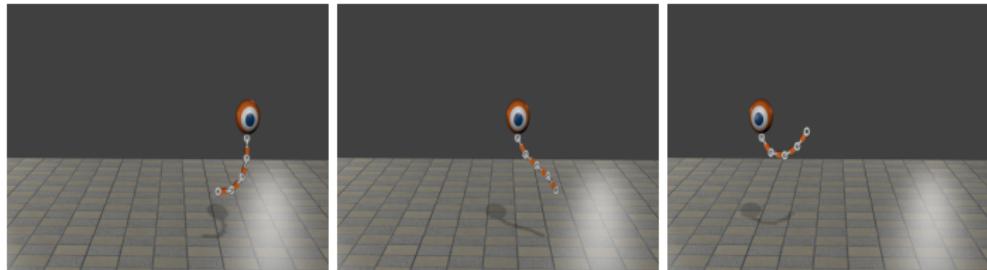
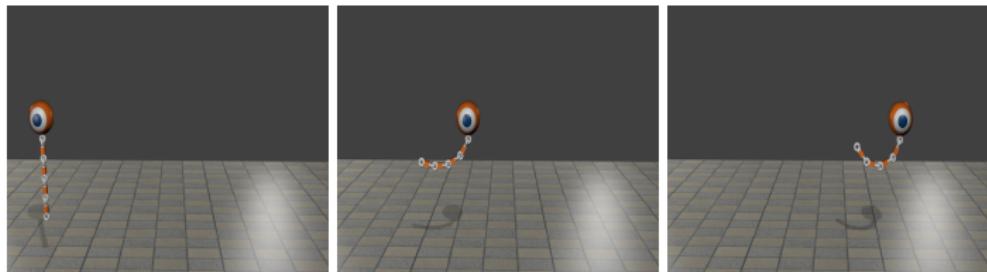
# Aula #3 - Follow Through

Criando uma animação de 'bola com corrente' para produzir o efeito de *follow through*:



# Aula #3 - Follow Through

---



# Animação de Personagens

---

- Animação de personagens (bípedes, quadrúpedes, etc.) segue processo análogo: pose-to-pose

# Animação de Personagens

---

- Animação de personagens (bípedes, quadrúpedes, etc.) segue processo análogo: pose-to-pose
- Abordagem:

# Animação de Personagens

---

- Animação de personagens (bípedes, quadrúpedes, etc.) segue processo análogo: pose-to-pose
- Abordagem:
  1. Animação de ações primárias (movimentos cíclicos)

# Animação de Personagens

---

- Animação de personagens (bípedes, quadrúpedes, etc.) segue processo análogo: pose-to-pose
- Abordagem:
  1. Animação de ações primárias (movimentos cíclicos)
  2. Definição de ações compostas em uma animação não-linear (*Non-Linear Animation - NLA*)

# Animação de Personagens

---

- Animação de personagens (bípedes, quadrúpedes, etc.) segue processo análogo: pose-to-pose
- Abordagem:
  1. Animação de ações primárias (movimentos cíclicos)
  2. Definição de ações compostas em uma animação não-linear (*Non-Linear Animation - NLA*)
  3. Composição (*Compositing*) de efeitos visuais e especiais (VFX)

## Animação de Ações Cíclicas

---

- Ações cíclicas são aquelas que se auto-repetem após algum tempo

# Animação de Ações Cíclicas

---

- Ações cíclicas são aquelas que se auto-repetem após algum tempo
- A construção consiste em se garantir que o primeiro e o último slides do ciclo sejam iguais

# Animação de Ações Cíclicas

---

- Ações cíclicas são aquelas que se auto-repetem após algum tempo
- A construção consiste em se garantir que o primeiro e o último slides do ciclo sejam iguais
- Técnica principalmente utilizada na animação de personagens para jogos e ferramentas de animação interativas

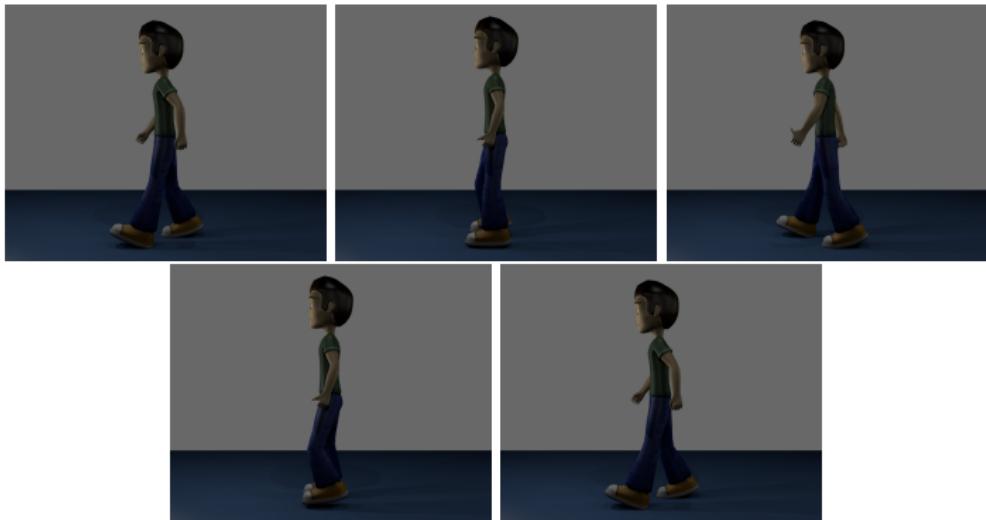
# Animação de Ações Cíclicas

---

- Ações cíclicas são aquelas que se auto-repetem após algum tempo
- A construção consiste em se garantir que o primeiro e o último slides do ciclo sejam iguais
- Técnica principalmente utilizada na animação de personagens para jogos e ferramentas de animação interativas
- Exemplos de ações cíclicas: andar, correr, saltar, se abaixar, aguardar, etc.

# Animação de Ações Cíclicas - Andar

---



# Animação de Ações Cíclicas - Andar

Exemplo de ciclo de caminhada (*Walk cycle*) com 32 frames:

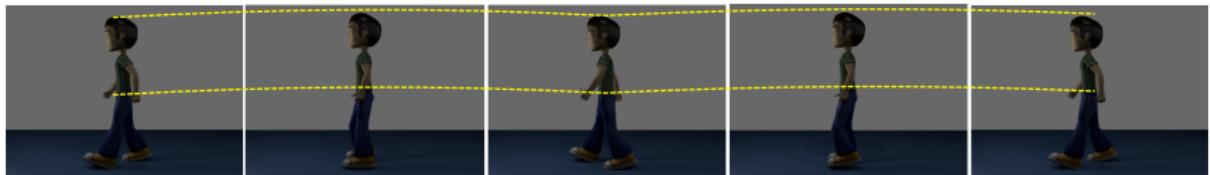
**Frame #1** o inicio do passo começa com os dois pés em contato com o solo, pernas e braços levemente alongados, cabeça levemente inclinada olhando para o local a ser pisado

**Frame #8** a passagem da perna direita representa o momento que a perna esquerda está plenamente ereta (assim como a coluna e a cabeça) para que a perna direita possa ser erguida do solo e efetue a passagem para ser posicionada à frente do corpo para o próximo passo

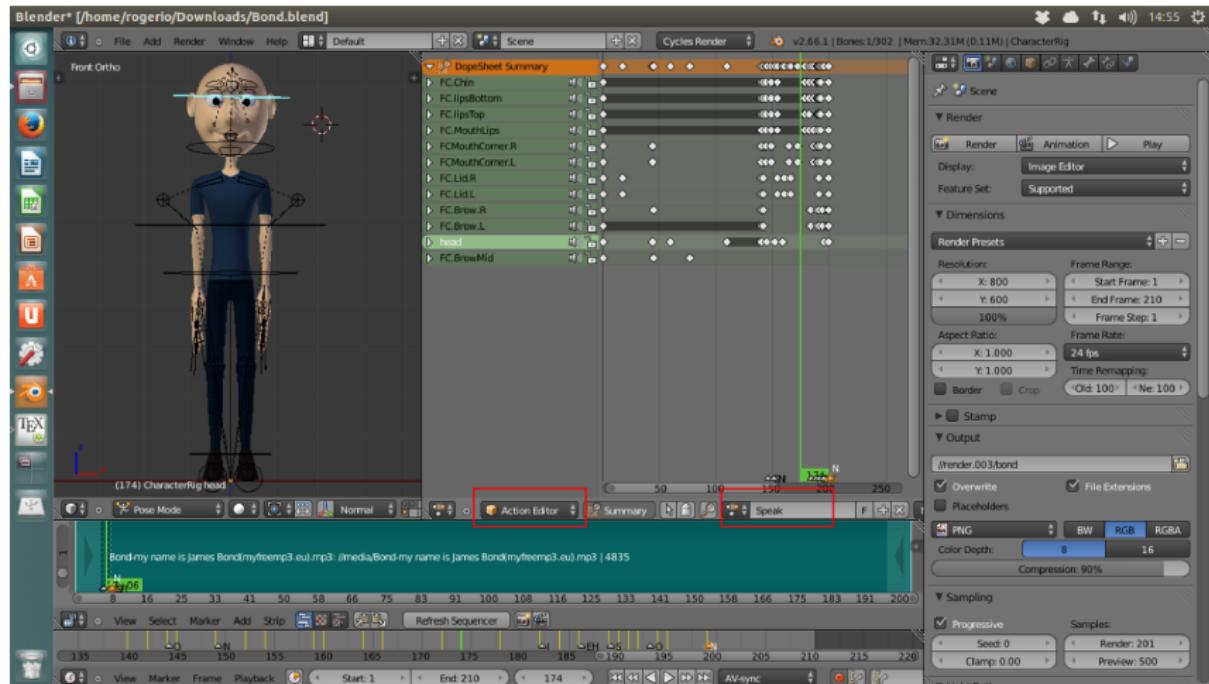
**Frame #16** o contato invertido; é a pose #1 com as pernas e braços em posição trocadas (quem estava à frente agora está atrás e vice-versa)

**Frame #24** segunda passagem, movimento análogo à passagem da perna direita, porém agora realizado com a perna esquerda

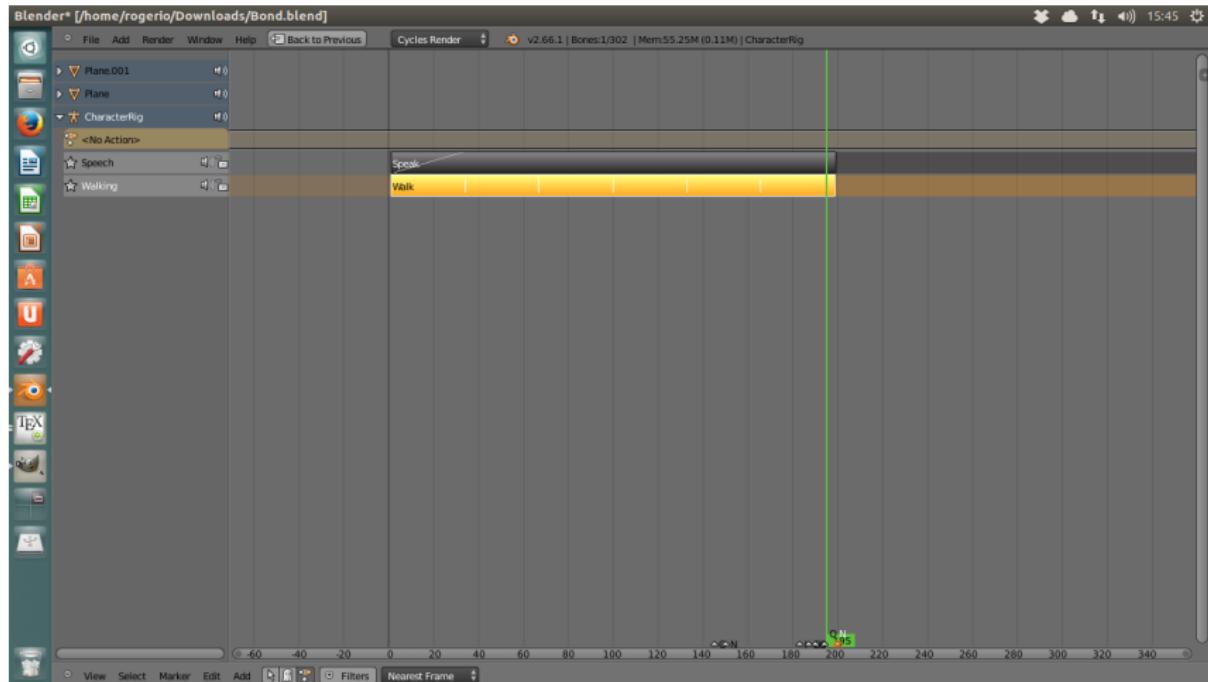
**Frame #32** repetição do frame #1 e o movimento se completa



# Animação de Ações Cíclicas - Action Editor



# Animação de Ações Cíclicas - NLA Editor



# Exercício Prático Individual - EPI #2

---

**Nome:** *Pose-to-Pose Animation*

**Objetivo:** fixação dos fundamentos de animação por quadros-chave baseados em poses; trabalhar com modelos hierárquicos e personagens articulados; construir uma animação não-linear em múltiplas trilhas.

- Enunciado:**
1. escolher/confeccionar um roteiro a ser animado (estória)
  2. selecionar em repositórios online, os modelos/personagens necessários *opensource*
  3. produzir uma sequência de quadros através de *keyframing* com base na estrutura *blocking – splining – polishing*
  4. compilar um filme resultante

- Requisitos:**
- Estória deve conter, no mínimo, dois personagens, bem como outros modelos convencionais
  - Devem ser utilizadas as técnicas vistas em aula: *keyframing*, modelos hierárquicos e personagens articulados
  - Sugestão: animar elementos como luzes e câmeras
  - Não é obrigatório conter diálogos
  - Deve conter trilha sonora e/ou efeitos sonoros
  - Duração mínima: 30 segundos
  - Deve apresentar tela de créditos iniciais e finais (o tempo de apresentação dos créditos não contabiliza para o tempo da animação)

**Prazo para Entrega:** 2 semanas (filme + estória + slides ilustrando o processo)

**Peso:** 15% da média semestral

# Atores Digitais

- Modelos Inspirados na Dramaturgia
- Postura Corporal
- Animação Facial - FACS
- Sincronização Labial (Lip Sync)
- Atores Digitais Autônomos - ADA

# Atores Digitais

---

- São referidos na literatura por diversos nomes: personagens sintéticos, *Synthespians*, humanos virtuais ou personagens credíveis

# Atores Digitais

---

- São referidos na literatura por diversos nomes: personagens sintéticos, *Synthespians*, humanos virtuais ou personagens credíveis
- Atores Digitais são personagens animados manualmente que existem apenas no mundo virtual

# Atores Digitais

---

- São referidos na literatura por diversos nomes: personagens sintéticos, *Synthespians*, humanos virtuais ou personagens credíveis
- Atores Digitais são personagens animados manualmente que existem apenas no mundo virtual
- Uma variante mais recente desses atores define que eles são representações animadas de atuações de atores reais (capturadas através de dispositivos especiais), transpostas para personagens animados, de maneira que estes se pareçam se comportar como nós (humanos)

# Atores Digitais

---



## Buscando Inspiração no Teatro & Cinema

---

- Estudantes de dramaturgia aprendem que existem duas correntes distintas a escolher durante o processo de atuação cênica: teatro vs. filmes

# Buscando Inspiração no Teatro & Cinema

---

- Estudantes de dramaturgia aprendem que existem duas correntes distintas a escolher durante o processo de atuação cênica: teatro vs. filmes
- No teatro, um ator desenvolve uma encenação em 'tempo real', diante de uma audiência, enquanto que nos filmes, o ator atua diante de uma câmera que está registrando as ações, para apenas posteriormente, serem exibidas para a audiência

# Buscando Inspiração no Teatro & Cinema

---

- Estudantes de dramaturgia aprendem que existem duas correntes distintas a escolher durante o processo de atuação cênica: teatro vs. filmes
- No teatro, um ator desenvolve uma encenação em 'tempo real', diante de uma audiência, enquanto que nos filmes, o ator atua diante de uma câmera que está registrando as ações, para apenas posteriormente, serem exibidas para a audiência
- Essa distinção produz uma enorme diferença no processo de atuação

# Buscando Inspiração no Teatro & Cinema

---

- Estudantes de dramaturgia aprendem que existem duas correntes distintas a escolher durante o processo de atuação cênica: teatro vs. filmes
- No teatro, um ator desenvolve uma encenação em 'tempo real', diante de uma audiência, enquanto que nos filmes, o ator atua diante de uma câmera que está registrando as ações, para apenas posteriormente, serem exibidas para a audiência
- Essa distinção produz uma enorme diferença no processo de atuação
- Atores de teatro devem exagerar seus movimentos a fim de enfatizar aspectos importantes da atuação (para que a audiência compreenda). Tal preocupação se torna secundária para atores cinematográficos, dado que essas decisões são delegadas ao diretor que posiciona as câmeras de acordo

# Buscando Inspiração no Teatro & Cinema

---

- Estudantes de dramaturgia aprendem que existem duas correntes distintas a escolher durante o processo de atuação cênica: teatro vs. filmes
- No teatro, um ator desenvolve uma encenação em 'tempo real', diante de uma audiência, enquanto que nos filmes, o ator atua diante de uma câmera que está registrando as ações, para apenas posteriormente, serem exibidas para a audiência
- Essa distinção produz uma enorme diferença no processo de atuação
- Atores de teatro devem exagerar seus movimentos a fim de enfatizar aspectos importantes da atuação (para que a audiência compreenda). Tal preocupação se torna secundária para atores cinematográficos, dado que essas decisões são delegadas ao diretor que posiciona as câmeras de acordo
- Apesar de parecer que nas animações, a abordagem cinematográfica seja a mais apropriada, na verdade é o teatro que produz melhores resultados

# Buscando Inspiração no Teatro & Cinema

---

- Estudantes de dramaturgia aprendem que existem duas correntes distintas a escolher durante o processo de atuação cênica: teatro vs. filmes
- No teatro, um ator desenvolve uma encenação em 'tempo real', diante de uma audiência, enquanto que nos filmes, o ator atua diante de uma câmera que está registrando as ações, para apenas posteriormente, serem exibidas para a audiência
- Essa distinção produz uma enorme diferença no processo de atuação
- Atores de teatro devem exagerar seus movimentos a fim de enfatizar aspectos importantes da atuação (para que a audiência compreenda). Tal preocupação se torna secundária para atores cinematográficos, dado que essas decisões são delegadas ao diretor que posiciona as câmeras de acordo
- Apesar de parecer que nas animações, a abordagem cinematográfica seja a mais apropriada, na verdade é o teatro que produz melhores resultados
- Existem duas abordagens possíveis no ensino de atuação cênica para teatro:  
*Method Acting vs. Theatrical Acting*

# Buscando Inspiração no Teatro & Cinema

---

**Method Acting** também chamado de atuação de ‘dentro para fora’; a ideia é que nosso comportamento é o resultado de nossos estados emocionais e experiências. Nesse tipo de atuação pergunta-se “qual a minha motivação para a cena?”

Em animação de personagens, esta abordagem é utilizada para se determinar como a personagem deve se comportar em cada situação em particular

# Buscando Inspiração no Teatro & Cinema

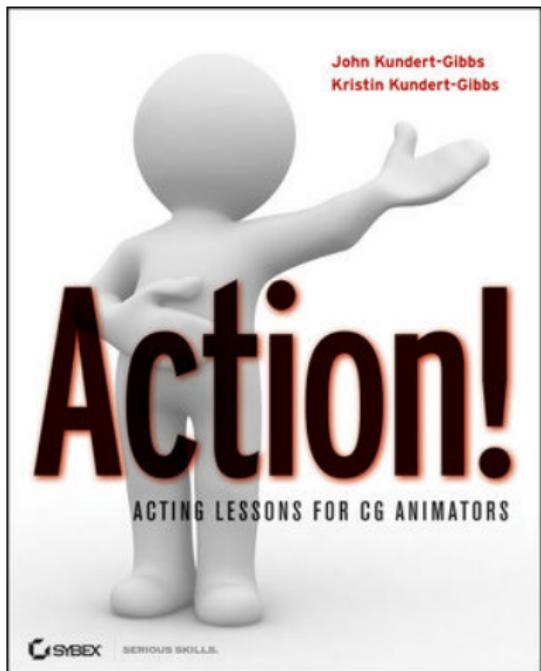
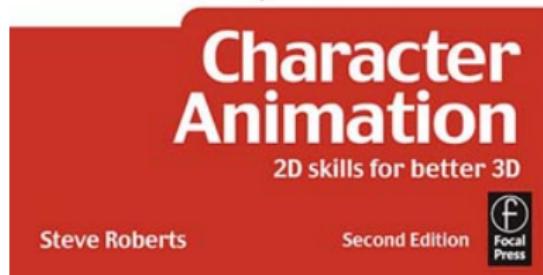
---

**Method Acting** também chamado de atuação de ‘dentro para fora’; a ideia é que nosso comportamento é o resultado de nossos estados emocionais e experiências. Nesse tipo de atuação pergunta-se “qual a minha motivação para a cena?”

Em animação de personagens, esta abordagem é utilizada para se determinar como a personagem deve se comportar em cada situação em particular

**Theatrical Acting** é praticamente a abordagem inversa (‘de fora para dentro’), onde o ator preocupa-se apenas com o roteiro sendo encenado (o qual define as ações a serem atuadas), restando ao ator apenas definir como demonstrar tais ações. Nesse tipo de atuação pergunta-se “o que eu devo atuar?”

# Buscando Inspiração no Teatro & Cinema



# Atores Digitais

---

Na vida real, nossos comportamento (postura corporal, expressões faciais e entonação de voz) são resultados de consequências de emoções sendo sentidas internamente:

# Atores Digitais

---

Na vida real, nossos comportamento (postura corporal, expressões faciais e entonação de voz) são resultados de consequências de emoções sendo sentidas internamente:

**Postura Corporal** utilização do corpo do personagem para demonstrar intenção, fisiologia (p.ex.: cansaço), etc.

# Atores Digitais

---

Na vida real, nossos comportamento (postura corporal, expressões faciais e entonação de voz) são resultados de consequências de emoções sendo sentidas internamente:

**Postura Corporal** utilização do corpo do personagem para demonstrar intenção, fisiologia (p.ex.: cansaço), etc.

**Expressão Facial** utilização da face para demonstrar emoções

# Atores Digitais

---

Na vida real, nossos comportamento (postura corporal, expressões faciais e entonação de voz) são resultados de consequências de emoções sendo sentidas internamente:

**Postura Corporal** utilização do corpo do personagem para demonstrar intenção, fisiologia (p.ex.: cansaço), etc.

**Expressão Facial** utilização da face para demonstrar emoções

**Sincronização Labial** utilização dos lábios para expressar diálogos e emoções

## Atores Digitais - Postura Corporal

---

- O posicionamento das diversas partes do corpo: cabeça, pescoço, espinha dorsal, braços, pernas, etc, refletem os estados mentais de cada pessoa

## Atores Digitais - Postura Corporal

---

- O posicionamento das diversas partes do corpo: cabeça, pescoço, espinha dorsal, braços, pernas, etc, refletem os estados mentais de cada pessoa
- É possível que essas posturas se complementem ou se contradigam a fim de melhor refletir o estado emocional do ator

## Atores Digitais - Postura Corporal

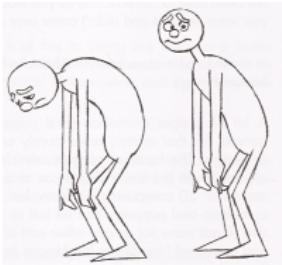
---

- O posicionamento das diversas partes do corpo: cabeça, pescoço, espinha dorsal, braços, pernas, etc, refletem os estados mentais de cada pessoa
- É possível que essas posturas se complementem ou se contradigam a fim de melhor refletir o estado emocional do ator
- Por exemplo: imagine uma personagem severamente deprimida = o corpo inteiro lhe parece 'pesado' e portanto curva-se para baixo. Agora imagine que a pessoa deseja tentar esconder esse fato (depressão reprimida) = neste caso, a pessoa tenta assumir uma posição ereta e despende muito esforço para tal

# Atores Digitais - Postura Corporal

---

- O posicionamento das diversas partes do corpo: cabeça, pescoço, espinha dorsal, braços, pernas, etc, refletem os estados mentais de cada pessoa
- É possível que essas posturas se complementem ou se contradigam a fim de melhor refletir o estado emocional do ator
- Por exemplo: imagine uma personagem severamente deprimida = o corpo inteiro lhe parece 'pesado' e portanto curva-se para baixo. Agora imagine que a pessoa deseja tentar esconder esse fato (depressão reprimida) = neste caso, a pessoa tenta assumir uma posição ereta e despende muito esforço para tal



# Atores Digitais - Postura Corporal

---

- Existem 4 formas básicas de postura corporal:

# Atores Digitais - Postura Corporal

---

- Existem 4 formas básicas de postura corporal:

**Aberta** braços afastados do corpo, mãos com as palmas abertas, pernas afastadas com os pés firmemente plantados contra o solo; e geralmente com o corpo orientado na direção do objeto de interesse

# Atores Digitais - Postura Corporal

---

- Existem 4 formas básicas de postura corporal:

**Aberta** braços afastados do corpo, mãos com as palmas abertas, pernas afastadas com os pés firmemente plantados contra o solo; e geralmente com o corpo orientado na direção do objeto de interesse

**Fechada** braços dobrados próximos ao corpo, pernas cruzadas (no caso do personagem estar sentado), corpo orientado em oposição ao objeto de interesse, cabeça baixa

# Atores Digitais - Postura Corporal

---

- Existem 4 formas básicas de postura corporal:

**Aberta** braços afastados do corpo, mãos com as palmas abertas, pernas afastadas com os pés firmemente plantados contra o solo; e geralmente com o corpo orientado na direção do objeto de interesse

**Fechada** braços dobrados próximos ao corpo, pernas cruzadas (no caso do personagem estar sentado), corpo orientado em oposição ao objeto de interesse, cabeça baixa

**Inclinado para Frente** indica que a personagem está ativamente receptivo/negativo para com a mensagem sendo recebida

# Atores Digitais - Postura Corporal

---

- Existem 4 formas básicas de postura corporal:

**Aberta** braços afastados do corpo, mãos com as palmas abertas, pernas afastadas com os pés firmemente plantados contra o solo; e geralmente com o corpo orientado na direção do objeto de interesse

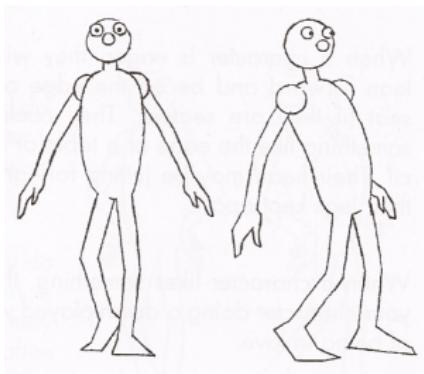
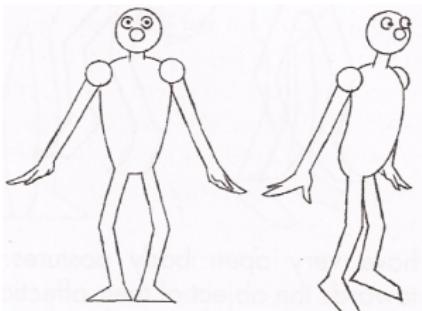
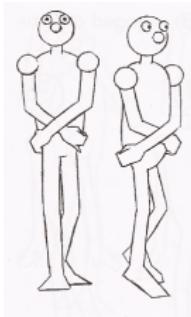
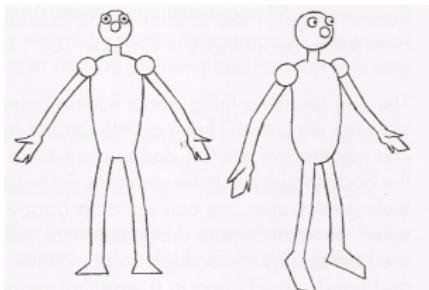
**Fechada** braços dobrados próximos ao corpo, pernas cruzadas (no caso do personagem estar sentado), corpo orientado em oposição ao objeto de interesse, cabeça baixa

**Inclinado para Frente** indica que a personagem está ativamente receptivo/negativo para com a mensagem sendo recebida

**Inclinado para Trás** indica passividade, que a personagem está ignorando a mensagem sendo recebida

# Atores Digitais - Postura Corporal

---



## Atores Digitais - Postura Corporal - Responsivo

---

- Combinação das posturas **aberta** e **inclinado para frente**

## Atores Digitais - Postura Corporal - Responsivo

---

- Combinação das posturas **aberta** e **inclinado para frente**
- Indica pessoas felizes, interessadas, envolvidas e ocupadas com alguma coisa, apaixonadas, desejando algo, ansiosas e gostando de algo

## Atores Digitais - Postura Corporal - Reflexivo

---

- Combinação das posturas **aberta** e **inclinado para trás**

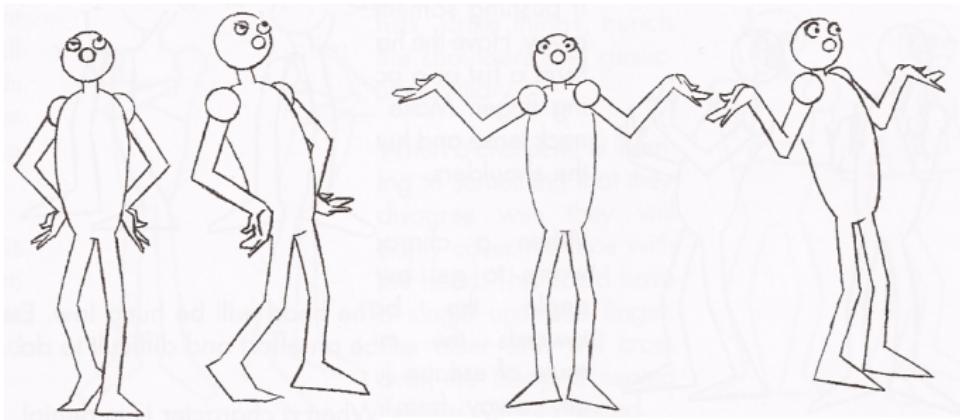
## Atores Digitais - Postura Corporal - Reflexivo

---

- Combinação das posturas **aberta** e **inclinado para trás**
- Indica pessoas contemplativas, pensativas, avaliativas, quando estão perplexas

## Atores Digitais - Postura Corporal - Reflexivo

- Combinação das posturas **aberta** e **inclinado para trás**
- Indica pessoas contemplativas, pensativas, avaliativas, quando estão perplexas



## Atores Digitais - Postura Corporal - Fugitivo

---

- Combinação das posturas **fechada** e **inclinado para trás**

## Atores Digitais - Postura Corporal - Fugitivo

---

- Combinação das posturas **fechada** e **inclinado para trás**
- Indica pessoas se sentindo: rejeitadas, entediadas, tristes, miseráveis, em negação, em dúvida sobre si próprias, mentindo, resistentes a ideias

# Atores Digitais - Postura Corporal - Fugitivo

- Combinação das posturas **fechada** e **inclinado para trás**
- Indica pessoas se sentindo: rejeitadas, entediadas, tristes, miseráveis, em negação, em dúvida sobre si próprias, mentindo, resistentes a ideias



## Atores Digitais - Postura Corporal - Combativo

---

- Combinação das posturas **fechada** e **inclinado para frente**

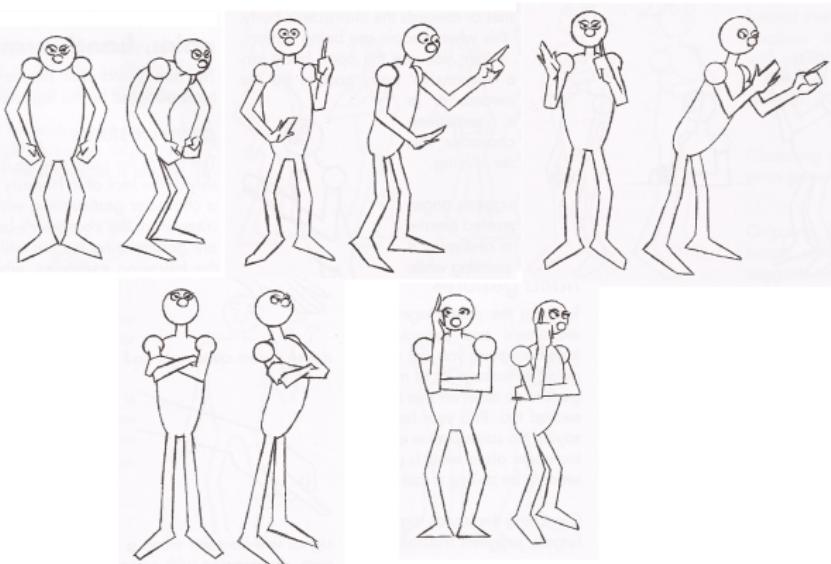
## Atores Digitais - Postura Corporal - Combativo

---

- Combinação das posturas **fechada** e **inclinado para frente**
- Indica pessoas com raiva, querendo se impor à força, desafiadoras, com intenção de ou realmente tendo uma discussão

## Atores Digitais - Postura Corporal - Combativo

- Combinação das posturas **fechada** e **inclinado para frente**
- Indica pessoas com raiva, querendo se impor à força, desafiadoras, com intenção de ou realmente tendo uma discussão



# Atores Digitais - Postura Corporal - Exercício



# Atores Digitais - Postura Corporal - Exercício



*Monica*      **Reflexiva**  
*Cebolinha*      **Reflexivo**



*Monica*      ?  
*Cebolinha*      **Combativo**



*Monica*      **Fugitiva**  
*Cebolinha*      **Combativo**

# Atores Digitais - Postura Corporal - Exercício



*Monica*  
*Cebolinha*      **Combativa**  
**Reflexivo**



*Monica*  
*Cebolinha*      ?  
**Responsivo**



*Monica*  
*Cebolinha*      **Responsiva**  
**Reflexivo**

## Atores Digitais - Palmas, Mãos, Braços e Pernas

---

**Palmas** quando **abertas voltadas para cima** sugerem abertura, honestidade (ou o oposto quando voltadas para baixo); **palmas fechadas** sugerem stress, raiva, medo

# Atores Digitais - Palmas, M  os, Bra  os e Pernas

---

**Palmas** quando **abertas voltadas para cima** sugerem abertura, honestidade (ou o oposto quando voltadas para baixo); **palmas fechadas** sugerem stress, raiva, medo

**M  os** **esfregar** as m  os sugere ansiedade, nervosismo; **apertar** as m  os com os **dedos cruzados** indica emo  es negativas (frustra  o, p.ex.); **tocar as pontas dos dedos** e segurar as m  os no alto = sensa  o de superioridade; m  os para baixo = aten  o, foco

# Atores Digitais - Palmas, Mãos, Braços e Pernas

---

**Palmas** quando **abertas voltadas para cima** sugerem abertura, honestidade (ou o oposto quando voltadas para baixo); **palmas fechadas** sugerem stress, raiva, medo

**Mãos** **esfregar** as mãos sugere ansiedade, nervosismo; **apertar** as mãos com os **dedos cruzados** indica emoções negativas (frustração, p.ex.); **tocar as pontas dos dedos** e segurar as mãos no alto = sensação de superioridade; mãos para baixo = atenção, foco

**Braços** quando **cruzados** = sensação de insegurança; e se ainda, estiver segurando os braços indica nervosismo, stress.

**Segurar o cotovelo ou pulsos** indica sensação de desconforto, inferioridade

# Atores Digitais - Palmas, M  os, Bra  os e Pernas

---

**Palmas** quando **abertas voltadas para cima** sugerem abertura, honestidade (ou o oposto quando voltadas para baixo); **palmas fechadas** sugerem stress, raiva, medo

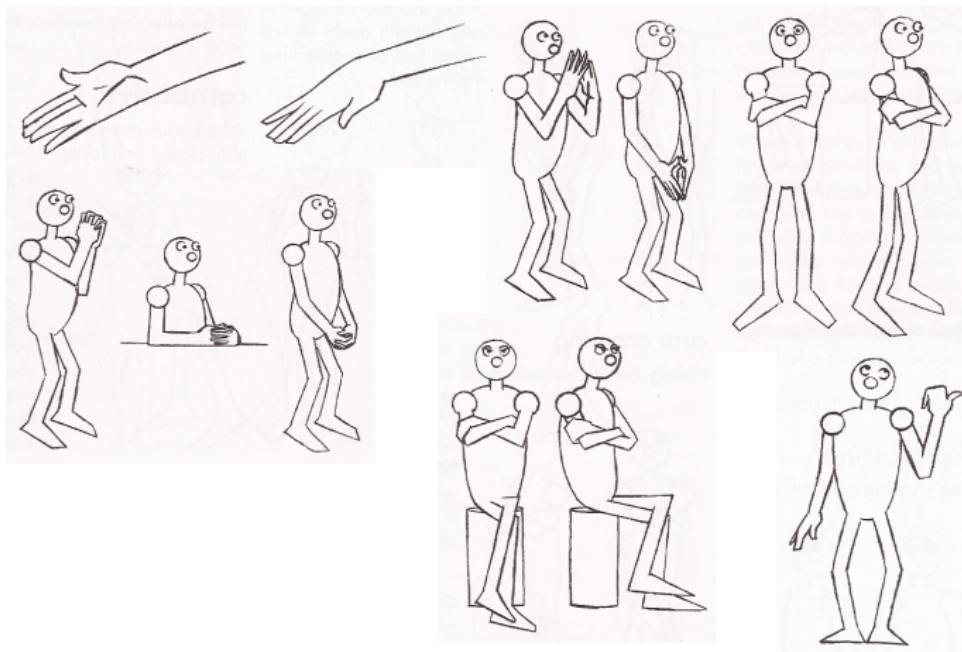
**M  os** **esfregar** as m  os sugere ansiedade, nervosismo; **apertar** as m  os com os **dedos cruzados** indica emo  es negativas (frustra  o, p.ex.); **tocar as pontas dos dedos** e segurar as m  os no alto = sensa  o de superioridade; m  os para baixo = aten  o, foco

**Bra  os** quando **cruzados** = sensa  o de inseguran  a; e se ainda, estiver segurando os bra  os indica nervosismo, stress.

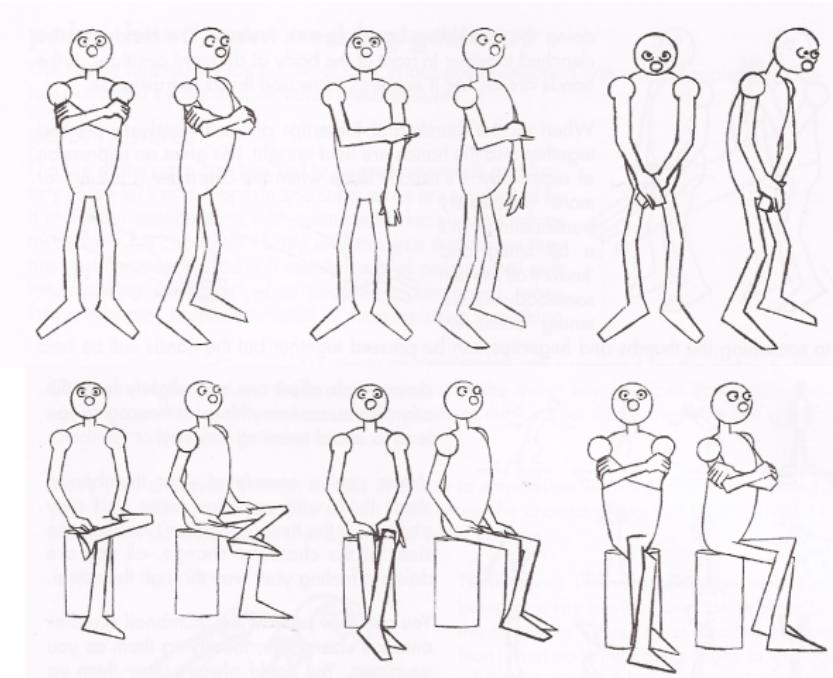
**Segurar o cotovelo ou pulsos** indica sensa  o de desconforto, inferioridade

**Pernas** similar aos bra  os por  m com maior intensidade. **Pernas cruzadas** denotam 'defensiva' (ou simplesmente procura por conforto). Combinar poses para bra  os e pernas resulta em uma atua  o bem mais convincente

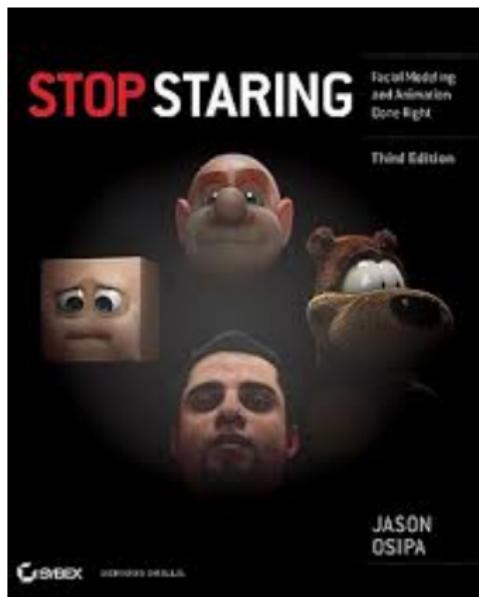
# Atores Digitais - Palmas, Mão, Braços e Pernas



# Atores Digitais - Palmas, Mãos, Braços e Pernas



# Atores Digitais - Expressão Facial e Lip Sync



INTRODUCING  
CHARACTER ANIMATION  
WITH BLENDER

 | SERIOUS SKILLS.

Tony Mullen

Foreword by *Tim Rosemeyer*, Blender Foundation  
Chairman, and *Rossen Radov*, Director of the world's  
first open-source film, *Elchhorn's Dream*.

# Atores Digitais - Animação Esquelética vs *Blend Shapes*

---

Animação Esquelética é um conjunto de objetos hierárquicos (ossos) agem como origens para sistemas de coordenadas locais de malhas deformáveis de polígonos

# Atores Digitais - Animação Esquelética vs *Blend Shapes*

---

**Animação Esquelética** um conjunto de objetos hierárquicos (ossos) agem como origens para sistemas de coordenadas locais de malhas deformáveis de polígonos  
Ao se reposicionar o osso, toda a malha deformável associada a este, também se ajusta

# Atores Digitais - Animação Esquelética vs *Blend Shapes*

---

**Animação Esquelética** um conjunto de objetos hierárquicos (ossos) agem como origens para sistemas de coordenadas locais de malhas deformáveis de polígonos

Ao se reposicionar o osso, toda a malha deformável associada a este, também se ajusta

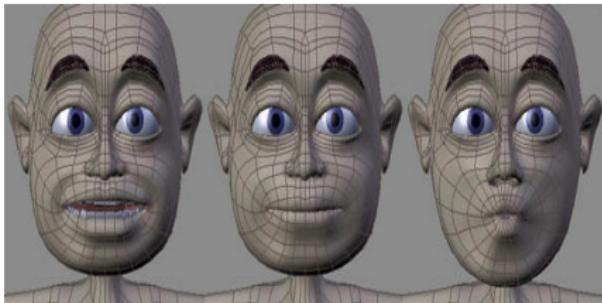
**Blend Shapes** um conjunto de malhas poligonais independentes (representando diferentes estados da personagens) são combinadas através de interpolação linear para produzir um estado resultante

# Atores Digitais - Animação Esquelética vs *Blend Shapes*

**Animação Esquelética** um conjunto de objetos hierárquicos (ossos) agem como origens para sistemas de coordenadas locais de malhas deformáveis de polígonos

Ao se reposicionar o osso, toda a malha deformável associada a este, também se ajusta

**Blend Shapes** um conjunto de malhas poligonais independentes (representando diferentes estados da personagens) são combinadas através de interpolação linear para produzir um estado resultante



## Atores Digitais - *Blend Shapes*

---

- Pose Resultante

$$\sum_{i=0}^N \varpi_i \times \text{Pose}_i$$

# Atores Digitais - *Blend Shapes*

---

- Pose Resultante

$$\sum_{i=0}^N \varpi_i \times Pose_i$$

- **Vantagens:**

- permite a representação de movimentos suaves (detalhados)

# Atores Digitais - *Blend Shapes*

---

- Pose Resultante

$$\sum_{i=0}^N \varpi_i \times \text{Pose}_i$$

- **Vantagens:**

- permite a representação de movimentos suaves (detalhados)
  - duas **blend shapes** quaisquer sempre podem ser interpoladas linearmente sem oclusão

# Atores Digitais - *Blend Shapes*

---

- Pose Resultante

$$\sum_{i=0}^N \varpi_i \times Pose_i$$

- **Vantagens:**

- permite a representação de movimentos suaves (detalhados)
  - duas **blend shapes** quaisquer sempre podem ser interpoladas linearmente sem oclusão

- **Desvantagens:**

- precisa-se ter um banco grande de formas básicas para se produzir variabilidade de composição de formas

# Atores Digitais - *Blend Shapes*

---

## ■ Pose Resultante

$$\sum_{i=0}^N \varpi_i \times \text{Pose}_i$$

## ■ Vantagens:

- permite a representação de movimentos suaves (detalhados)
- duas **blend shapes** quaisquer sempre podem ser interpoladas linearmente sem oclusão

## ■ Desvantagens:

- precisa-se ter um banco grande de formas básicas para se produzir variabilidade de composição de formas
- é chamado de 'colisão' quando duas formas não misturam de forma apropriada. Dadas duas formas  $A$  e  $B$ , existe uma colisão possível  $AB$ . Com 3 formas temos:  $AB, AC, BC, ABC$ . Com 4 formas já são 11 possíveis colisões. Conclusão: quantas mais formas houverem no banco de formas, mais complexo se torna tratar o problema das colisões

# Atores Digitais - *Blend Shapes*

---

## ■ Pose Resultante

$$\sum_{i=0}^N \varpi_i \times Pose_i$$

## ■ Vantagens:

- permite a representação de movimentos suaves (detalhados)
- duas **blend shapes** quaisquer sempre podem ser interpoladas linearmente sem oclusão

## ■ Desvantagens:

- precisa-se ter um banco grande de formas básicas para se produzir variabilidade de composição de formas
- é chamado de 'colisão' quando duas formas não misturam de forma apropriada. Dadas duas formas  $A$  e  $B$ , existe uma colisão possível  $AB$ . Com 3 formas temos:  $AB, AC, BC, ABC$ . Com 4 formas já são 11 possíveis colisões. Conclusão: quantas mais formas houverem no banco de formas, mais complexo se torna tratar o problema das colisões

[http://wiki.blender.org/index.php/Doc:2.6/Manual/Animation/Techs/Shape/Shape\\_Keys](http://wiki.blender.org/index.php/Doc:2.6/Manual/Animation/Techs/Shape/Shape_Keys)

# Atores Digitais - Animação Esquelética vs *Blend Shapes*



# Atores Digitais - Expressões Faciais

---

- 7 emoções básicas: desprezo, felicidade, medo, nojo, raiva, surpresa e tristeza



Ekman, P. and Friesen, W. V. (1978).

*The Facial Action Coding System: A Technique for the Measurement of Facial Movement.*

Consulting Psychologists Press, Palo Alto.

# Atores Digitais - Expressões Faciais

---

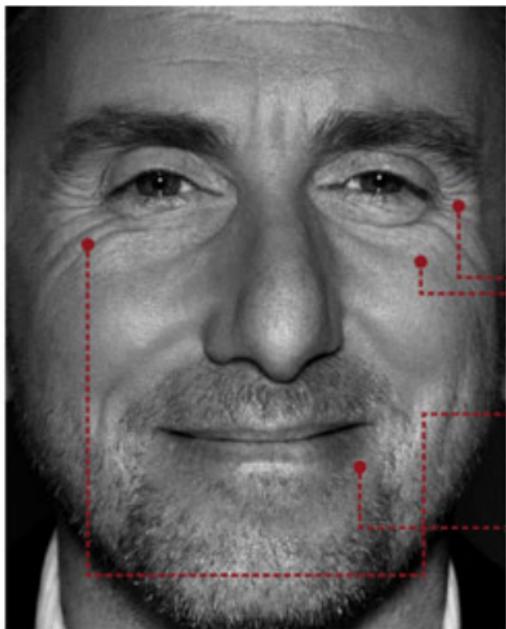
- 7 emoções básicas: desprezo, felicidade, medo, nojo, raiva, surpresa e tristeza
  -  Ekman, P. and Friesen, W. V. (1978).  
*The Facial Action Coding System: A Technique for the Measurement of Facial Movement.*  
Consulting Psychologists Press, Palo Alto.
- Um modelo de descrição de movimentação da musculatura facial foi proposta: *Facial Action Coding System (FACS)*
  -  Keitner, D. and Ekman, P. (2000).  
Facial expression of emotion.  
In Lewis, M. and Haviland-Jones, J., editors, *Handbook of Emotions*, pages 236–249. Guilford Publications Inc., New York, NY, USA, 2nd edition.

# Atores Digitais - Expressões Faciais

AU1 	AU2 	AU4 	AU5 	AU6 
AU7 	AU9 	AU12 	AU15 	AU17 
AU23 	AU24 	AU25 	AU26 	AU27 

[http://en.wikipedia.org/wiki/Facial\\_Action\\_Coding\\_System](http://en.wikipedia.org/wiki/Facial_Action_Coding_System)

# Atores Digitais - Expressões Faciais - Alegria



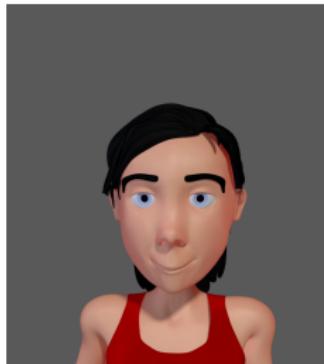
## happiness

A real smile always includes:

- ① crow's feet wrinkles
- ② pushed up cheeks
- ③ movement from muscle that orbits the eye
- ④ lips drawn up and back (teeth showing or not)

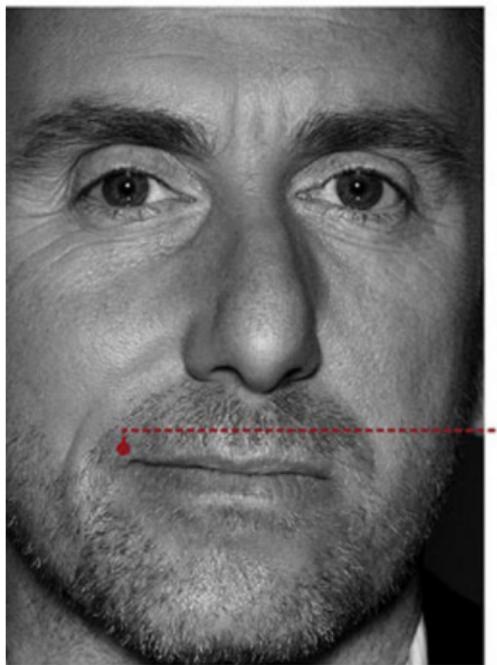
## Atores Digitais - Expressões Faciais - Alegria

---



## Atores Digitais - Expressões Faciais - Desprezo

---

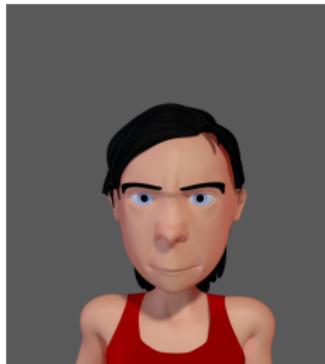
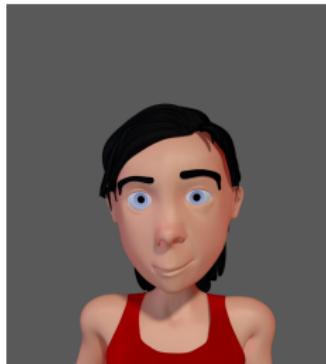


# contempt

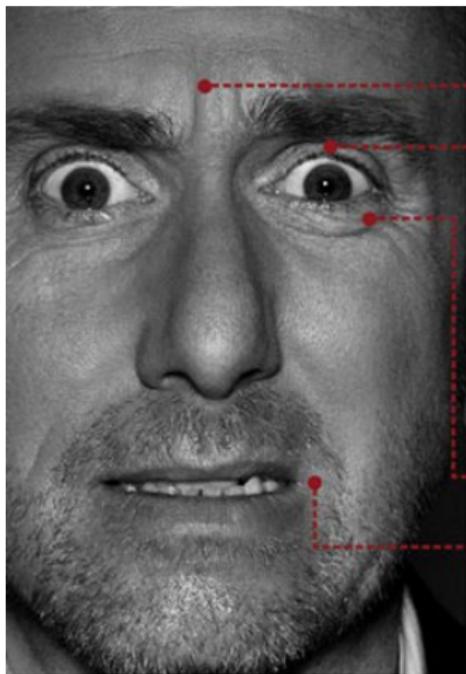
① lip corner tightened  
and raised on only  
one side of face

# Atores Digitais - Expressões Faciais - Desprezo

---



## Atores Digitais - Expressões Faciais - Medo

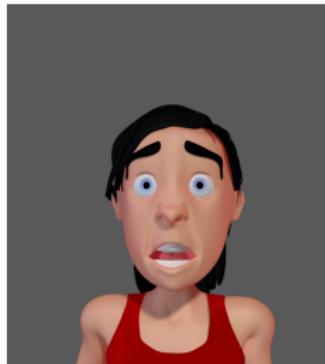
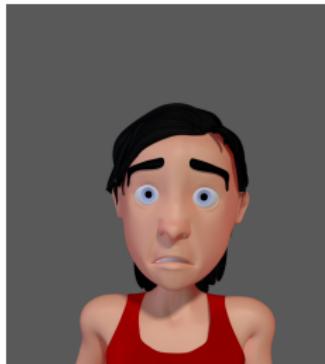
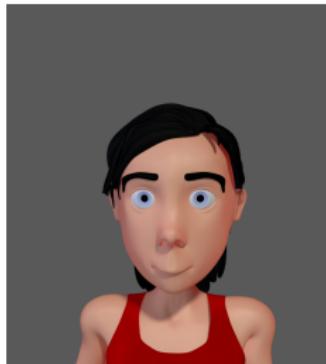


# fear

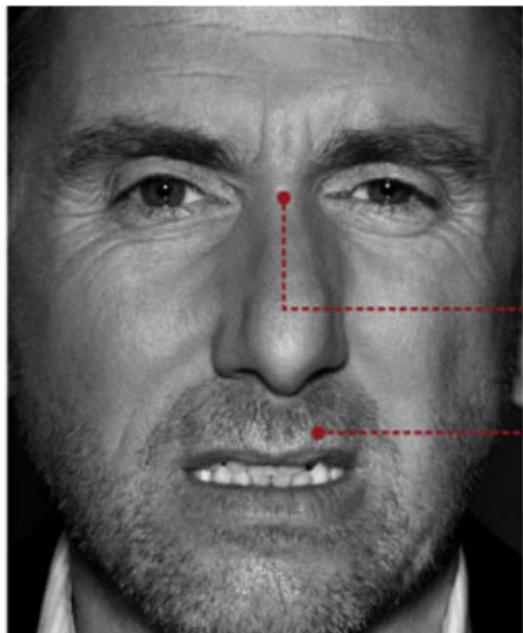
- ① eyebrows raised and pulled together
- ② raised upper eyelids
- ③ tensed lower eyelids
- ④ lips slightly stretched horizontally back to ears

## Atores Digitais - Expressões Faciais - Medo

---



## Atores Digitais - Expressões Faciais - Nojo



**disgust**

① nose wrinkling

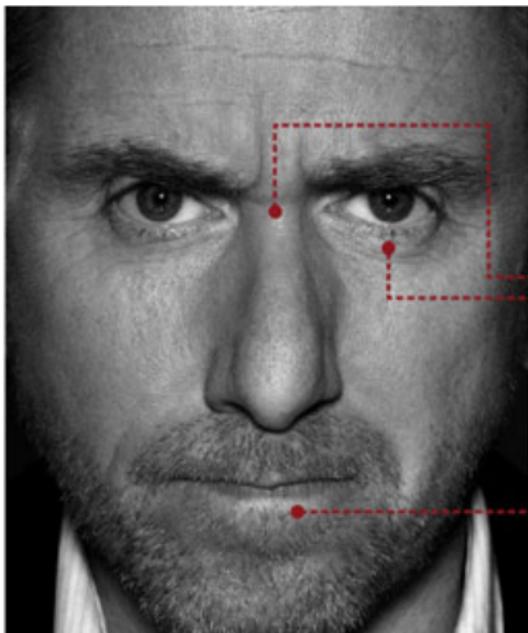
② upper lip raised

## Atores Digitais - Expressões Faciais - Nojo

---



## Atores Digitais - Expressões Faciais - Raiva



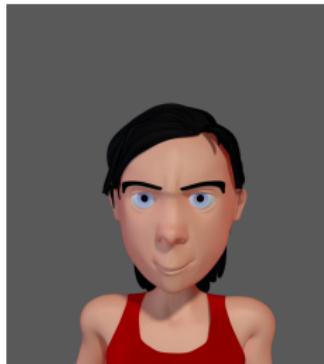
### anger

- ① eyebrows down and together
- ② eyes glare, lower lid tense
- ③ narrowing of the lips, jaw juts forward

*People often purse their lips when they're holding something back from you*

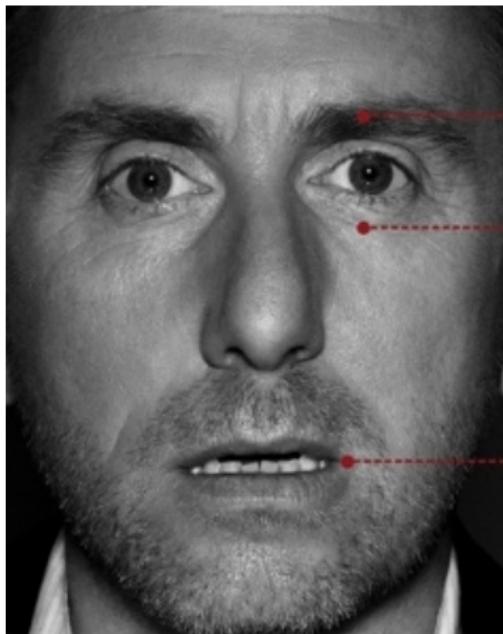
# Atores Digitais - Expressões Faciais - Raiva

---



## Atores Digitais - Expressões Faciais - Surpresa

---



# surprise

Lasts for only one second:

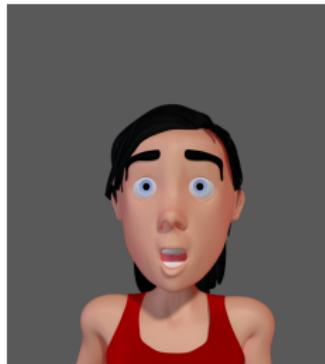
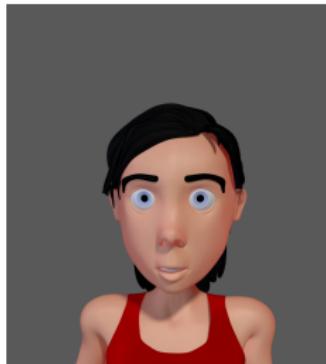
① eyebrows raised

② eyes widened

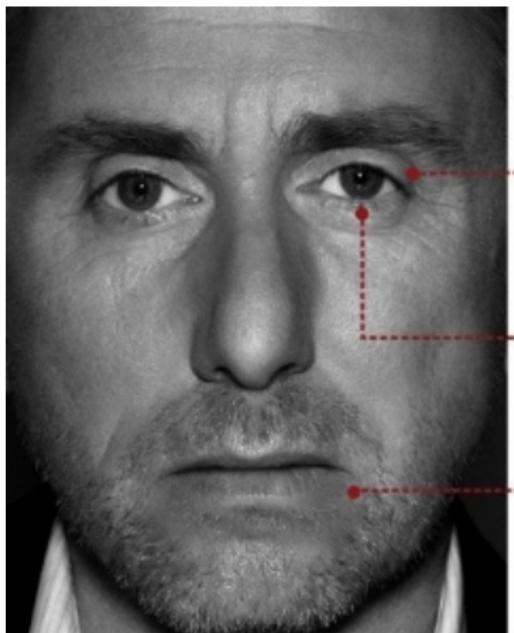
③ mouth open

## Atores Digitais - Expressões Faciais - Surpresa

---



## Atores Digitais - Expressões Faciais - Tristeza

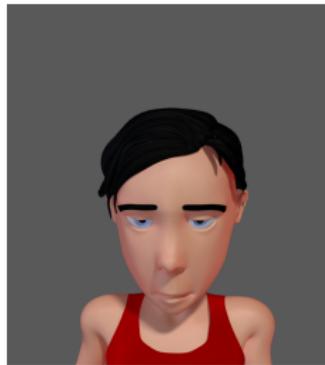


# sadness

- ① drooping upper eyelids
- ② losing focus in eyes
- ③ slight pulling down of lip corners

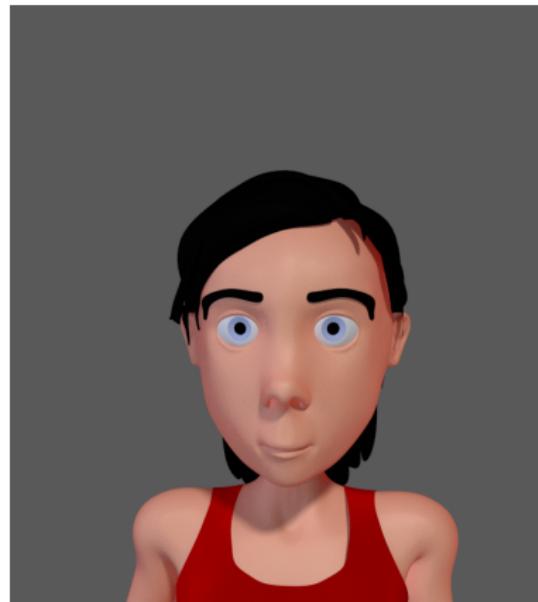
# Atores Digitais - Expressões Faciais - Tristeza

---



## Atores Digitais - Expressões Faciais - Neutro

---



## Atores Digitais - Expressões Faciais - Blender

---

- Em Blender o processo de animação de FACS é realizado em dois passos:

# Atores Digitais - Expressões Faciais - Blender

---

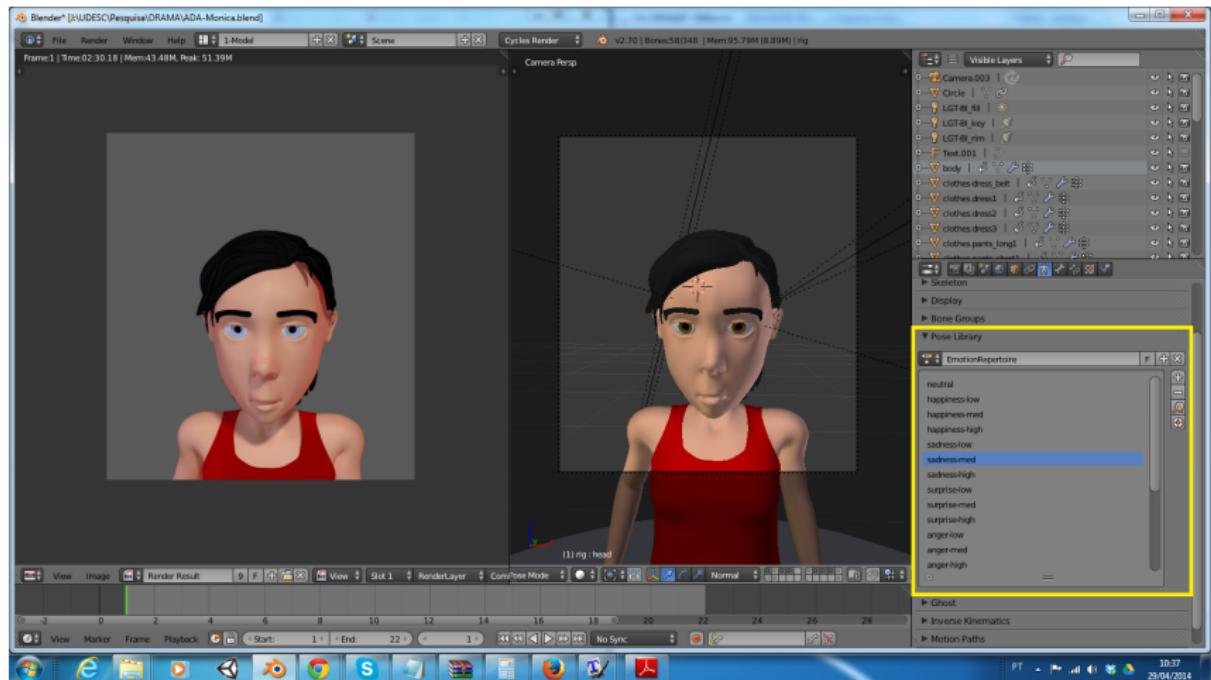
- Em Blender o processo de animação de FACS é realizado em dois passos:
  1. criação de uma biblioteca de poses (*pose library*) contendo todas as expressões faciais

# Atores Digitais - Expressões Faciais - Blender

---

- Em Blender o processo de animação de FACS é realizado em dois passos:
  1. criação de uma biblioteca de poses (*pose library*) contendo todas as expressões faciais
  2. criação de uma timeline via quadros-chave onde as poses são importadas da biblioteca

# Atores Digitais - Expressões Faciais - Blender



## Atores Digitais - Sincronização Labial

---

- Sincronização labial é o processo de se animar as posições dos lábios a fim de imitar o movimento da boca durante a fala humana

## Atores Digitais - Sincronização Labial

---

- Sincronização labial é o processo de se animar as posições dos lábios a fim de imitar o movimento da boca durante a fala humana
- Em sua forma mais simples são duas as dimensões a serem consideradas:

## Atores Digitais - Sincronização Labial

---

- Sincronização labial é o processo de se animar as posições dos lábios a fim de imitar o movimento da boca durante a fala humana
- Em sua forma mais simples são duas as dimensões a serem consideradas:  
**Aberto vs Fechado** define o grau de abertura do maxilar, variando de totalmente aberto a totalmente fechado

## Atores Digitais - Sincronização Labial

---

- Sincronização labial é o processo de se animar as posições dos lábios a fim de imitar o movimento da boca durante a fala humana
- Em sua forma mais simples são duas as dimensões a serem consideradas:  
**Aberto vs Fechado** define o grau de abertura do maxilar, variando de totalmente aberto a totalmente fechado  
**Largo vs Curto** define o grau de compressão dos lábios, variando de totalmente comprimidos (curto) a totalmente esticados (largo)

## Atores Digitais - Sincronização Labial

---

- Sincronização labial é o processo de se animar as posições dos lábios a fim de imitar o movimento da boca durante a fala humana
- Em sua forma mais simples são duas as dimensões a serem consideradas:  
**Aberto vs Fechado** define o grau de abertura do maxilar, variando de totalmente aberto a totalmente fechado  
**Largo vs Curto** define o grau de compressão dos lábios, variando de totalmente comprimidos (curto) a totalmente esticados (largo)
- O processo de animação de sincronização labial passa pelo diferenciação entre os conceitos de **fonemas** e **visemas**

# Atores Digitais - Sincronização Labial

---

- O que são **fonemas** ?

# Atores Digitais - Sincronização Labial

---

- O que são **fonemas** ?

- O *fonema* é a menor unidade linguística. É o elemento a partir do qual os vocábulos (sílabas) de um idioma são formados

# Atores Digitais - Sincronização Labial

---

## ■ O que são **fonemas** ?

- O *fonema* é a menor unidade linguística. É o elemento a partir do qual os vocábulos (sílabas) de um idioma são formados
- A fonoaudiologia estuda os sons produzidos pela voz e define os fonemas linguísticos como contrastem sonoros que nos permitem diferenciar palavras

# Atores Digitais - Sincronização Labial

---

## ■ O que são **fonemas** ?

- O *fonema* é a menor unidade linguística. É o elemento a partir do qual os vocábulos (sílabas) de um idioma são formados
- A fonoaudiologia estuda os sons produzidos pela voz e define os fonemas linguísticos como contrastem sonoros que nos permitem diferenciar palavras
- Exemplo: a palavra **taxi** (4 letras) = /t/ /a/ /k/ /s/ /i/ (5 fonemas)

# Atores Digitais - Sincronização Labial

---

## ■ O que são **fonemas** ?

- O *fonema* é a menor unidade linguística. É o elemento a partir do qual os vocábulos (sílabas) de um idioma são formados
- A fonoaudiologia estuda os sons produzidos pela voz e define os fonemas linguísticos como contrastem sonoros que nos permitem diferenciar palavras
- Exemplo: a palavra **taxi** (4 letras) = /t/ /a/ /k/ /s/ /i/ (5 fonemas)
- Exemplo 2: a palavra **manhã** (5 letras) = /m/ /a/ /nh/ /ã/ (4 fonemas)

# Atores Digitais - Sincronização Labial

---

## ■ O que são **fonemas** ?

- O *fonema* é a menor unidade linguística. É o elemento a partir do qual os vocábulos (sílabas) de um idioma são formados
- A fonoaudiologia estuda os sons produzidos pela voz e define os fonemas linguísticos como contrastem sonoros que nos permitem diferenciar palavras
- Exemplo: a palavra **taxi** (4 letras) = /t/ /a/ /k/ /s/ /i/ (5 fonemas)
- Exemplo 2: a palavra **manhã** (5 letras) = /m/ /a/ /nh/ /ã/ (4 fonemas)

## ■ O que são **visemas** ?

# Atores Digitais - Sincronização Labial

---

## ■ O que são **fonemas** ?

- O *fonema* é a menor unidade linguística. É o elemento a partir do qual os vocábulos (sílabas) de um idioma são formados
- A fonoaudiologia estuda os sons produzidos pela voz e define os fonemas linguísticos como contrastem sonoros que nos permitem diferenciar palavras
- Exemplo: a palavra **taxi** (4 letras) = /t/ /a/ /k/ /s/ /i/ (5 fonemas)
- Exemplo 2: a palavra **manhã** (5 letras) = /m/ /a/ /nh/ /ã/ (4 fonemas)

## ■ O que são **visemas** ?

- O *visema* é a menor unidade da sincronização labial. Um visema representa o formato dos lábios ao pronunciar determinado fonema

# Atores Digitais - Sincronização Labial

---

## ■ O que são **fonemas** ?

- O *fonema* é a menor unidade linguística. É o elemento a partir do qual os vocábulos (sílabas) de um idioma são formados
- A fonoaudiologia estuda os sons produzidos pela voz e define os fonemas linguísticos como contrastem sonoros que nos permitem diferenciar palavras
- Exemplo: a palavra **taxi** (4 letras) = /t/ /a/ /k/ /s/ /i/ (5 fonemas)
- Exemplo 2: a palavra **manhã** (5 letras) = /m/ /a/ /nh/ /ã/ (4 fonemas)

## ■ O que são **visemas** ?

- O *visema* é a menor unidade da sincronização labial. Um visema representa o formato dos lábios ao pronunciar determinado fonema
- Contrariamente ao que se possa imaginar, a relação entre fonemas e visemas não é 1:1, ou seja, pode existir mais de um fonema associado a um mesmo visema

# Atores Digitais - Sincronização Labial

---

## ■ O que são **fonemas** ?

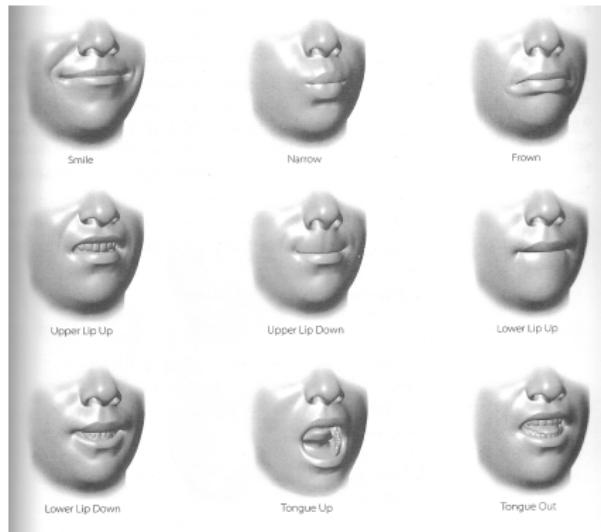
- O *fonema* é a menor unidade linguística. É o elemento a partir do qual os vocábulos (sílabas) de um idioma são formados
- A fonoaudiologia estuda os sons produzidos pela voz e define os fonemas linguísticos como contrastem sonoros que nos permitem diferenciar palavras
- Exemplo: a palavra **taxi** (4 letras) = /t/ /a/ /k/ /s/ /i/ (5 fonemas)
- Exemplo 2: a palavra **manhã** (5 letras) = /m/ /a/ /nh/ /ã/ (4 fonemas)

## ■ O que são **visemas** ?

- O *visema* é a menor unidade da sincronização labial. Um visema representa o formato dos lábios ao pronunciar determinado fonema
- Contrariamente ao que se possa imaginar, a relação entre fonemas e visemas não é 1:1, ou seja, pode existir mais de um fonema associado a um mesmo visema
- Na prática, há apenas um pequeno conjunto de visemas em relação à grande quantidade de fonemas de um dado idioma

# Atores Digitais - Sincronização Labial

---

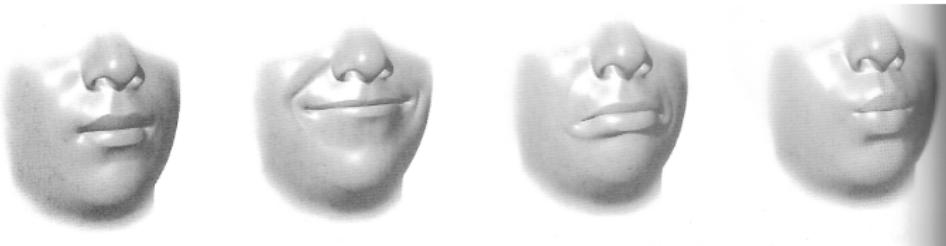


9 formatos básicos dos lábios

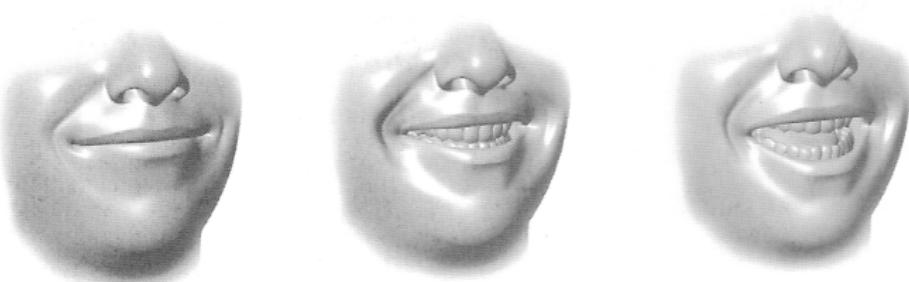
## Atores Digitais - Sincronização Labial

---

- **B/M/P**



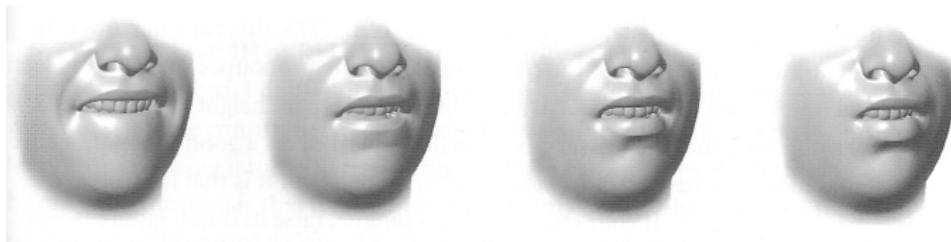
- **EE**



# Atores Digitais - Sincronização Labial

---

- F/V

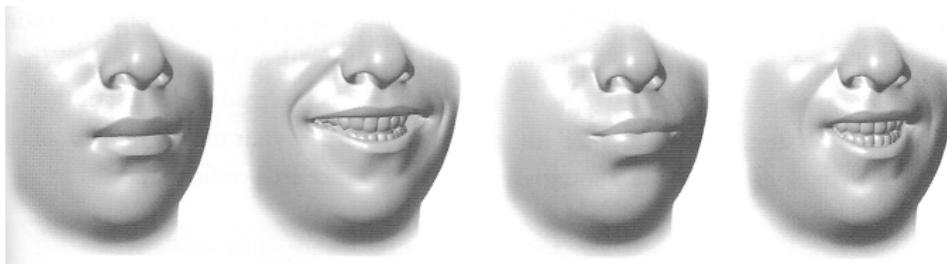


- OO

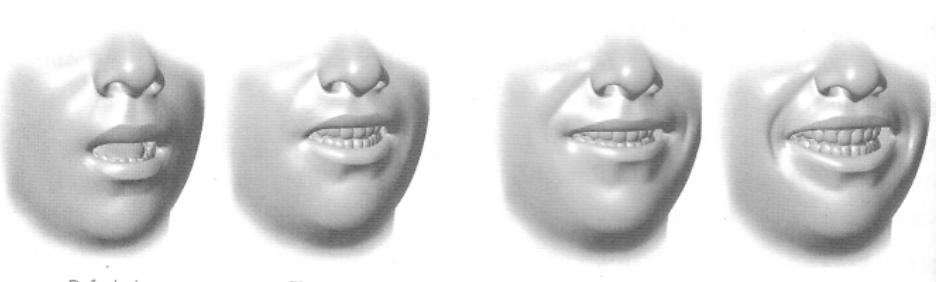


# Atores Digitais - Sincronização Labial

- IH/T/S



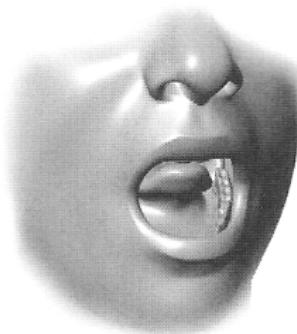
- R



## Atores Digitais - Sincronização Labial

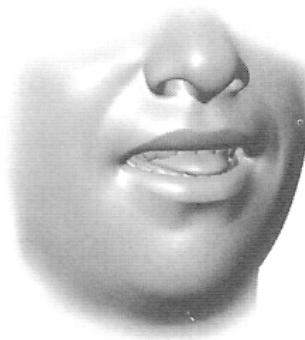
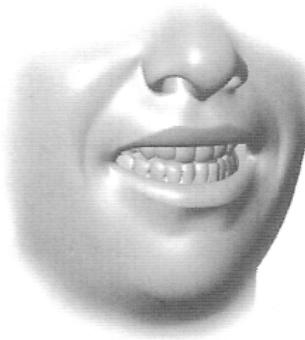
---

- Um **não-visema** representa um formato dos lábios que sofre inflexão dependendo do visema anterior.
- L/N



## Atores Digitais - Sincronização Labial

- D/SH/TH/NG/J (G suave)/H



- AW/OH/UH



# Atores Digitais - Sincronização Labial

- EH/AH/UH



- **G forte/K** não tem envolvimento labial apenas som gutural (garganta)

## Atores Digitais - Sincronização Labial

---

- O processo de transformação de uma frase em uma lista de visemas animáveis depende muito da sensibilidade do animador

## Atores Digitais - Sincronização Labial

---

- O processo de transformação de uma frase em uma lista de visemas animáveis depende muito da sensibilidade do animador
- Deve-se converter a lista de fonemas em uma lista de visemas e então remover aqueles visemas redundantes e/ou que não produzem um movimento efetivo dos lábios (em relação ao visema anterior)

## Atores Digitais - Sincronização Labial

---

- O processo de transformação de uma frase em uma lista de visemas animáveis depende muito da sensibilidade do animador
- Deve-se converter a lista de fonemas em uma lista de visemas e então remover aqueles visemas redundantes e/ou que não produzem um movimento efetivo dos lábios (em relação ao visema anterior)
- Exemplo: a palavra “*spelling*”

## Atores Digitais - Sincronização Labial

---

- O processo de transformação de uma frase em uma lista de visemas animáveis depende muito da sensibilidade do animador
- Deve-se converter a lista de fonemas em uma lista de visemas e então remover aqueles visemas redundantes e/ou que não produzem um movimento efetivo dos lábios (em relação ao visema anterior)
- Exemplo: a palavra “*spelling*”
- Separando em visemas temos: **S - P - EH - L - IH - NG**

## Atores Digitais - Sincronização Labial

---

- O processo de transformação de uma frase em uma lista de visemas animáveis depende muito da sensibilidade do animador
- Deve-se converter a lista de fonemas em uma lista de visemas e então remover aqueles visemas redundantes e/ou que não produzem um movimento efetivo dos lábios (em relação ao visema anterior)
- Exemplo: a palavra “*spelling*”
- Separando em visemas temos: **S - P - EH - L - IH - NG**
- O visema ‘S’ é um dos casos onde o som é produzido principalmente na língua (ao invés de nos lábios). Uma sugestão seria então suprimí-lo: **P - EH - L - IH - NG**

## Atores Digitais - Sincronização Labial

---

- O processo de transformação de uma frase em uma lista de visemas animáveis depende muito da sensibilidade do animador
- Deve-se converter a lista de fonemas em uma lista de visemas e então remover aqueles visemas redundantes e/ou que não produzem um movimento efetivo dos lábios (em relação ao visema anterior)
- Exemplo: a palavra “*spelling*”
- Separando em visemas temos: **S - P - EH - L - IH - NG**
- O visema ‘S’ é um dos casos onde o som é produzido principalmente na língua (ao invés de nos lábios). Uma sugestão seria então suprimí-lo: **P - EH - L - IH - NG**
- Idem para o visema ‘L’ (também ocorre dentro da boca, na língua): **P - EH - IH - NG**

## Atores Digitais - Sincronização Labial

---

- O processo de transformação de uma frase em uma lista de visemas animáveis depende muito da sensibilidade do animador
- Deve-se converter a lista de fonemas em uma lista de visemas e então remover aqueles visemas redundantes e/ou que não produzem um movimento efetivo dos lábios (em relação ao visema anterior)
- Exemplo: a palavra “*spelling*”
- Separando em visemas temos: **S - P - EH - L - IH - NG**
- O visema ‘S’ é um dos casos onde o som é produzido principalmente na língua (ao invés de nos lábios). Uma sugestão seria então suprimí-lo: **P - EH - L - IH - NG**
- Idem para o visema ‘L’ (também ocorre dentro da boca, na língua): **P - EH - IH - NG**
- O visema ‘NG’ é um som gutural (originado na garganta) e portanto também pode ser suprimido: **P - EH - IH**

## Atores Digitais - Sincronização Labial

---

- O processo de transformação de uma frase em uma lista de visemas animáveis depende muito da sensibilidade do animador
- Deve-se converter a lista de fonemas em uma lista de visemas e então remover aqueles visemas redundantes e/ou que não produzem um movimento efetivo dos lábios (em relação ao visema anterior)
- Exemplo: a palavra “*spelling*”
- Separando em visemas temos: **S - P - EH - L - IH - NG**
- O visema ‘S’ é um dos casos onde o som é produzido principalmente na língua (ao invés de nos lábios). Uma sugestão seria então suprimí-lo: **P - EH - L - IH - NG**
- Idem para o visema ‘L’ (também ocorre dentro da boca, na língua): **P - EH - IH - NG**
- O visema ‘NG’ é um som gutural (originado na garganta) e portanto também pode ser suprimido: **P - EH - IH**
- Agora temos uma escolha a fazer: o visema ‘IH’ é similar ao visema ‘EE’ (apenas entonação diferente). Alternativamente, pode-se substituí-lo para efeito de ênfase. Portanto, o resultado final ficaria **P - EH - EE**

## Atores Digitais - Sincronização Labial

---

Palavra	Passo #1	Final
fountain	F-AH-OO-N-T-IH-N	F-AH-OO-IH
photograph	F-OH-T-OH-R-AH-F	F-OH-T-AH-F
sheperd	SH-EH-P-R-D	SH-EH-P-R
stop staring	S-T-AW-P-S-T-EH-R-NG	S-AW-P-S-EH-R-IH

# Atores Digitais - Sincronização Labial

---

- Alguns aspectos a considerar:

# Atores Digitais - Sincronização Labial

---

- Alguns aspectos a considerar:
  1. identificar a lista de visemas para cada palavra

# Atores Digitais - Sincronização Labial

---

- Alguns aspectos a considerar:
  1. identificar a lista de visemas para cada palavra
  2. identificar as posições aberto/fechado

## Atores Digitais - Sincronização Labial

---

- Alguns aspectos a considerar:

1. identificar a lista de visemas para cada palavra
2. identificar as posições aberto/fechado
3. identificar as posições relativas entre visemas absolutos

# Atores Digitais - Sincronização Labial

---

- Alguns aspectos a considerar:

1. identificar a lista de visemas para cada palavra
2. identificar as posições aberto/fechado
3. identificar as posições relativas entre visemas absolutos
4. inserir '*opositores*' e '*escaladores*'

# Atores Digitais - Sincronização Labial

---

- Alguns aspectos a considerar:

1. identificar a lista de visemas para cada palavra
2. identificar as posições aberto/fechado
3. identificar as posições relativas entre visemas absolutos
4. inserir '*opositores*' e '*escaladores*'

**Opositores** são visemas extras que se coloca entre quadros-chave porém na direção oposta, para efeito de ênfase

# Atores Digitais - Sincronização Labial

---

- Alguns aspectos a considerar:

1. identificar a lista de visemas para cada palavra
2. identificar as posições aberto/fechado
3. identificar as posições relativas entre visemas absolutos
4. inserir 'opositores' e 'escaladores'

**Opositores** são visemas extras que se coloca entre quadros-chave porém na direção oposta, para efeito de ênfase

Exemplo: *Sheet Metal* = SH-EE-T-M-EH-T-UH-L →  
EE-M-EH-T-UH

# Atores Digitais - Sincronização Labial

---

- Alguns aspectos a considerar:

1. identificar a lista de visemas para cada palavra
2. identificar as posições aberto/fechado
3. identificar as posições relativas entre visemas absolutos
4. inserir 'opositores' e 'escaladores'

**Opositores** são visemas extras que se coloca entre quadros-chave porém na direção oposta, para efeito de ênfase

Exemplo: *Sheet Metal* = SH-EE-T-M-EH-T-UH-L →  
EE-M-EH-T-UH

**Escaladores** são visemas extras que se coloca na mesma direção dos antecessores, porém com maior intensidade, para efeito de ênfase

# Atores Digitais - Sincronização Labial

---

- Alguns aspectos a considerar:

1. identificar a lista de visemas para cada palavra
2. identificar as posições aberto/fechado
3. identificar as posições relativas entre visemas absolutos
4. inserir 'opositores' e 'escaladores'

**Opositores** são visemas extras que se coloca entre quadros-chave porém na direção oposta, para efeito de ênfase

Exemplo: *Sheet Metal* = SH-EE-T-M-EH-T-UH-L →  
EE-M-EH-T-UH

**Escaladores** são visemas extras que se coloca na mesma direção dos antecessores, porém com maior intensidade, para efeito de ênfase

Exemplo: *Believe* = B-EE-L-EE-V → B-EE-EE-V

## Atores Digitais - Sincronização Labial - Blender

---

- O processo de sincronização labial em Blender é análogo ao já visto para expressões faciais, e segue os seguintes passos:

## Atores Digitais - Sincronização Labial - Blender

---

- O processo de sincronização labial em Blender é análogo ao já visto para expressões faciais, e segue os seguintes passos:
  1. construção de uma *pose library* para a lista de visemas

## Atores Digitais - Sincronização Labial - Blender

---

- O processo de sincronização labial em Blender é análogo ao já visto para expressões faciais, e segue os seguintes passos:
  1. construção de uma *pose library* para a lista de visemas
  2. inclusão de arquivo de áudio para sincronização áudio-vídeo

# Atores Digitais - Sincronização Labial - Blender

---

- O processo de sincronização labial em Blender é análogo ao já visto para expressões faciais, e segue os seguintes passos:
  1. construção de uma *pose library* para a lista de visemas
  2. inclusão de arquivo de áudio para sincronização áudio-vídeo
  3. anotação do áudio com “marcadores” para representar o quadro exato onde cada visema deve ser inserido

## Atores Digitais - Sincronização Labial - Blender

---

- O processo de sincronização labial em Blender é análogo ao já visto para expressões faciais, e segue os seguintes passos:
  1. construção de uma *pose library* para a lista de visemas
  2. inclusão de arquivo de áudio para sincronização áudio-vídeo
  3. anotação do áudio com “marcadores” para representar o quadro exato onde cada visema deve ser inserido
  4. construção da timeline com os respectivos visemas

# Atores Digitais - Sincronização Labial - Blender



# Exercício Prático Individual - EPI #3

---

**Nome:** Atores Digitais

**Objetivo:** construção de um personagem capaz de expressar atuação cênica (contar uma estória) através de elementos da dramaturgia.

- Enunciado:**
1. escolher um roteiro a ser animado (trecho de uma cena de filme ou animação)
  2. separar apenas o áudio para uso na animação
  3. selecionar em repositórios online, os modelos/personagens necessários (*opensource*)
  4. criar bibliotecas de poses (expressões faciais e lip sync)
  5. produzir uma sequência de quadros através de *keyframing* com base na estrutura *blocking – splining – polishing* para cada uma das camadas de animação vistos (postura corporal, expressão facial e sincronização labial)
  6. compilar um filme resultante

- Requisitos:**
- Estória pode ser um monólogo
  - Devem ser utilizadas as técnicas vistas em aula: *keyframing*, personagens articulados e atores digitais
  - Sugestão: animar elementos como luzes e câmeras
  - É obrigatório conter diálogos
  - Deve conter trilha sonora e/ou efeitos sonoros
  - Duração mínima: 30 segundos
  - Deve apresentar tela de créditos iniciais e finais (o tempo de apresentação dos créditos não contabiliza para o tempo da animação)

**Prazo para Entrega:** 2 semanas (filme + estória + slides ilustrando o processo)

**Peso:** 15% da média semestral

# Simulação Física

- Animação *Straight-ahead*
- Simulação de Corpos Rígidos
- Tratamento de Colisões
- Simulação de Corpos Flexíveis
- Tecidos

# Simulação Física

---

- Existem diversos momentos onde a animação *Pose-to-Pose* se mostra extremamente trabalhosa e complexa

# Simulação Física

---

- Existem diversos momentos onde a animação *Pose-to-Pose* se mostra extremamente trabalhosa e complexa
- E ainda, os resultados obtidos nem sempre são satisfatórios o suficiente em relação à qualidade pretendida

# Simulação Física

---

- Existem diversos momentos onde a animação *Pose-to-Pose* se mostra extremamente trabalhosa e complexa
- E ainda, os resultados obtidos nem sempre são satisfatórios o suficiente em relação à qualidade pretendida
- A animação de **fenômenos realistas**, tais como quedas, arremessos, colisões, explosões, pêndulos, etc. ⇒ **movimentos que resultem da interação de forças**

# Simulação Física

---

- Existem diversos momentos onde a animação *Pose-to-Pose* se mostra extremamente trabalhosa e complexa
- E ainda, os resultados obtidos nem sempre são satisfatórios o suficiente em relação à qualidade pretendida
- A animação de **fenômenos realistas**, tais como quedas, arremessos, colisões, explosões, pêndulos, etc. ⇒ **movimentos que resultem da interação de forças**
- São chamadas de *engine físicas* aos sistemas responsáveis por calcular tais interações de forças, de maneira a simular fenômenos físicos: cinemática, hidrodinâmica, eletromagnetismo, gravitação, etc.

# Simulação Física

---

- Existem diversos momentos onde a animação *Pose-to-Pose* se mostra extremamente trabalhosa e complexa
- E ainda, os resultados obtidos nem sempre são satisfatórios o suficiente em relação à qualidade pretendida
- A animação de **fenômenos realistas**, tais como quedas, arremessos, colisões, explosões, pêndulos, etc. ⇒ **movimentos que resultem da interação de forças**
- São chamadas de *engine físicas* aos sistemas responsáveis por calcular tais interações de forças, de maneira a simular fenômenos físicos: cinemática, hidrodinâmica, eletromagnetismo, gravitação, etc.
- Para tal, tais engines fazem uso das leis da Física: Newton, Einstein, Hooke, Galileo, Bohr, etc.

# Simulação Física

---

- Existem diversos momentos onde a animação *Pose-to-Pose* se mostra extremamente trabalhosa e complexa
- E ainda, os resultados obtidos nem sempre são satisfatórios o suficiente em relação à qualidade pretendida
- A animação de **fenômenos realistas**, tais como quedas, arremessos, colisões, explosões, pêndulos, etc. ⇒ **movimentos que resultem da interação de forças**
- São chamadas de *engine físicas* aos sistemas responsáveis por calcular tais interações de forças, de maneira a simular fenômenos físicos: cinemática, hidrodinâmica, eletromagnetismo, gravitação, etc.
- Para tal, tais engines fazem uso das leis da Física: Newton, Einstein, Hooke, Galileo, Bohr, etc.
- Este tipo de animação é denominado *straight-ahead* dado que, ao invés de o animador construir quadros-chaves, ele configura uma **situação inicial** e um **sistema de forças**, e então aciona a engine de física para realizar a simulação

# Simulação Física em Blender

---

- Blender traz incorporado em seu sistema, a possibilidade de evocar uma engine de física externa (*Bullet*) para a realização de simulações físicas

# Simulação Física em Blender

---

- Blender traz incorporado em seu sistema, a possibilidade de evocar uma engine de física externa (*Bullet*) para a realização de simulações físicas
- Alguns dos recursos da *bullet* são exclusivo para uso da Blender game engine em animações em tempo real (jogos)

# Simulação Física em Blender

---

- Blender traz incorporado em seu sistema, a possibilidade de evocar uma engine de física externa (*Bullet*) para a realização de simulações físicas
- Alguns dos recursos da *bullet* são exclusivo para uso da Blender game engine em animações em tempo real (jogos)
- Caso haja interesse em se utilizar tais recursos em animações *batch render* (timeline), é possível se “gravar” uma simulação como sendo uma sequência de quadros-chaves e então cuidadosamente sincronizar com animações *pose-to-pose* (p.ex. via NLA editor)

# Simulação Física

---

A engine de física do Blender permite suporte a duas grandes classes de simulações:

# Simulação Física

---

A engine de física do Blender permite suporte a duas grandes classes de simulações:

**Simulações de Corpos Rígidos** são aqueles objetos que não sofrem deformação, mesmo sob a ação de forças externas. Exemplos: concreto, metal, pedras

# Simulação Física

---

A engine de física do Blender permite suporte a duas grandes classes de simulações:

**Simulações de Corpos Rígidos** são aqueles objetos que não sofrem deformação, mesmo sob a ação de forças externas. Exemplos: concreto, metal, pedras

Neste tipo de abordagem, as seguintes simulações são possíveis: quedas de corpos, arremessos, colisões e fraturas

# Simulação Física

---

A engine de física do Blender permite suporte a duas grandes classes de simulações:

**Simulações de Corpos Rígidos** são aqueles objetos que não sofrem deformação, mesmo sob a ação de forças externas. Exemplos: concreto, metal, pedras

Neste tipo de abordagem, as seguintes simulações são possíveis: quedas de corpos, arremessos, colisões e fraturas

**Simulações de Corpos Flexíveis** são aqueles objetos que sofrem deformação de seu volume quando sob a ação de forças externas.

Exemplos: tecidos, borracha, objetos amorfos (líquidos, fogo, fumaça, ...), cabelos

# Corpos Rígidos

---

- O processo de simulação de corpos rígidos é bastante simples, dado que este é o uso padrão da engine *bullet*

# Corpos Rígidos

---

- O processo de simulação de corpos rígidos é bastante simples, dado que este é o uso padrão da engine *bullet*
- Passos:

# Corpos Rígidos

---

- O processo de simulação de corpos rígidos é bastante simples, dado que este é o uso padrão da engine *bullet*
- Passos:
  1. Modelar a cena tridimensional, posicionando os objetos

# Corpos Rígidos

---

- O processo de simulação de corpos rígidos é bastante simples, dado que este é o uso padrão da engine *bullet*
- Passos:
  1. Modelar a cena tridimensional, posicionando os objetos
  2. Alterar a engine em uso pelo Blender para **Blender Game**

# Corpos Rígidos

---

- O processo de simulação de corpos rígidos é bastante simples, dado que este é o uso padrão da engine *bullet*
- Passos:
  1. Modelar a cena tridimensional, posicionando os objetos
  2. Alterar a engine em uso pelo Blender para **Blender Game**
  3. Ao fazer isso, na aba *Physics*, todo um novo grupo de parâmetros de configuração se torna disponível

# Corpos Rígidos

---

- O processo de simulação de corpos rígidos é bastante simples, dado que este é o uso padrão da engine *bullet*
- Passos:
  1. Modelar a cena tridimensional, posicionando os objetos
  2. Alterar a engine em uso pelo Blender para **Blender Game**
  3. Ao fazer isso, na aba *Physics*, todo um novo grupo de parâmetros de configuração se torna disponível
  4. Para cada objeto na cena, configura-se o tipo de física a ser aplicado durante a simulação para aquele objeto: *No collision, static, dynamic, rigid body, soft body, occlude, sensor, navigation mesh, character*

# Corpos Rígidos

---

- O processo de simulação de corpos rígidos é bastante simples, dado que este é o uso padrão da engine *bullet*
- Passos:
  1. Modelar a cena tridimensional, posicionando os objetos
  2. Alterar a engine em uso pelo Blender para **Blender Game**
  3. Ao fazer isso, na aba *Physics*, todo um novo grupo de parâmetros de configuração se torna disponível
  4. Para cada objeto na cena, configura-se o tipo de física a ser aplicado durante a simulação para aquele objeto: *No collision, static, dynamic, rigid body, soft body, occlude, sensor, navigation mesh, character*
  5. Notem que cada tipo possui a sua própria lista de parâmetros a configurar

# Corpos Rígidos

---

- O processo de simulação de corpos rígidos é bastante simples, dado que este é o uso padrão da engine *bullet*
- Passos:
  1. Modelar a cena tridimensional, posicionando os objetos
  2. Alterar a engine em uso pelo Blender para **Blender Game**
  3. Ao fazer isso, na aba *Physics*, todo um novo grupo de parâmetros de configuração se torna disponível
  4. Para cada objeto na cena, configura-se o tipo de física a ser aplicado durante a simulação para aquele objeto: *No collision, static, dynamic, rigid body, soft body, occlude, sensor, navigation mesh, character*
  5. Notem que cada tipo possui a sua própria lista de parâmetros a configurar
  6. Por fim, aciona-se o processo de simulação via menu **Game | Start Game Engine**, ou alternativamente pela tecla de atalho **P** no ambiente 3D View

# Corpos Rígidos

---

No Collision nenhuma simulação física é calculada sobre objetos desse tipo

# Corpos Rígidos

---

**No Collision** nenhuma simulação física é calculada sobre objetos desse tipo

**Static** objetos que não sofrem ação da gravidade ou forças externas, porém podem receber colisões (de outros objetos dinâmicos) = **tipo default para a engine de física**

# Corpos Rígidos

---

No Collision nenhuma simulação física é calculada sobre objetos desse tipo

Static objetos que não sofrem ação da gravidade ou forças externas, porém podem receber colisões (de outros objetos dinâmicos) = **tipo default para a engine de física**

Dynamic sofrem a ação da gravidade e colisões porém não são afetados por sistemas de forças externas (p.ex. torque)

# Corpos Rígidos

---

No Collision nenhuma simulação física é calculada sobre objetos desse tipo

Static objetos que não sofrem ação da gravidade ou forças externas, porém podem receber colisões (de outros objetos dinâmicos) = **tipo default para a engine de física**

Dynamic sofrem a ação da gravidade e colisões porém não são afetados por sistemas de forças externas (p.ex. torque)

Rigid body sofrem ação completa do sistema de forças de corpos rígidos

# Corpos Rígidos

---

No Collision nenhuma simulação física é calculada sobre objetos desse tipo

Static objetos que não sofrem ação da gravidade ou forças externas, porém podem receber colisões (de outros objetos dinâmicos) = **tipo default para a engine de física**

Dynamic sofrem a ação da gravidade e colisões porém não são afetados por sistemas de forças externas (p.ex. torque)

Rigid body sofrem ação completa do sistema de forças de corpos rígidos

Soft body análogos aos *rigid body*, permitem ainda a simulação de deformação de malha devido à forças externas

# Corpos Rígidos

---

No Collision nenhuma simulação física é calculada sobre objetos desse tipo

Static objetos que não sofrem ação da gravidade ou forças externas, porém podem receber colisões (de outros objetos dinâmicos) = **tipo default para a engine de física**

Dynamic sofrem a ação da gravidade e colisões porém não são afetados por sistemas de forças externas (p.ex. torque)

Rigid body sofrem ação completa do sistema de forças de corpos rígidos

Soft body análogos aos *rigid body*, permitem ainda a simulação de deformação de malha devido à forças externas

Occlude objetos especiais utilizados como forma de 'esconder' partes do cenário da engine física (occlusão)

# Corpos Rígidos

---

No Collision nenhuma simulação física é calculada sobre objetos desse tipo

Static objetos que não sofrem ação da gravidade ou forças externas, porém podem receber colisões (de outros objetos dinâmicos) = **tipo default para a engine de física**

Dynamic sofrem a ação da gravidade e colisões porém não são afetados por sistemas de forças externas (p.ex. torque)

Rigid body sofrem ação completa do sistema de forças de corpos rígidos

Soft body análogos aos *rigid body*, permitem ainda a simulação de deformação de malha devido à forças externas

Occlude objetos especiais utilizados como forma de 'esconder' partes do cenário da engine física (occlusão)

Sensor utilizados pela Blender game engine para detecção de eventos no ambiente de jogo

# Corpos Rígidos

---

**No Collision** nenhuma simulação física é calculada sobre objetos desse tipo

**Static** objetos que não sofrem ação da gravidade ou forças externas, porém podem receber colisões (de outros objetos dinâmicos) = **tipo default para a engine de física**

**Dynamic** sofrem a ação da gravidade e colisões porém não são afetados por sistemas de forças externas (p.ex. torque)

**Rigid body** sofrem ação completa do sistema de forças de corpos rígidos

**Soft body** análogos aos *rigid body*, permitem ainda a simulação de deformação de malha devido à forças externas

**Occlude** objetos especiais utilizados como forma de ‘esconder’ partes do cenário da engine física (occlusão)

**Sensor** utilizados pela Blender game engine para detecção de eventos no ambiente de jogo

**Navigation Mesh** malhas de navegação utilizadas pela Blender game engine para a construção de personagens autônomos

# Corpos Rígidos

---

**No Collision** nenhuma simulação física é calculada sobre objetos desse tipo

**Static** objetos que não sofrem ação da gravidade ou forças externas, porém podem receber colisões (de outros objetos dinâmicos) = **tipo default para a engine de física**

**Dynamic** sofrem a ação da gravidade e colisões porém não são afetados por sistemas de forças externas (p.ex. torque)

**Rigid body** sofrem ação completa do sistema de forças de corpos rígidos

**Soft body** análogos aos *rigid body*, permitem ainda a simulação de deformação de malha devido à forças externas

**Occlude** objetos especiais utilizados como forma de ‘esconder’ partes do cenário da engine física (occlusão)

**Sensor** utilizados pela Blender game engine para detecção de eventos no ambiente de jogo

**Navigation Mesh** malhas de navegação utilizadas pela Blender game engine para a construção de personagens autônomos

**Character** personagens autônomos (game AI) capazes de navegar em um ambiente 3D através de malhas de navegação

## Corpos Rígidos - Exercício #1

---

- Utilizando a cena inicial padrão (do cubo, câmera e luz), mude a engine para *Blender Game*

## Corpos Rígidos - Exercício #1

---

- Utilizando a cena inicial padrão (do cubo, câmera e luz), mude a engine para *Blender Game*
- Com o objeto Cubo selecionado, na aba *Physics* mude o tipo de física para **Dynamic**

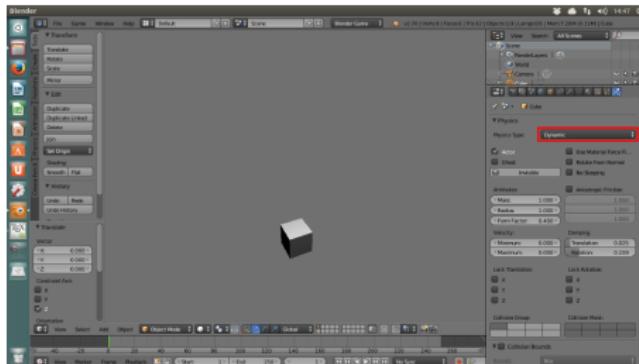
## Corpos Rígidos - Exercício #1

---

- Utilizando a cena inicial padrão (do cubo, câmera e luz), mude a engine para *Blender Game*
- Com o objeto Cubo selecionado, na aba *Physics* mude o tipo de física para **Dynamic**
- No editor 3D View tecle **P** para acionar a engine física

# Corpos Rígidos - Exercício #1

- Utilizando a cena inicial padrão (do cubo, câmera e luz), mude a engine para *Blender Game*
- Com o objeto Cubo selecionado, na aba *Physics* mude o tipo de física para **Dynamic**
- No editor 3D View tecle **P** para acionar a engine física



# Configurando a Engine de Física

---

- No contexto **World** (quando a Blender Game engine está ativa) é possível configurações diversos parâmetros relativos a como a engine de física irá atuar

# Configurando a Engine de Física

- No contexto **World** (quando a Blender Game engine está ativa) é possível configurações diversos parâmetros relativos a como a engine de física irá atuar



## Corpos Rígidos - Exercício #2

---

- Construir um plano horizontal (piso) e definí-lo como tipo físico: *Static*

## Corpos Rígidos - Exercício #2

---

- Construir um plano horizontal (piso) e definí-lo como tipo físico: *Static*
- Construir um cubo achatado em um eixo a fim de se parecer como uma peça de dominó

## Corpos Rígidos - Exercício #2

---

- Construir um plano horizontal (piso) e definí-lo como tipo físico: *Static*
- Construir um cubo achatado em um eixo a fim de se parecer como uma peça de dominó
- Definir o objeto como sendo do tipo **Rigid Body**, especificar parâmetros como **Radius** a fim de comportar o tamanho do objeto, e ainda ativar a opção **Collision Bounds**

## Corpos Rígidos - Exercício #2

---

- Construir um plano horizontal (piso) e definí-lo como tipo físico: *Static*
- Construir um cubo achatado em um eixo a fim de se parecer como uma peça de dominó
- Definir o objeto como sendo do tipo **Rigid Body**, especificar parâmetros como **Radius** a fim de comportar o tamanho do objeto, e ainda ativar a opção **Collision Bounds**
- Através do recurso de duplicação (**Ctrl+D**), criar cópias e posicioná-las lado a lado como na figura

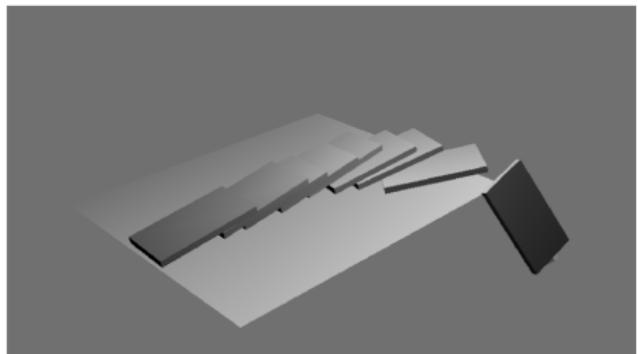
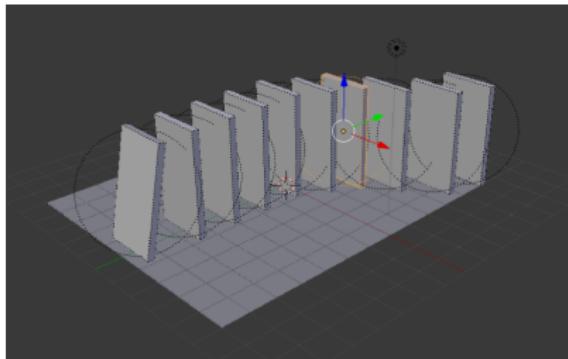
## Corpos Rígidos - Exercício #2

---

- Construir um plano horizontal (piso) e definí-lo como tipo físico: *Static*
- Construir um cubo achatado em um eixo a fim de se parecer como uma peça de dominó
- Definir o objeto como sendo do tipo **Rigid Body**, especificar parâmetros como **Radius** a fim de comportar o tamanho do objeto, e ainda ativar a opção **Collision Bounds**
- Através do recurso de duplicação (**Ctrl+D**), criar cópias e posicioná-las lado a lado como na figura
- Selecionar a primeira peça e rotacioná-la ligeiramente a fim de desestabilizar sua posição

## Corpos Rígidos - Exercício #2

---



## Gravando Simulações Físicas

---

- Simulações físicas per se não podem ser renderizadas em um filme ( dado que são calculadas em tempo real)

## Gravando Simulações Físicas

---

- Simulações físicas por se não podem ser renderizadas em um filme (dados que são calculados em tempo real)
- Porém, o Blender oferece um recurso de ‘buferização’ de simulação = gravar os quadros gerados pela simulação como quadros-chaves na timeline para posterior utilização em um filme animado

## Gravando Simulações Físicas

---

- Simulações físicas por se não podem ser renderizadas em um filme (dados que são calculados em tempo real)
- Porém, o Blender oferece um recurso de ‘buferização’ de simulação = gravar os quadros gerados pela simulação como quadros-chaves na timeline para posterior utilização em um filme animado
- O processo para se realizar isso segue dois passos:

# Gravando Simulações Físicas

---

- Simulações físicas por se não podem ser renderizadas em um filme (dados que são calculados em tempo real)
- Porém, o Blender oferece um recurso de ‘buferização’ de simulação = gravar os quadros gerados pela simulação como quadros-chaves na timeline para posterior utilização em um filme animado
- O processo para se realizar isso segue dois passos:
  1. Ativar o recurso de gravação de quadros no menu **Game | Record Animation**

# Gravando Simulações Físicas

---

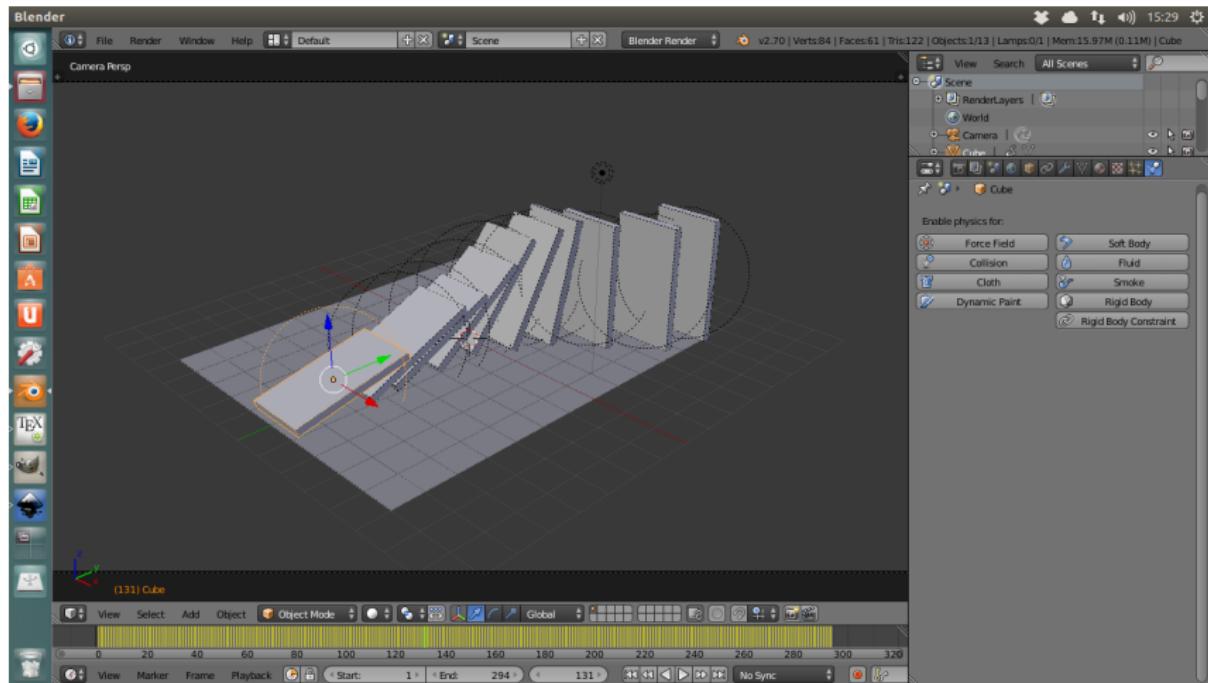
- Simulações físicas por se não podem ser renderizadas em um filme (dados que são calculados em tempo real)
- Porém, o Blender oferece um recurso de ‘buferização’ de simulação = gravar os quadros gerados pela simulação como quadros-chaves na timeline para posterior utilização em um filme animado
- O processo para se realizar isso segue dois passos:
  1. Ativar o recurso de gravação de quadros no menu **Game | Record Animation**
  2. Executar a simulação (tecla P)

# Gravando Simulações Físicas

---

- Simulações físicas por se não podem ser renderizadas em um filme (dados que são calculados em tempo real)
- Porém, o Blender oferece um recurso de ‘buferização’ de simulação = gravar os quadros gerados pela simulação como quadros-chaves na timeline para posterior utilização em um filme animado
- O processo para se realizar isso segue dois passos:
  1. Ativar o recurso de gravação de quadros no menu **Game | Record Animation**
  2. Executar a simulação (tecla P)
- **IMPORTANTE!** a engine de física não encerrará a simulação até que a tecla **ESC** seja pressionada; ou seja, durante todo o tempo da simulação, o Blender estará gravando quadros. Portanto, encerre a simulação o mais cedo possível para evitar a geração de uma quantidade MUITO grande de quadros-chave

# Gravando Simulações Físicas



## *Collision Bounds*

---

- O processo de cálculo de colisões no Blender é realizado através de dois passos:

## *Collision Bounds*

---

- O processo de cálculo de colisões no Blender é realizado através de dois passos:

**Detecção** processo de se determinar quando dois objetos estão ocupando o mesmo lugar no espaço

## *Collision Bounds*

---

- O processo de cálculo de colisões no Blender é realizado através de dois passos:

**Detecção** processo de se determinar quando dois objetos estão ocupando o mesmo lugar no espaço

**Resposta** processo de se determinar as forças envolvidas durante a colisão e como estas afetam o movimento dos objetos envolvidos

## Collision Bounds

---

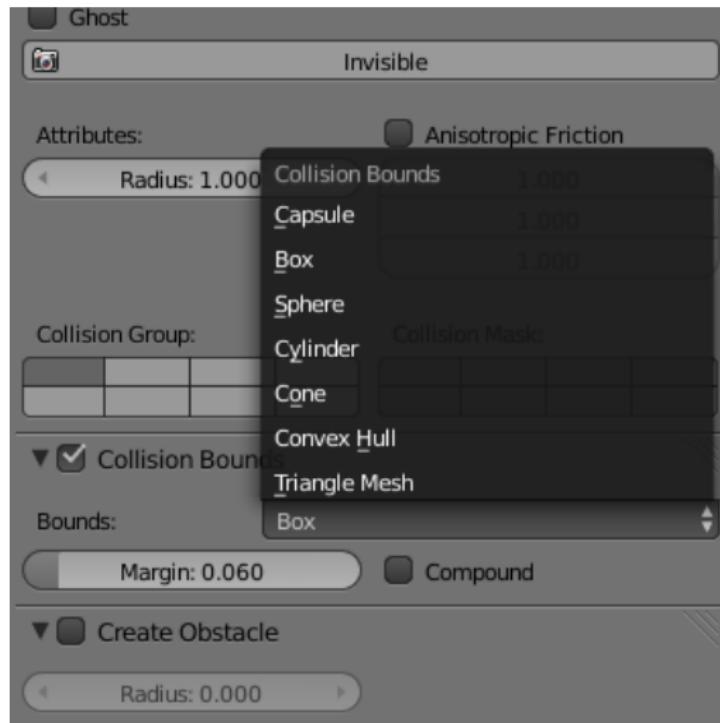
- O processo de cálculo de colisões no Blender é realizado através de dois passos:
  - Detecção** processo de se determinar quando dois objetos estão ocupando o mesmo lugar no espaço
  - Resposta** processo de se determinar as forças envolvidas durante a colisão e como estas afetam o movimento dos objetos envolvidos
- Dado que objetos são representados por malhas poligonais, uma processo de simplificação geométrica geralmente é utilizado para a etapa de detecção de colisões, denominado *collision bounds*

## Collision Bounds

---

- O processo de cálculo de colisões no Blender é realizado através de dois passos:
  - Detecção** processo de se determinar quando dois objetos estão ocupando o mesmo lugar no espaço
  - Resposta** processo de se determinar as forças envolvidas durante a colisão e como estas afetam o movimento dos objetos envolvidos
- Dado que objetos são representados por malhas poligonais, uma processo de simplificação geométrica geralmente é utilizado para a etapa de detecção de colisões, denominado *collision bounds*
- Um ‘volume de colisão’ é um objeto geométrico simples que envolve (ao nível lógico) a malha poligonal. O Blender tem suporte a vários tipos de volumes: *Capsule, Box, Sphere, Cylinder, Cone, Convex Hull, Triangle Mesh*

# Collision Bounds



## Criando Relações Dinâmicas entre Objetos

---

- **Relações dinâmicas** significa que dois objetos apresentarão algum tipo de restrição entre eles durante a simulação física

# Criando Relações Dinâmicas entre Objetos

---

- **Relações dinâmicas** significa que dois objetos apresentarão algum tipo de restrição entre eles durante a simulação física
- Tipos:

# Criando Relações Dinâmicas entre Objetos

---

- **Relações dinâmicas** significa que dois objetos apresentarão algum tipo de restrição entre eles durante a simulação física
- Tipos:
  - Dobradiças permite que um objeto *A* rotacione (em um eixo fixo) sobre um objeto *B*

# Criando Relações Dinâmicas entre Objetos

---

- **Relações dinâmicas** significa que dois objetos apresentarão algum tipo de restrição entre eles durante a simulação física
- Tipos:

**Dobradiças** permite que um objeto *A* rotacione (em um eixo fixo) sobre um objeto *B*

**Articulação** permite que um objeto *A* rotacione (em qualquer eixo) sobre um objeto *B*

# Criando Relações Dinâmicas entre Objetos

---

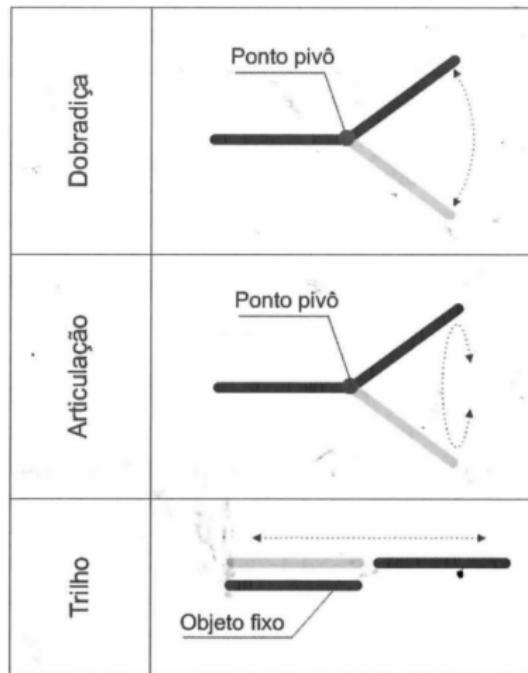
- **Relações dinâmicas** significa que dois objetos apresentarão algum tipo de restrição entre eles durante a simulação física
- Tipos:

**Dobradiças** permite que um objeto *A* rotacione (em um eixo fixo) sobre um objeto *B*

**Articulação** permite que um objeto *A* rotacione (em qualquer eixo) sobre um objeto *B*

**Trilho** permite que um objeto *A* deslize sobre um objeto *B*

# Criando Relações Dinâmicas entre Objetos



## Criando Uma Porta através de Relações Dinâmicas

---

1. Modelar uma cena que contenha: um piso, uma parede, uma porta uma rampa inclinada, e uma bola (ver figura abaixo)

# Criando Uma Porta através de Relações Dinâmicas

---

1. Modelar uma cena que contenha: um piso, uma parede, uma porta uma rampa inclinada, e uma bola (ver figura abaixo)
2. O piso, a parede e a rampa são objetos estáticos

# Criando Uma Porta através de Relações Dinâmicas

---

1. Modelar uma cena que contenha: um piso, uma parede, uma porta uma rampa inclinada, e uma bola (ver figura abaixo)
2. O piso, a parede e a rampa são objetos estáticos
3. A bola é um *Rigid Body* com massa igual a 10 (ou superior)

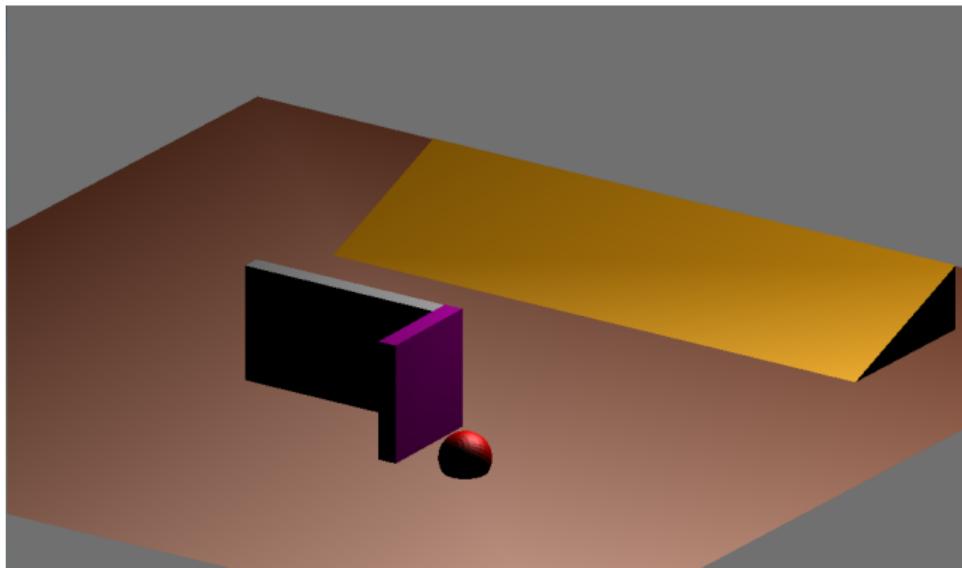
# Criando Uma Porta através de Relações Dinâmicas

---

1. Modelar uma cena que contenha: um piso, uma parede, uma porta uma rampa inclinada, e uma bola (ver figura abaixo)
2. O piso, a parede e a rampa são objetos estáticos
3. A bola é um *Rigid Body* com massa igual a 10 (ou superior)
4. A porta é um corpo rígido com uma **constraint** do tipo *Hinge* conectando-a à parede. Configure também a posição do pivot e o eixo de rotação (Blender sempre usa o eixo X como referência).

# Criando Uma Porta através de Relações Dinâmicas

---



# Sistemas de Forças

---

- Entende-se por **sistemas de forças** a um ambiente onde um objeto físico (rígido ou não) está sob ação de uma ou várias forças externas

# Sistemas de Forças

---

- Entende-se por **sistemas de forças** a um ambiente onde um objeto físico (rígido ou não) está sob ação de uma ou várias forças externas
- O processo de especificação de forças externas é feito através da evocação de rotinas (*Logic Bricks*) da BGE API a fim de especificar **regras de comportamento**

# Sistemas de Forças

---

- Entende-se por **sistemas de forças** a um ambiente onde um objeto físico (rígido ou não) está sob ação de uma ou várias forças externas
- O processo de especificação de forças externas é feito através da evocação de rotinas (*Logic Bricks*) da BGE API a fim de especificar **regras de comportamento**
- Uma regra de comportamento é definida com base em três elementos:

# Sistemas de Forças

---

- Entende-se por **sistemas de forças** a um ambiente onde um objeto físico (rígido ou não) está sob ação de uma ou várias forças externas
- O processo de especificação de forças externas é feito através da evocação de rotinas (*Logic Bricks*) da BGE API a fim de especificar **regras de comportamento**
- Uma regra de comportamento é definida com base em três elementos:  
**Controlador** é o elemento responsável por definir a execução da regra

# Sistemas de Forças

---

- Entende-se por **sistemas de forças** a um ambiente onde um objeto físico (rígido ou não) está sob ação de uma ou várias forças externas
- O processo de especificação de forças externas é feito através da evocação de rotinas (*Logic Bricks*) da BGE API a fim de especificar **regras de comportamento**
- Uma regra de comportamento é definida com base em três elementos:
  - Controlador** é o elemento responsável por definir a execução da regra
  - Sensor** é o elemento responsável por detectar eventos no ambiente

# Sistemas de Forças

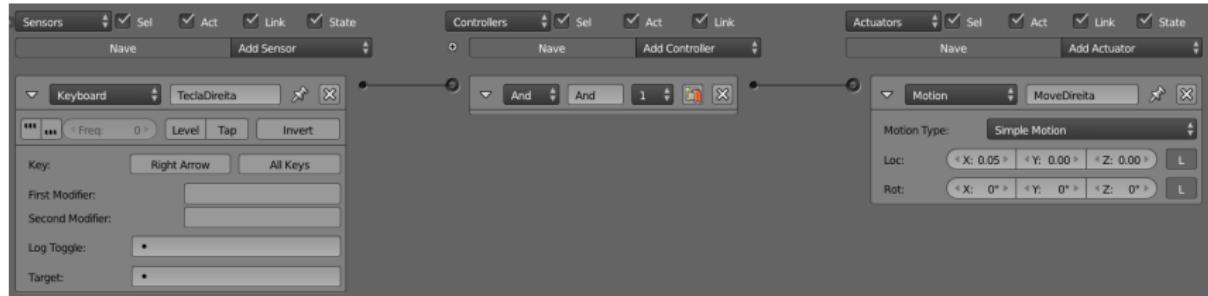
---

- Entende-se por **sistemas de forças** a um ambiente onde um objeto físico (rígido ou não) está sob ação de uma ou várias forças externas
- O processo de especificação de forças externas é feito através da evocação de rotinas (*Logic Bricks*) da BGE API a fim de especificar **regras de comportamento**
- Uma regra de comportamento é definida com base em três elementos:
  - Controlador** é o elemento responsável por definir a execução da regra
  - Sensor** é o elemento responsável por detectar eventos no ambiente
  - Actuator** é o elemento responsável por executar ações no ambiente

# Sistemas de Forças

---

- Entende-se por **sistemas de forças** a um ambiente onde um objeto físico (rígido ou não) está sob ação de uma ou várias forças externas
- O processo de especificação de forças externas é feito através da evocação de rotinas (*Logic Bricks*) da BGE API a fim de especificar **regras de comportamento**
- Uma regra de comportamento é definida com base em três elementos:
  - Controlador** é o elemento responsável por definir a execução da regra
  - Sensor** é o elemento responsável por detectar eventos no ambiente
  - Actuator** é o elemento responsável por executar ações no ambiente



# Sistemas de Forças

---

- Um atuador que nos é relevante para a disciplina é o *Motion* que, como o nome sugere, é o atuador responsável por definir movimento a objetos

# Sistemas de Forças

---

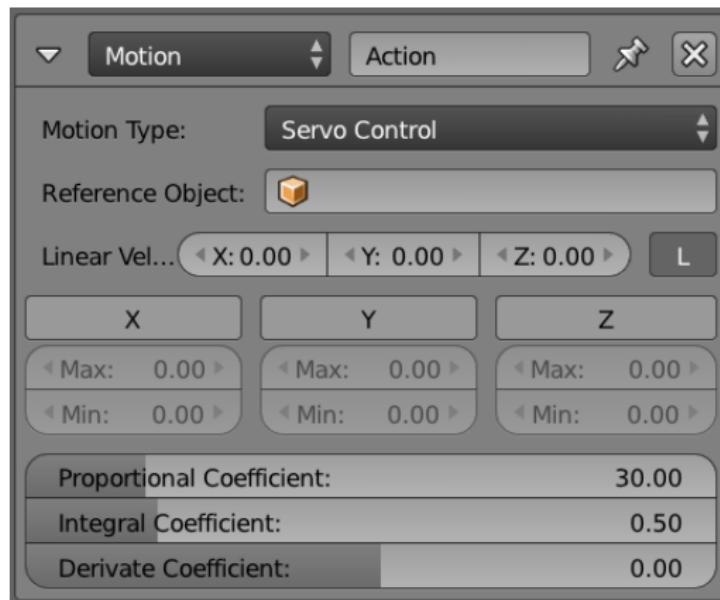
- Um atuador que nos é relevante para a disciplina é o *Motion* que, como o nome sugere, é o atuador responsável por definir movimento a objetos
- Este atuador apresenta-se em três tipos sendo o do tipo *Servo Control* aquele responsável por aplicar uma força sobre um objeto físico do tipo *rigid body*

# Sistemas de Forças

---

- Um atuador que nos é relevante para a disciplina é o *Motion* que, como o nome sugere, é o atuador responsável por definir movimento a objetos
- Este atuador apresenta-se em três tipos sendo o do tipo *Servo Control* aquele responsável por aplicar uma força sobre um objeto físico do tipo *rigid body*
- Uma força é definida como um vetor 3D denominado “*Linear Velocity*”

# Sistemas de Forças



## Addon: Fratura Celular

---

- Um *addon* bastante útil na animação de sistemas baseados em física é o **fratura celular** (*Cell Fracture*)

## Addon: Fratura Celular

---

- Um *addon* bastante útil na animação de sistemas baseados em física é o **fratura celular** (*Cell Fracture*)
- Para habilitar seu uso no Blender: menu **File | User Preferences...** na aba **Addons**, busque por “cell”; habilite o *addon* “Cell Fracture”

## Addon: Fratura Celular

---

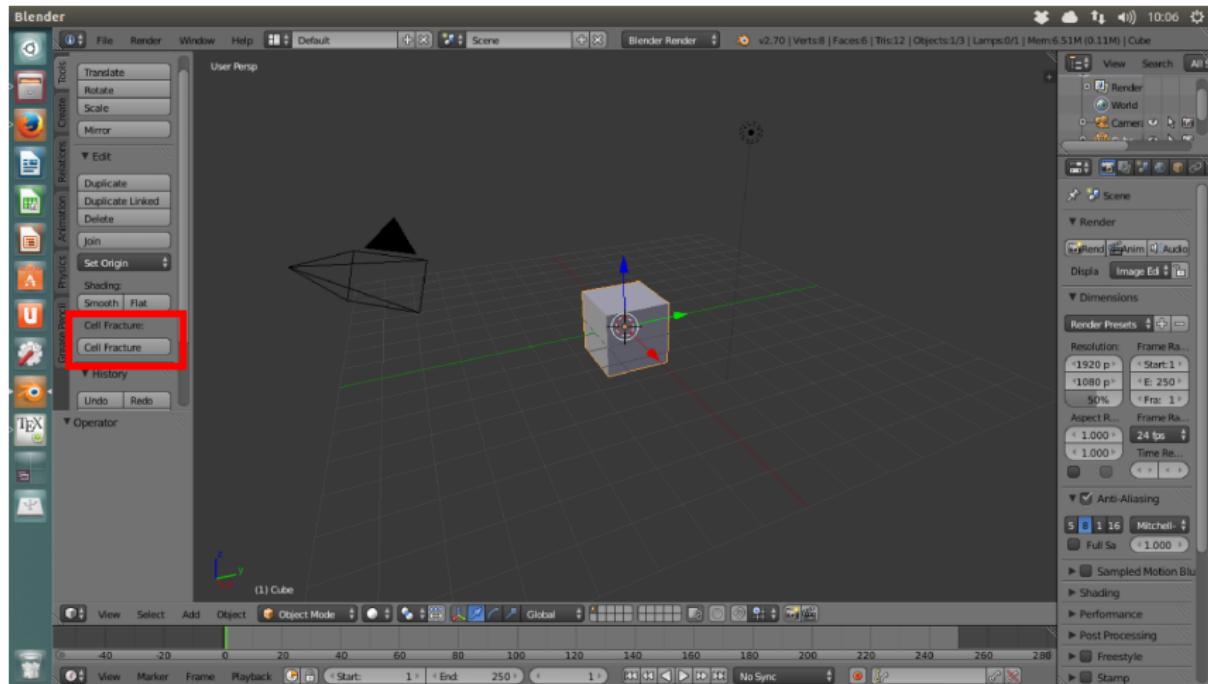
- Um *addon* bastante útil na animação de sistemas baseados em física é o **fratura celular** (*Cell Fracture*)
- Para habilitar seu uso no Blender: menu **File | User Preferences...** na aba **Addons**, busque por “cell”; habilite o *addon* “Cell Fracture”
- **DICA!** Para evitar ter que refazer esse passo toda vez, clique em “Save User Settings”

## Addon: Fratura Celular

---

- Um *addon* bastante útil na animação de sistemas baseados em física é o **fratura celular** (*Cell Fracture*)
- Para habilitar seu uso no Blender: menu **File | User Preferences...** na aba **Addons**, busque por “cell”; habilite o *addon* “Cell Fracture”
- **DICA!** Para evitar ter que refazer esse passo toda vez, clique em “Save User Settings”
- Agora na aba flutuante *Tools* na 3D View (tecla de atalho: **T**) aparecerá um novo botão nomeado **Cell Fracture**

# Addon: Fratura Celular



## Addon: Fratura Celular

---

- Para seu uso seleciona-se o objeto que se deseja “fraturar”

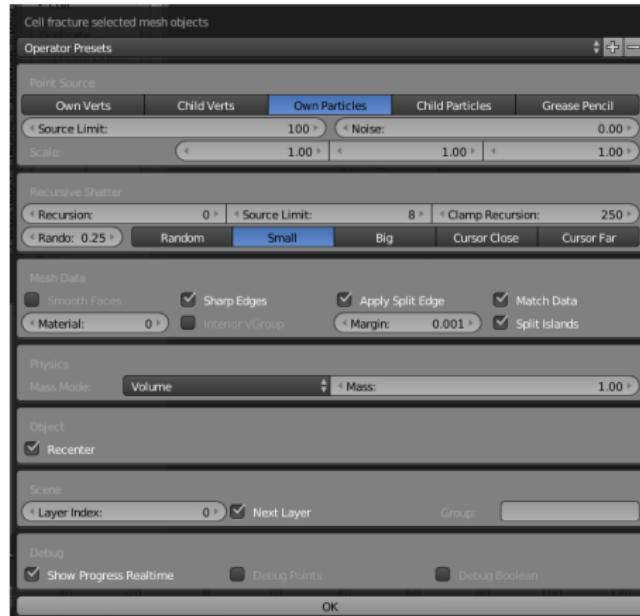
## Addon: Fratura Celular

---

- Para seu uso seleciona-se o objeto que se deseja “fraturar”
- Selecionando-se o botão *Cell Fracture* uma janela de opções é habilitada

# Addon: Fratura Celular

- Para seu uso seleciona-se o objeto que se deseja “fraturar”
- Selecionando-se o botão *Cell Fracture* uma janela de opções é habilitada



## Corpos Flexíveis

---

- Corpos flexíveis (ou maleáveis) são aqueles que podem sofrer deformação de seu volume quando submetidos à ação de forças externas

# Corpos Flexíveis

---

- Corpos flexíveis (ou maleáveis) são aqueles que podem sofrer deformação de seu volume quando submetidos à ação de forças externas
- Em Blender, este tipo de objeto é representado pela opção *Soft Body* na aba Physics da BGE

# Corpos Flexíveis

---

- Corpos flexíveis (ou maleáveis) são aqueles que podem sofrer deformação de seu volume quando submetidos à ação de forças externas
- Em Blender, este tipo de objeto é representado pela opção *Soft Body* na aba Physics da BGE
- Há uma série de opções que podem ser ajustadas para modificar o comportamento do corpo flexível. As mais comuns são:

## Corpos Flexíveis

---

- Corpos flexíveis (ou maleáveis) são aqueles que podem sofrer deformação de seu volume quando submetidos à ação de forças externas
- Em Blender, este tipo de objeto é representado pela opção *Soft Body* na aba Physics da BGE
- Há uma série de opções que podem ser ajustadas para modificar o comportamento do corpo flexível. As mais comuns são:
  - **Shape Match** e **Threshold** especificam que se (e quanto) o objeto deve tentar manter a forma original

# Corpos Flexíveis

---

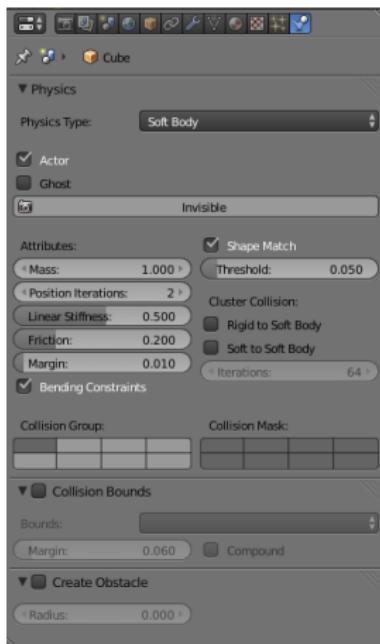
- Corpos flexíveis (ou maleáveis) são aqueles que podem sofrer deformação de seu volume quando submetidos à ação de forças externas
- Em Blender, este tipo de objeto é representado pela opção *Soft Body* na aba Physics da BGE
- Há uma série de opções que podem ser ajustadas para modificar o comportamento do corpo flexível. As mais comuns são:
  - **Shape Match** e **Threshold** especificam que se (e quanto) o objeto deve tentar manter a forma original
  - **Position iterations** especifica o quanto preciso será o processo de simulação de forças de deformação em relação à vizinhança de cada vértice (quanto maior esse valor, mais vértices vizinhos influenciarão o resultado da simulação)

# Corpos Flexíveis

---

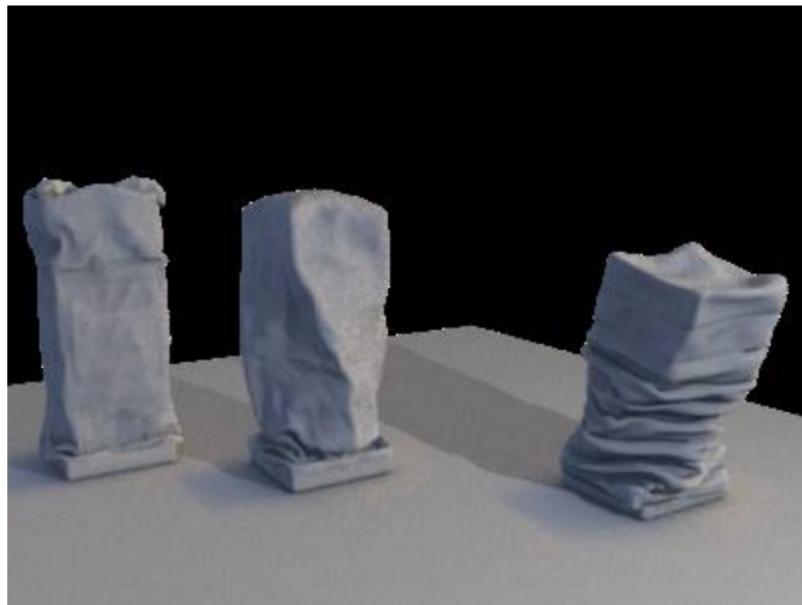
- Corpos flexíveis (ou maleáveis) são aqueles que podem sofrer deformação de seu volume quando submetidos à ação de forças externas
- Em Blender, este tipo de objeto é representado pela opção *Soft Body* na aba Physics da BGE
- Há uma série de opções que podem ser ajustadas para modificar o comportamento do corpo flexível. As mais comuns são:
  - **Shape Match** e **Threshold** especificam que se (e quanto) o objeto deve tentar manter a forma original
  - **Position iterations** especifica o quão preciso será o processo de simulação de forças de deformação em relação à vizinhança de cada vértice (quanto maior esse valor, mais vértices vizinhos influenciarão o resultado da simulação)
  - **Linear Stiffness** define o grau de flexibilidade do material

# Corpos Flexíveis



# Corpos Flexíveis

---



## Corpos Flexíveis - Tecidos

---

- Tecidos são um caso particular de animação de corpos flexíveis, com a diferença que frequentemente são aplicados a superfícies planares (ao invés de volumes)

## Corpos Flexíveis - Tecidos

---

- Tecidos são um caso particular de animação de corpos flexíveis, com a diferença que frequentemente são aplicados a superfícies planares (ao invés de volumes)
- Para se configurar uma animação de tecidos, utiliza-se a **Blender Render** ao invés da *Blender Game*

## Corpos Flexíveis - Tecidos

---

- Tecidos são um caso particular de animação de corpos flexíveis, com a diferença que frequentemente são aplicados a superfícies planares (ao invés de volumes)
- Para se configurar uma animação de tecidos, utiliza-se a **Blender Render** ao invés da *Blender Game*
- Com o objeto selecionado, define-se (na aba *Physics*) a opção **Cloth**. As principais opções para simulação de tecidos são:

## Corpos Flexíveis - Tecidos

---

- Tecidos são um caso particular de animação de corpos flexíveis, com a diferença que frequentemente são aplicados a superfícies planares (ao invés de volumes)
- Para se configurar uma animação de tecidos, utiliza-se a **Blender Render** ao invés da *Blender Game*
- Com o objeto selecionado, define-se (na aba *Physics*) a opção **Cloth**. As principais opções para simulação de tecidos são:

**Presets** apresenta um conjunto de parâmetros de tecidos pré-definidos: Algodão = *Cotton*; Jeans = *Denim*; Couro = *Leather*; Borracha = *Rubber* e Seda = *Silk*

# Corpos Flexíveis - Tecidos

---

- Tecidos são um caso particular de animação de corpos flexíveis, com a diferença que frequentemente são aplicados a superfícies planares (ao invés de volumes)
- Para se configurar uma animação de tecidos, utiliza-se a **Blender Render** ao invés da *Blender Game*
- Com o objeto selecionado, define-se (na aba *Physics*) a opção **Cloth**. As principais opções para simulação de tecidos são:

**Presets** apresenta um conjunto de parâmetros de tecidos pré-definidos: Algodão = *Cotton*; Jeans = *Denim*; Couro = *Leather*; Borracha = *Rubber* e Seda = *Silk*

**Pinning** permite especificar que um dado grupo de vértices (*Vertex Group*) criado previamente na aba **Data** não deve sofrer os efeitos da deformação física (ficam presos à sua posição inicial)

# Corpos Flexíveis - Tecidos

---

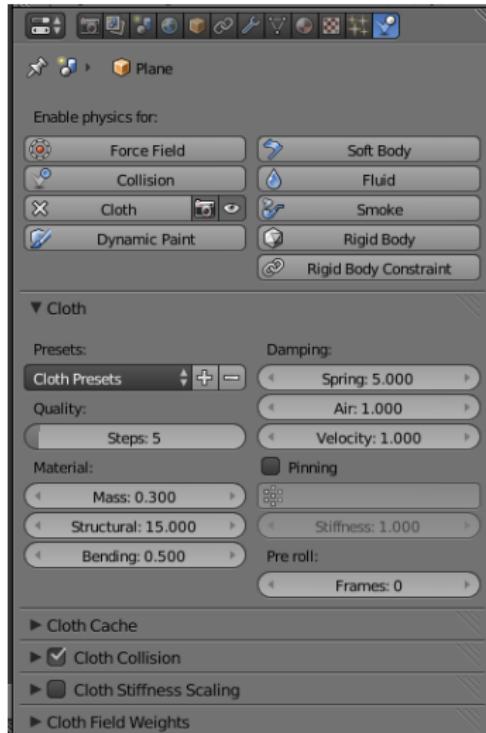
- Tecidos são um caso particular de animação de corpos flexíveis, com a diferença que frequentemente são aplicados a superfícies planares (ao invés de volumes)
- Para se configurar uma animação de tecidos, utiliza-se a **Blender Render** ao invés da *Blender Game*
- Com o objeto selecionado, define-se (na aba *Physics*) a opção **Cloth**. As principais opções para simulação de tecidos são:

**Presets** apresenta um conjunto de parâmetros de tecidos pré-definidos: Algodão = *Cotton*; Jeans = *Denim*; Couro = *Leather*; Borracha = *Rubber* e Seda = *Silk*

**Pinning** permite especificar que um dado grupo de vértices (*Vertex Group*) criado previamente na aba **Data** não deve sofrer os efeitos da deformação física (ficam presos à sua posição inicial)

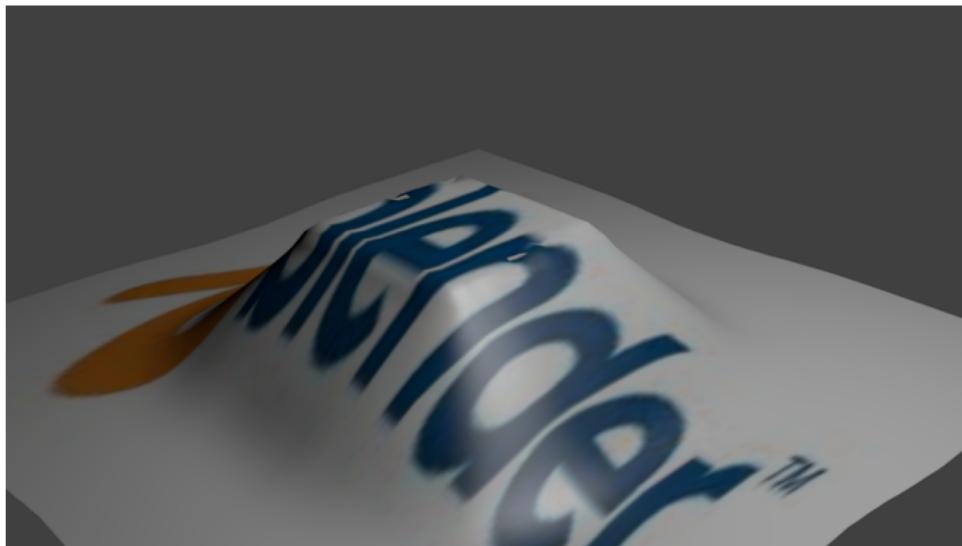
Particularmente útil para animar tecidos presos a outros objetos da animação (ex.: bandeira tremulando num mastro ou capa de super-herói presa ao pescoço do personagem)

# Corpos Flexíveis - Tecidos



## Corpos Flexíveis - Tecidos

---



# Corpos Flexíveis - Simulação de Campos de Forças Externas

---

- Outro elemento que pode ser adicionado durante simulações de forças físicas (como tecidos e sistemas de partículas) é o uso de 'campos de forças' externas (*Force Fields*)

# Corpos Flexíveis - Simulação de Campos de Forças Externas

---

- Outro elemento que pode ser adicionado durante simulações de forças físicas (como tecidos e sistemas de partículas) é o uso de 'campos de forças' externas (*Force Fields*)
- O Blender implementa uma série dessas forças: *Force*, *Wind*, *Vortex*, *Magnetic*, *Harmonic*, *Charge*, *Lennard-Jones*, *Texture*, *Curve*, *Boid*

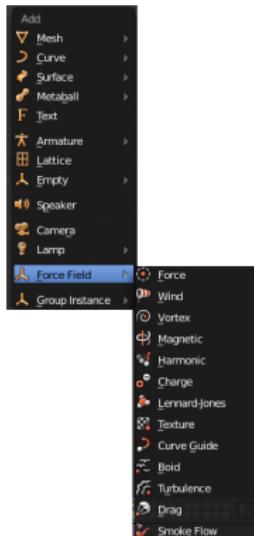
# Corpos Flexíveis - Simulação de Campos de Forças Externas

---

- Outro elemento que pode ser adicionado durante simulações de forças físicas (como tecidos e sistemas de partículas) é o uso de 'campos de forças' externas (*Force Fields*)
- O Blender implementa uma série dessas forças: *Force*, *Wind*, *Vortex*, *Magnetic*, *Harmonic*, *Charge*, *Lennard-Jones*, *Texture*, *Curve*, *Boid*
- Mais detalhes sobre a utilização de campos de forças podem ser encontrados em [http://wiki.blender.org/index.php/Doc:2.6/Manual/Physics/Force\\_Fields](http://wiki.blender.org/index.php/Doc:2.6/Manual/Physics/Force_Fields)

# Corpos Flexíveis - Simulação de Campos de Forças Externas

---



# Exercício Prático Individual - EPI #4

---

**Nome:** *Simulação de Forças Físicas*

**Objetivo:** construção de uma máquina de Rube Golbert capaz de simular os modelos de forças físicas estudados

- Enunciado:**
1. projetar uma máquina que seja capaz de ilustrar **todos** os aspectos físicos estudados
  2. produzir uma sequência de quadros através de *straight-ahead* e armazenar os quadros-chaves resultantes em uma timeline, à qual possa ser posteriormente editada
  3. compilar um filme resultante

- Requisitos:**
- Os seguintes modelos físicos devem ser ilustrados na animação resultante:  
quedas de corpos, colisões (detecção e tratamento), lançamento oblíquo, corpos rígidos, corpos flexíveis, tecidos, campos de forças
  - Devem ser utilizadas as técnicas vistas em aula: *straight-ahead* + armazenamento de quadros-chaves
  - Sugestão: animar elementos como luzes e câmeras
  - Pode conter trilha sonora e/ou efeitos sonoros
  - Duração mínima: 30 segundos
  - Deve apresentar tela de créditos iniciais e finais (o tempo de apresentação dos créditos não contabiliza para o tempo da animação)

**Prazo para Entrega:** 2 semanas (filme + estória + slides ilustrando o processo)

**Peso:** 15% da média semestral

# Animação de Grupos

- Sistemas de Partículas
- Controle de Manobras
- Fluídos
- Partículas Estáticas

# Controle de Grupos

---

- Até o momento estudamos com animar cada ator individualmente em uma cena, e então sincronizar suas ações de forma a produzirem uma encenação coletiva de uma estória

# Controle de Grupos

---

- Até o momento estudamos com animar cada ator individualmente em uma cena, e então sincronizar suas ações de forma a produzirem uma encenação coletiva de uma estória
- **Problema:** como animar uma cena caso o número de atores presentes seja proibitivo de serem animados individualmente (centenas ou milhares ou até milhões) ?

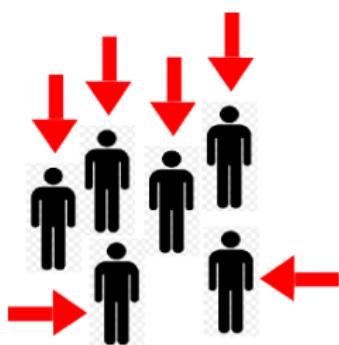
# Controle de Grupos

---

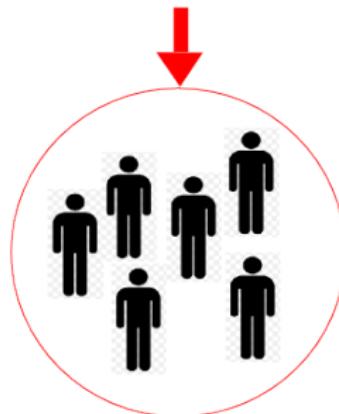
- Até o momento estudamos com animar cada ator individualmente em uma cena, e então sincronizar suas ações de forma a produzirem uma encenação coletiva de uma estória
- **Problema:** como animar uma cena caso o número de atores presentes seja proibitivo de serem animados individualmente (centenas ou milhares ou até milhões) ?
- *Resposta:* através de agrupá-los em agentes abstratos que podem então serem animados individualmente da maneira convencional, retransmitindo os comandos para cada ator pertencente ao grupo

# Controle de Grupos

Comandos de  
Animação



Comandos de  
Animação



# Controle de Grupos

---



**MASSIVE**  
SOFTWARE

 Artificial Life for Animation™

# Controle de Grupos

---

Tipos de Grupos de Atores:

# Controle de Grupos

---

Tipos de Grupos de Atores:

**Sistemas de Partículas** representam indivíduos simples, sem autonomia, animáveis através de comportamentos genéricos (como simulações físicas). Geralmente utilizados para representação de sistemas amorfos como fenômenos naturais (chuva, neve, fumaça, fogo, etc.)

# Controle de Grupos

---

Tipos de Grupos de Atores:

**Sistemas de Partículas** representam indivíduos simples, sem autonomia, animáveis através de comportamentos genéricos (como simulações físicas).

Geralmente utilizados para representação de sistemas amorfos como fenômenos naturais (chuva, neve, fumaça, fogo, etc.)

**Controle de Manobras** (ou *Steering Behaviors*) representam indivíduos com um certo grau de autonomia, animáveis através de regras de restrições de movimento. Geralmente utilizados para comportamento de bandos como cardumes, enxames, multidões, etc.

# Controle de Grupos

---

## Tipos de Grupos de Atores:

**Sistemas de Partículas** representam indivíduos simples, sem autonomia, animáveis através de comportamentos genéricos (como simulações físicas). Geralmente utilizados para representação de sistemas amorfos como fenômenos naturais (chuva, neve, fumaça, fogo, etc.)

**Controle de Manobras** (ou *Steering Behaviors*) representam indivíduos com um certo grau de autonomia, animáveis através de regras de restrições de movimento. Geralmente utilizados para comportamento de bandos como cardumes, enxames, multidões, etc.

**Comportamento Autônomo** representam indivíduos autonomamente capazes de perceber o ambiente a seu redor (mudanças no cenário) e decidir (em função de conhecimentos prévios) a melhor ação a tomar, desta forma se “auto-animando” sem a intervenção direta do animador humano. Geralmente utilizados para autoria de **personagens sintéticos** como *Non-Player Characters*, assistentes virtuais, atores digitais autônomos, etc.

# Controle de Grupos

---

<b>Tipo</b>	<b># Elementos</b>	<b>Simulação Física</b>	<b>Inteligência</b>
<b>Partículas</b>	Milhões	Muita (com ambiente)	Nenhuma
<b>Manobras</b>	Centenas	Alguma (com ambiente/Indiv.)	Limitada
<b>Personagens</b>	Dezenas	Pouca	Muita

# Controle de Grupos

---

- A principal motivação para se trabalhar com controle de grupos é a construção do chamado **comportamento emergente**

# Controle de Grupos

---

- A principal motivação para se trabalhar com controle de grupos é a construção do chamado **comportamento emergente**
- Produz-se indivíduos com comportamentos específicos simples e, quando apresentados em grupos, o resultado é um comportamento coletivo característico que **emerge** das interações entre os elementos do grupo (sem a necessidade de uma especificação explícita para esse comportamento)

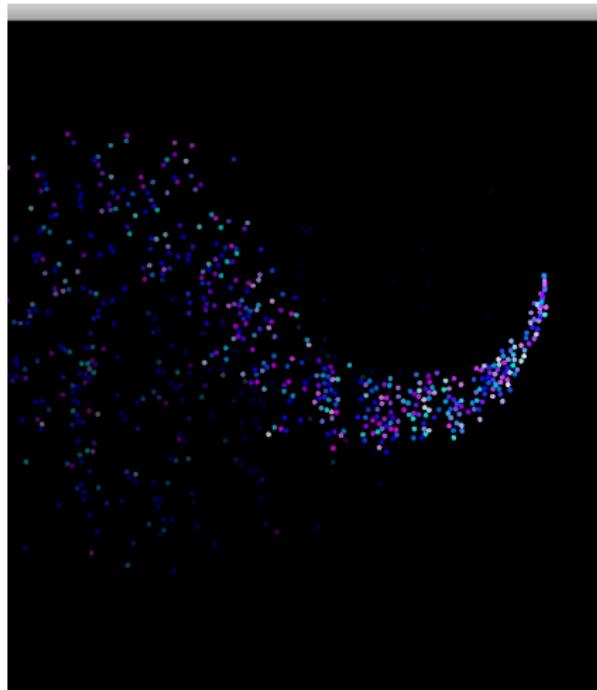
# Controle de Grupos

---

- A principal motivação para se trabalhar com controle de grupos é a construção do chamado **comportamento emergente**
- Produz-se indivíduos com comportamentos específicos simples e, quando apresentados em grupos, o resultado é um comportamento coletivo característico que **emerge** das interações entre os elementos do grupo (sem a necessidade de uma especificação explícita para esse comportamento)
- Imitar comportamentos emergentes específicos requer experiência, prática e uma enorme capacidade de abstração e/ou investigação da natureza ou da sociedade que se quer imitar

# Controle de Grupos

---



# Sistemas de Partículas

---

- Sistemas de Partículas são grandes grupos de indivíduos (chamados *partículas*), aos quais são feitas muitas restrições/limitações para determinação de sua movimentação e renderização = partículas são extremamente simples!

# Sistemas de Partículas

---

- Sistemas de Partículas são grandes grupos de indivíduos (chamados *partículas*), aos quais são feitas muitas restrições/limitações para determinação de sua movimentação e renderização = partículas são extremamente simples!
- Limitações comuns às partículas:

# Sistemas de Partículas

---

- Sistemas de Partículas são grandes grupos de indivíduos (chamados *partículas*), aos quais são feitas muitas restrições/limitações para determinação de sua movimentação e renderização = partículas são extremamente simples!
- Limitações comuns às partículas:
  - partículas não colidem entre si (apenas contra o ambiente)

# Sistemas de Partículas

---

- Sistemas de Partículas são grandes grupos de indivíduos (chamados *partículas*), aos quais são feitas muitas restrições/limitações para determinação de sua movimentação e renderização = partículas são extremamente simples!
- Limitações comuns às partículas:
  - partículas não colidem entre si (apenas contra o ambiente)
  - partículas não produzem sombra umas nas outras (apenas o grupo no ambiente)

# Sistemas de Partículas

---

- Sistemas de Partículas são grandes grupos de indivíduos (chamados *partículas*), aos quais são feitas muitas restrições/limitações para determinação de sua movimentação e renderização = partículas são extremamente simples!
- Limitações comuns às partículas:
  - partículas não colidem entre si (apenas contra o ambiente)
  - partículas não produzem sombra umas nas outras (apenas o grupo no ambiente)
  - partículas não refletem luz

# Sistemas de Partículas

---

- Sistemas de Partículas são grandes grupos de indivíduos (chamados *partículas*), aos quais são feitas muitas restrições/limitações para determinação de sua movimentação e renderização = partículas são extremamente simples!
- Limitações comuns às partículas:
  - partículas não colidem entre si (apenas contra o ambiente)
  - partículas não produzem sombra umas nas outras (apenas o grupo no ambiente)
  - partículas não refletem luz
  - partículas geralmente tem um tempo de vida definido

# Sistemas de Partículas

---

- O processo para construção e animação de um sistema de partículas é:

# Sistemas de Partículas

---

- O processo para construção e animação de um sistema de partículas é:
  1. Criar novas partículas a cada frame da animação e adicioná-las ao sistema de partículas

# Sistemas de Partículas

---

- O processo para construção e animação de um sistema de partículas é:
  1. Criar novas partículas a cada frame da animação e adicioná-las ao sistema de partículas
  2. Gerar condições iniciais estocásticas para as novas partículas recém-criadas

# Sistemas de Partículas

---

- O processo para construção e animação de um sistema de partículas é:
  1. Criar novas partículas a cada frame da animação e adicioná-las ao sistema de partículas
  2. Gerar condições iniciais estocásticas para as novas partículas recém-criadas
  3. Eliminar (do sistema) as partículas que tenham excedido seu tempo de vida

# Sistemas de Partículas

---

- O processo para construção e animação de um sistema de partículas é:
  1. Criar novas partículas a cada frame da animação e adicioná-las ao sistema de partículas
  2. Gerar condições iniciais estocásticas para as novas partículas recém-criadas
  3. Eliminar (do sistema) as partículas que tenham excedido seu tempo de vida
  4. Animar as partículas ainda presentes no sistema de partículas

# Sistemas de Partículas

---

- O processo para construção e animação de um sistema de partículas é:
  1. Criar novas partículas a cada frame da animação e adicioná-las ao sistema de partículas
  2. Gerar condições iniciais estocásticas para as novas partículas recém-criadas
  3. Eliminar (do sistema) as partículas que tenham excedido seu tempo de vida
  4. Animar as partículas ainda presentes no sistema de partículas
  5. Renderizar cada partícula do sistema de partículas

# Sistemas de Partículas

---

- Uma partícula é geralmente definida como um conjunto de atributos físicos (massa, posição, velocidade, aceleração, tempo de vida, etc.) e visuais (cor, textura, transparência, forma geométrica, etc.)

# Sistemas de Partículas

---

- Uma partícula é geralmente definida como um conjunto de atributos físicos (massa, posição, velocidade, aceleração, tempo de vida, etc.) e visuais (cor, textura, transparência, forma geométrica, etc.)
- Exemplo de partículas: primitivas geométricas (pontos, linhas, etc.), sólidos geométricos (cubos, esferas, cones, etc.), **Billboards** (planos texturizados cuja orientação se alinha automaticamente com a câmera = impostores 3D)

# Sistemas de Partículas

---

- Uma partícula é geralmente definida como um conjunto de atributos físicos (massa, posição, velocidade, aceleração, tempo de vida, etc.) e visuais (cor, textura, transparência, forma geométrica, etc.)
- Exemplo de partículas: primitivas geométricas (pontos, linhas, etc.), sólidos geométricos (cubos, esferas, cones, etc.), **Billboards** (planos texturizados cuja orientação se alinha automaticamente com a câmera = impostores 3D)
- O processo de criação de novas partículas é denominado **emissão** de partículas e é realizado a partir de uma abordagem estocásticas (aleatório com critérios controlados). Um objeto especial (*Emissor de Partículas*) é o responsável por controlar esse processo

# Sistemas de Partículas

---

- Uma partícula é geralmente definida como um conjunto de atributos físicos (massa, posição, velocidade, aceleração, tempo de vida, etc.) e visuais (cor, textura, transparência, forma geométrica, etc.)
- Exemplo de partículas: primitivas geométricas (pontos, linhas, etc.), sólidos geométricos (cubos, esferas, cones, etc.), **Billboards** (planos texturizados cuja orientação se alinha automaticamente com a câmera = impostores 3D)
- O processo de criação de novas partículas é denominado **emissão** de partículas e é realizado a partir de uma abordagem estocásticas (aleatório com critérios controlados). Um objeto especial (*Emissor de Partículas*) é o responsável por controlar esse processo  
Detalhe: um emissor de partículas pode também ser uma partícula!

# Sistemas de Partículas

---

- Uma partícula é geralmente definida como um conjunto de atributos físicos (massa, posição, velocidade, aceleração, tempo de vida, etc.) e visuais (cor, textura, transparência, forma geométrica, etc.)
- Exemplo de partículas: primitivas geométricas (pontos, linhas, etc.), sólidos geométricos (cubos, esferas, cones, etc.), **Billboards** (planos texturizados cuja orientação se alinha automaticamente com a câmera = impostores 3D)
- O processo de criação de novas partículas é denominado **emissão** de partículas e é realizado a partir de uma abordagem estocásticas (aleatório com critérios controlados). Um objeto especial (*Emissor de Partículas*) é o responsável por controlar esse processo  
Detalhe: um emissor de partículas pode também ser uma partícula!
- Exemplos de emissores: polígonos (plano, círculo, triângulo, linha, ponto); volumes (cubo, esfera, cone, cilindro, toróide, etc.)

# Sistemas de Partículas

---

- Outro aspecto importante a ser especificado para um sistema de partículas (e não para partículas individuais) é o **comportamento de grupo**

# Sistemas de Partículas

---

- Outro aspecto importante a ser especificado para um sistema de partículas (e não para partículas individuais) é o **comportamento de grupo**
- Ou seja, especificar o conjunto de forças que serão aplicadas à cada partícula do sistema a cada quadro para se produzir a animação resultante

# Sistemas de Partículas

---

- Outro aspecto importante a ser especificado para um sistema de partículas (e não para partículas individuais) é o **comportamento de grupo**
- Ou seja, especificar o conjunto de forças que serão aplicadas à cada partícula do sistema a cada quadro para se produzir a animação resultante
- Típicos comportamento são aqueles já estudados no capítulo de simulações físicas: arremessos, quedas, etc.

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

Number quantidade de partículas do sistema

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

**Number** quantidade de partículas do sistema

**Start/End** período no qual novas partículas serão emitidas

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

Number    quantidade de partículas do sistema  
Start/End    período no qual novas partículas serão emitidas  
Lifetime    tempo de vida de cada nova partícula

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo

**Emitter Object** indica a velocidade inicial das partículas (direção e força)

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo  
**Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo
  - Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas
  - Newtonian** as partículas serão animadas de acordo com as Leis de Newton

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo  
**Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

<b>Newtonian</b>	as partículas serão animadas de acordo com as Leis de Newton
<b>Keyed</b>	cada partícula já apresenta uma animação de quadros-chave que deve ser reproduzida durante a animação do sistema de partículas

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo  
**Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

<b>Newtonian</b>	as partículas serão animadas de acordo com as Leis de Newton
<b>Keyed</b>	cada partículas já apresenta uma animação de quadros-chave que deve ser reproduzida durante a animação do sistema de partículas
<b>Boids</b>	animação das partículas através de comportamentos de manobra ( <i>Steering Behaviors</i> )

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo  
**Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

<b>Newtonian</b>	as partículas serão animadas de acordo com as Leis de Newton
<b>Keyed</b>	cada partículas já apresenta uma animação de quadros-chave que deve ser reproduzida durante a animação do sistema de partículas
<b>Boids</b>	animação das partículas através de comportamentos de manobra ( <i>Steering Behaviors</i> )
<b>Fluids</b>	animação das partículas através de simulação de fluidos

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo  
**Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

<b>Newtonian</b>	as partículas serão animadas de acordo com as Leis de Newton
<b>Keyed</b>	cada partículas já apresenta uma animação de quadros-chave que deve ser reproduzida durante a animação do sistema de partículas
<b>Boids</b>	animação das partículas através de comportamentos de manobra ( <i>Steering Behaviors</i> )
<b>Fluids</b>	animação das partículas através de simulação de fluidos

6. Em **Render** especifica-se como renderizar as partículas

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo  
**Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

<b>Newtonian</b>	as partículas serão animadas de acordo com as Leis de Newton
<b>Keyed</b>	cada partículas já apresenta uma animação de quadros-chave que deve ser reproduzida durante a animação do sistema de partículas
<b>Boids</b>	animação das partículas através de comportamentos de manobra ( <i>Steering Behaviors</i> )
<b>Fluids</b>	animação das partículas através de simulação de fluidos

6. Em **Render** especifica-se como renderizar as partículas  
**Halo** cada partícula é representada como um objeto brilhante

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo
  - Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

<b>Newtonian</b>	as partículas serão animadas de acordo com as Leis de Newton
<b>Keyed</b>	cada partículas já apresenta uma animação de quadros-chave que deve ser reproduzida durante a animação do sistema de partículas
<b>Boids</b>	animação das partículas através de comportamentos de manobra ( <i>Steering Behaviors</i> )
<b>Fluids</b>	animação das partículas através de simulação de fluidos

6. Em **Render** especifica-se como renderizar as partículas

<b>Halo</b>	cada partícula é representada como um objeto brilhante
<b>Line</b>	cada partícula é representada como um segmento de reta

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo  
**Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

<b>Newtonian</b>	as partículas serão animadas de acordo com as Leis de Newton
<b>Keyed</b>	cada partículas já apresenta uma animação de quadros-chave que deve ser reproduzida durante a animação do sistema de partículas
<b>Boids</b>	animação das partículas através de comportamentos de manobra ( <i>Steering Behaviors</i> )
<b>Fluids</b>	animação das partículas através de simulação de fluidos

6. Em **Render** especifica-se como renderizar as partículas  
**Halo** cada partícula é representada como um objeto brilhante  
**Line** cada partícula é representada como um segmento de reta  
**Path** utilizada para partículas do tipo estáticas ou Keyed

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo  
**Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

<b>Newtonian</b>	as partículas serão animadas de acordo com as Leis de Newton
<b>Keyed</b>	cada partículas já apresenta uma animação de quadros-chave que deve ser reproduzida durante a animação do sistema de partículas
<b>Boids</b>	animação das partículas através de comportamentos de manobra ( <i>Steering Behaviors</i> )
<b>Fluids</b>	animação das partículas através de simulação de fluidos

6. Em **Render** especifica-se como renderizar as partículas

<b>Halo</b>	cada partícula é representada como um objeto brilhante
<b>Line</b>	cada partícula é representada como um segmento de reta
<b>Path</b>	utilizada para partículas do tipo estáticas ou Keyed
<b>Object</b>	cada partícula é um objeto pré-definido (modelo Blender)

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo  
**Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

<b>Newtonian</b>	as partículas serão animadas de acordo com as Leis de Newton
<b>Keyed</b>	cada partículas já apresenta uma animação de quadros-chave que deve ser reproduzida durante a animação do sistema de partículas
<b>Boids</b>	animação das partículas através de comportamentos de manobra ( <i>Steering Behaviors</i> )
<b>Fluids</b>	animação das partículas através de simulação de fluidos

6. Em **Render** especifica-se como renderizar as partículas

<b>Halo</b>	cada partícula é representada como um objeto brilhante
<b>Line</b>	cada partícula é representada como um segmento de reta
<b>Path</b>	utilizada para partículas do tipo estáticas ou Keyed
<b>Object</b>	cada partícula é um objeto pré-definido (modelo Blender)
<b>Group</b>	permite a composição de múltiplos tipos de partículas em um único sistema

# Sistemas de Partículas em Blender

---

1. Cria-se o objeto que servirá como emissor de partículas
2. Associa-se a este objeto, na aba *Particles*, um componente *Particle System* do tipo *Emitter* com um dado valor inicial de semente estocástica (*Seed*)
3. Em **Emission** especifica-se os parâmetros para nascimento de novas partículas no sistema

<b>Number</b>	quantidade de partículas do sistema
<b>Start/End</b>	período no qual novas partículas serão emitidas
<b>Lifetime</b>	tempo de vida de cada nova partícula
<b>Random</b>	fator de aleatoriedade para geração do tempo de vida das partículas
<b>Emit from</b>	indica a posição inicial das novas partículas

4. Em **Velocity** é especificado o comportamento das partículas no tempo  
**Emitter Object** indica a velocidade inicial das partículas (direção e força)
5. Em **Physics** é determinado o comportamento físico a ser utilizado durante o processo de simulação física (animação) das partículas

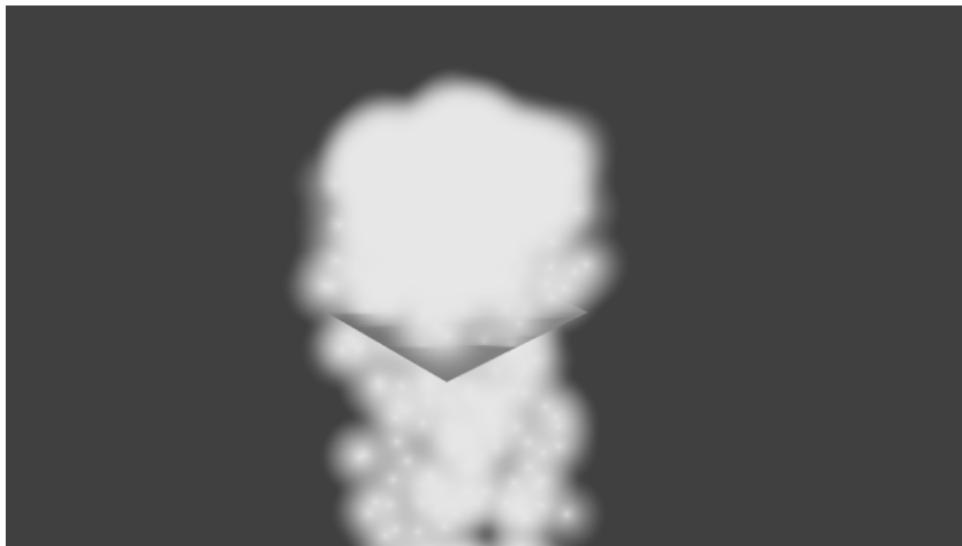
<b>Newtonian</b>	as partículas serão animadas de acordo com as Leis de Newton
<b>Keyed</b>	cada partículas já apresenta uma animação de quadros-chave que deve ser reproduzida durante a animação do sistema de partículas
<b>Boids</b>	animação das partículas através de comportamentos de manobra ( <i>Steering Behaviors</i> )
<b>Fluids</b>	animação das partículas através de simulação de fluidos

6. Em **Render** especifica-se como renderizar as partículas

<b>Halo</b>	cada partícula é representada como um objeto brilhante
<b>Line</b>	cada partícula é representada como um segmento de reta
<b>Path</b>	utilizada para partículas do tipo estáticas ou Keyed
<b>Object</b>	cada partícula é um objeto pré-definido (modelo Blender)
<b>Group</b>	permite a composição de múltiplos tipos de partículas em um único sistema
<b>Billboard</b>	partículas do tipo impostores 3D

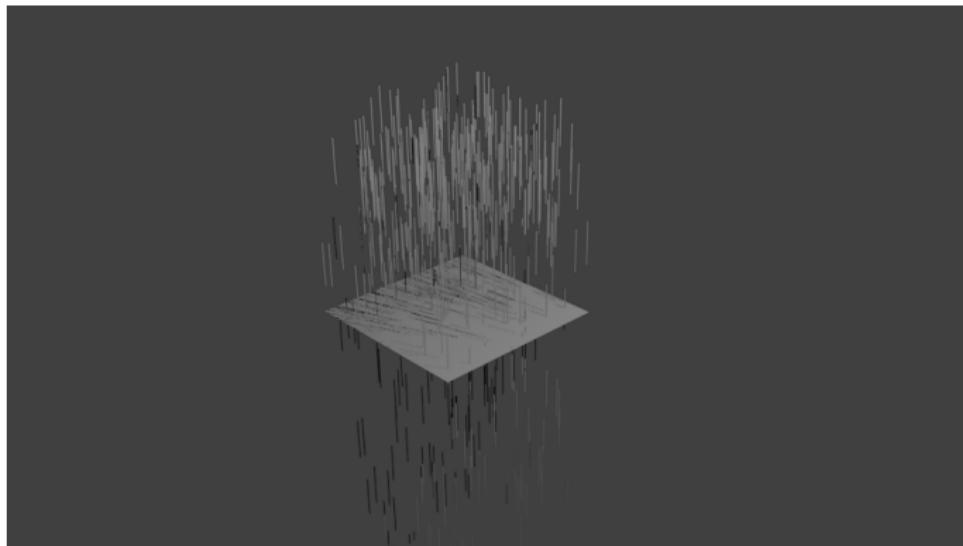
# Sistemas de Partículas em Blender

---



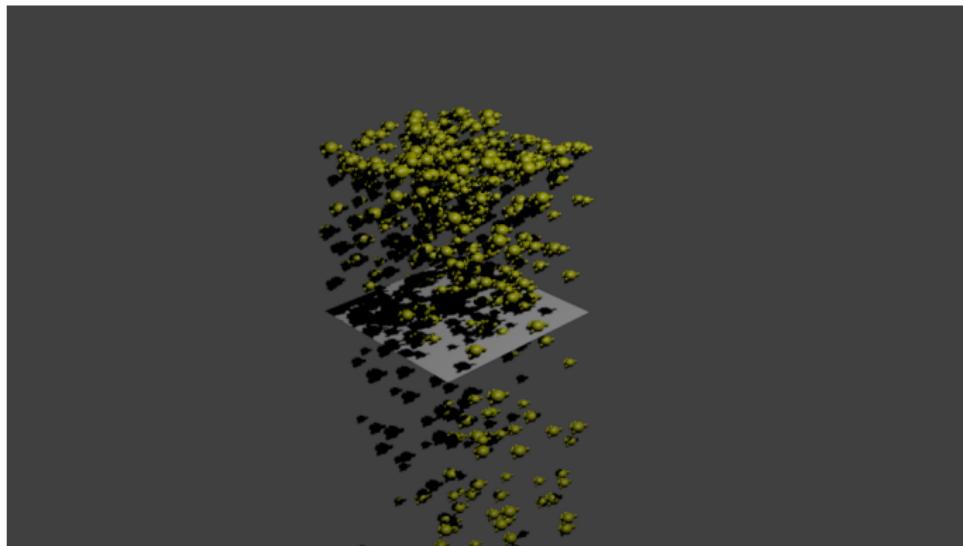
# Sistemas de Partículas em Blender

---



# Sistemas de Partículas em Blender

---



# Grupos de Partículas

---

- Grupos de partículas são úteis quando se quer animar diferentes tipos de partículas em um mesmo sistema de partículas

# Grupos de Partículas

---

- Grupos de partículas são úteis quando se quer animar diferentes tipos de partículas em um mesmo sistema de partículas
- O processo para se animar *Group Particle Systems* é:

# Grupos de Partículas

---

- Grupos de partículas são úteis quando se quer animar diferentes tipos de partículas em um mesmo sistema de partículas
- O processo para se animar *Group Particle Systems* é:
  1. Modelar um exemplar de cada tipo de partícula que se deseja animar

# Grupos de Partículas

---

- Grupos de partículas são úteis quando se quer animar diferentes tipos de partículas em um mesmo sistema de partículas
- O processo para se animar *Group Particle Systems* é:
  1. Modelar um exemplar de cada tipo de partícula que se deseja animar
  2. Criar um grupo de partículas que englobe todos os exemplares selecionados através do menu **Add | Group | Create New Group** (tecla de atalho: **Ctrl+G**)

# Grupos de Partículas

---

- Grupos de partículas são úteis quando se quer animar diferentes tipos de partículas em um mesmo sistema de partículas
- O processo para se animar *Group Particle Systems* é:
  1. Modelar um exemplar de cada tipo de partícula que se deseja animar
  2. Criar um grupo de partículas que englobe todos os exemplares selecionados através do menu **Add | Group | Create New Group** (tecla de atalho: **Ctrl+G**)
  3. Em seguida, nomear o novo grupo recém criado

# Grupos de Partículas

---

- Grupos de partículas são úteis quando se quer animar diferentes tipos de partículas em um mesmo sistema de partículas
- O processo para se animar *Group Particle Systems* é:
  1. Modelar um exemplar de cada tipo de partícula que se deseja animar
  2. Criar um grupo de partículas que englobe todos os exemplares selecionados através do menu **Add | Group | Create New Group** (tecla de atalho: **Ctrl+G**)
  3. Em seguida, nomear o novo grupo recém criado
  4. Criar um novo emissor de partículas, definindo todos os parâmetros desejados para a animação de partículas

# Grupos de Partículas

---

- Grupos de partículas são úteis quando se quer animar diferentes tipos de partículas em um mesmo sistema de partículas
- O processo para se animar *Group Particle Systems* é:
  1. Modelar um exemplar de cada tipo de partícula que se deseja animar
  2. Criar um grupo de partículas que englobe todos os exemplares selecionados através do menu **Add | Group | Create New Group** (tecla de atalho: **Ctrl+G**)
  3. Em seguida, nomear o novo grupo recém criado
  4. Criar um novo emissor de partículas, definindo todos os parâmetros desejados para a animação de partículas
  5. No painel *Render* selecione a opção **Group** indicando em *Dupli Group* o nome do grupo a ser utilizado

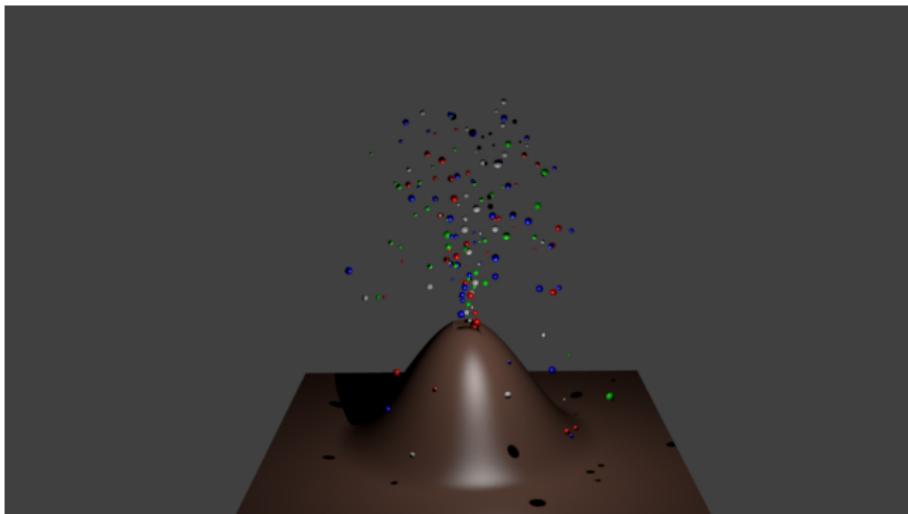
# Grupos de Partículas

---

- Grupos de partículas são úteis quando se quer animar diferentes tipos de partículas em um mesmo sistema de partículas
- O processo para se animar *Group Particle Systems* é:
  1. Modelar um exemplar de cada tipo de partícula que se deseja animar
  2. Criar um grupo de partículas que englobe todos os exemplares selecionados através do menu **Add | Group | Create New Group** (tecla de atalho: **Ctrl+G**)
  3. Em seguida, nomear o novo grupo recém criado
  4. Criar um novo emissor de partículas, definindo todos os parâmetros desejados para a animação de partículas
  5. No painel *Render* selecione a opção **Group** indicando em *Dupli Group* o nome do grupo a ser utilizado
    - Altenativamente, a opção **Use Count** permite a especificação de proporções distintas para cada tipo de partícula pertencente ao grupo

# Grupos de Partículas

---



# Controle de Manobras

---

- Controle de Manobras (ou *Steering Behaviors*) representa na verdade um conjunto de algoritmos que tem por objetivo animar grupos de objetos (chamados **boids**) a partir da combinação controlada de forças externas

# Controle de Manobras

---

- Controle de Manobras (ou *Steering Behaviors*) representa na verdade um conjunto de algoritmos que tem por objetivo animar grupos de objetos (chamados **boids**) a partir da combinação controlada de forças externas
- Originalmente proposto por Craig Reinolds no artigo disponível em <http://www.red3d.com/cwr/steer/>

# Controle de Manobras

---

- Controle de Manobras (ou *Steering Behaviors*) representa na verdade um conjunto de algoritmos que tem por objetivo animar grupos de objetos (chamados **boids**) a partir da combinação controlada de forças externas
- Originalmente proposto por Craig Reynolds no artigo disponível em <http://www.red3d.com/cwr/steer/>
- O próprio autor disponibiliza uma biblioteca (*OpenSteer*) em C++ com implementações para OpenGL dos *steering behaviors* disponível em <http://opensteer.sourceforge.net/>

## Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

## Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

Os controles para indivíduos são:

## Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

Os controles para indivíduos são:

Seek/Flee o boid segue/foge um determinado ponto fixo de referência (*target*)

# Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

Os controles para indivíduos são:

**Seek/Flee** o boid segue/foge um determinado ponto fixo de referência (*target*)

**Pursue/Evade** o boid persegue/se evade de um determinado ponto móvel de referência

# Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

Os controles para indivíduos são:

**Seek/Flee** o boid segue/foge um determinado ponto fixo de referência (*target*)

**Pursue/Evade** o boid persegue/se evade de um determinado ponto móvel de referência

**Wander** o boid anda sem trajetória definida (aleatório)

# Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

Os controles para indivíduos são:

**Seek/Flee** o boid segue/foge um determinado ponto fixo de referência (*target*)

**Pursue/Evade** o boid persegue/se evade de um determinado ponto móvel de referência

**Wander** o boid anda sem trajetória definida (aleatório)

**Arrival** o boid traça uma trajetória reta até parar em um ponto de referência

# Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

Os controles para indivíduos são:

**Seek/Flee** o boid segue/foge um determinado ponto fixo de referência (*target*)

**Pursue/Evade** o boid persegue/se evade de um determinado ponto móvel de referência

**Wander** o boid anda sem trajetória definida (aleatório)

**Arrival** o boid traça uma trajetória reta até parar em um ponto de referência

**Obstacle Avoidance** o boid evita colidir com um obstáculo fixo no ambiente (contorna)

# Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

Os controles para indivíduos são:

**Seek/Flee** o boid segue/foge um determinado ponto fixo de referência (*target*)

**Pursue/Evade** o boid persegue/se evade de um determinado ponto móvel de referência

**Wander** o boid anda sem trajetória definida (aleatório)

**Arrival** o boid traça uma trajetória reta até parar em um ponto de referência

**Obstacle Avoidance** o boid evita colidir com um obstáculo fixo no ambiente (contorna)

**Containment** o boid evita deixar um espaço confinado (área restrita)

# Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

Os controles para indivíduos são:

**Seek/Flee** o boid segue/foge um determinado ponto fixo de referência (*target*)

**Pursue/Evade** o boid persegue/se evade de um determinado ponto móvel de referência

**Wander** o boid anda sem trajetória definida (aleatório)

**Arrival** o boid traça uma trajetória reta até parar em um ponto de referência

**Obstacle Avoidance** o boid evita colidir com um obstáculo fixo no ambiente (contorna)

**Containment** o boid evita deixar um espaço confinado (área restrita)

**Wall Following** o boid tenta acompanhar uma trajetória delimitada a uma distância fixa

# Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

Os controles para indivíduos são:

**Seek/Flee** o boid segue/foge um determinado ponto fixo de referência (*target*)

**Pursue/Evade** o boid persegue/se evade de um determinado ponto móvel de referência

**Wander** o boid anda sem trajetória definida (aleatório)

**Arrival** o boid traça uma trajetória reta até parar em um ponto de referência

**Obstacle Avoidance** o boid evita colidir com um obstáculo fixo no ambiente (contorna)

**Containment** o boid evita deixar um espaço confinado (área restrita)

**Wall Following** o boid tenta acompanhar uma trajetória delimitada a uma distância fixa

**Path Following** o boid tenta seguir uma trajetória evitando transpor as margens do caminho

# Controle de Manobras

---

Os controles de manobras propostos por Reinolds são divididos em duas classes: controles de indivíduos e controles de bandos

Os controles para indivíduos são:

**Seek/Flee** o boid segue/foge um determinado ponto fixo de referência (*target*)

**Pursue/Evade** o boid persegue/se evade de um determinado ponto móvel de referência

**Wander** o boid anda sem trajetória definida (aleatório)

**Arrival** o boid traça uma trajetória reta até parar em um ponto de referência

**Obstacle Avoidance** o boid evita colidir com um obstáculo fixo no ambiente (contorna)

**Containment** o boid evita deixar um espaço confinado (área restrita)

**Wall Following** o boid tenta acompanhar uma trajetória delimitada a uma distância fixa

**Path Following** o boid tenta seguir uma trajetória evitando transpor as margens do caminho

**Flow Field Following** o boid se alinha a um campo de forças e se desloca na direção do mesmo

# Controle de Manobras

---

Os controles para grupos de boids são:

# Controle de Manobras

---

Os controles para grupos de boids são:

Crowd Path Following o boid analisa o comportamento médio de um grupo e o repete

# Controle de Manobras

---

Os controles para grupos de boids são:

**Crowd Path Following** o boid analisa o comportamento médio de um grupo e o repete

**Leader Following** o boid segue o comportamento de outro boid selecionado como líder do grupo

# Controle de Manobras

---

Os controles para grupos de boids são:

**Crowd Path Following** o boid analisa o comportamento médio de um grupo e o repete

**Leader Following** o boid segue o comportamento de outro boid selecionado como líder do grupo

**Unaligned Collision Avoidance** o boid tenta antever colisões e projeta mudanças de trajetórias para as evitar

# Controle de Manobras

---

Os controles para grupos de boids são:

**Crowd Path Following** o boid analisa o comportamento médio de um grupo e o repete

**Leader Following** o boid segue o comportamento de outro boid selecionado como líder do grupo

**Unaligned Collision Avoidance** o boid tenta antever colisões e projeta mudanças de trajetórias para as evitar

**Queuing** boids tendem a formar filas ao passar por espaços confinados

# Controle de Manobras

---

Os controles para grupos de boids são:

**Crowd Path Following** o boid analisa o comportamento médio de um grupo e o repete

**Leader Following** o boid segue o comportamento de outro boid selecionado como líder do grupo

**Unaligned Collision Avoidance** o boid tenta antever colisões e projeta mudanças de trajetórias para as evitar

**Queuing** boids tendem a formar filas ao passar por espaços confinados

**Flocking** boids tendem a se comportar como membros de um bando, ou seja, reagem ao comportamento dos demais boids no grupo

# Controle de Manobras

---

Os controles para grupos de boids são:

**Crowd Path Following** o boid analisa o comportamento médio de um grupo e o repete

**Leader Following** o boid segue o comportamento de outro boid selecionado como líder do grupo

**Unaligned Collision Avoidance** o boid tenta antever colisões e projeta mudanças de trajetórias para as evitar

**Queuing** boids tendem a formar filas ao passar por espaços confinados

**Flocking** boids tendem a se comportar como membros de um bando, ou seja, reagem ao comportamento dos demais boids no grupo

**Alignment** boids tendem a orientar sua posição em relação à posição dos boids vizinhos

# Controle de Manobras

---

Os controles para grupos de boids são:

**Crowd Path Following** o boid analisa o comportamento médio de um grupo e o repete

**Leader Following** o boid segue o comportamento de outro boid selecionado como líder do grupo

**Unaligned Collision Avoidance** o boid tenta antever colisões e projeta mudanças de trajetórias para as evitar

**Queuing** boids tendem a formar filas ao passar por espaços confinados

**Flocking** boids tendem a se comportar como membros de um bando, ou seja, reagem ao comportamento dos demais boids no grupo

**Alignment** boids tendem a orientar sua posição em relação à posição dos boids vizinhos

**Separation** boids tendem a manter uma distância mínima dos vizinhos

# Controle de Manobras

---

Os controles para grupos de boids são:

**Crowd Path Following** o boid analisa o comportamento médio de um grupo e o repete

**Leader Following** o boid segue o comportamento de outro boid selecionado como líder do grupo

**Unaligned Collision Avoidance** o boid tenta antever colisões e projeta mudanças de trajetórias para as evitar

**Queuing** boids tendem a formar filas ao passar por espaços confinados

**Flocking** boids tendem a se comportar como membros de um bando, ou seja, reagem ao comportamento dos demais boids no grupo

**Alignment** boids tendem a orientar sua posição em relação à posição dos boids vizinhos

**Separation** boids tendem a manter uma distância mínima dos vizinhos

**Cohesion** boids tendem a manter uma distância máxima em relação aos vizinhos

# Controle de Manobras em Blender

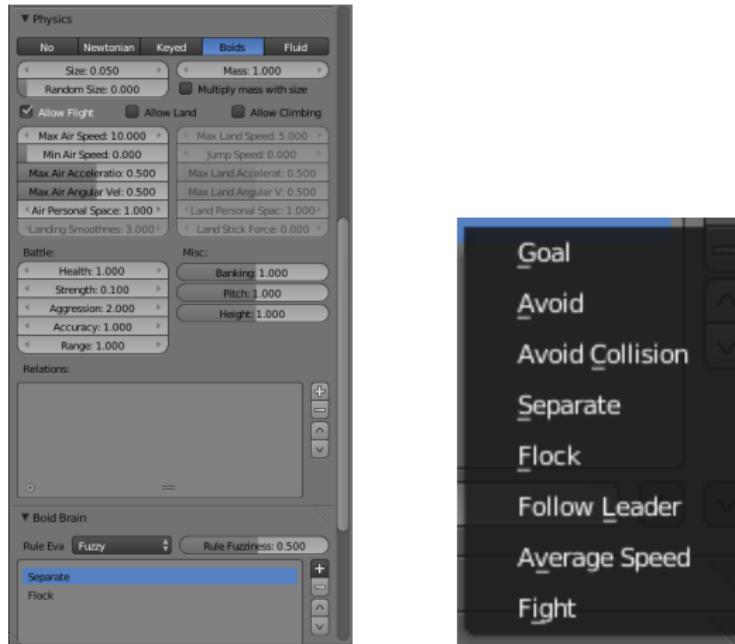
---

- Através de sistemas de partículas do tipo '*boids*'

# Controle de Manobras em Blender

---

- Através de sistemas de partículas do tipo '*boids*'



# Fluídos

---

- Dá-se o nome de **fluídos** para aqueles elementos que não apresentam forma geométrica definida, assumindo portanto a forma do volume que o contém

# Fluídos

---

- Dá-se o nome de **fluídos** para aqueles elementos que não apresentam forma geométrica definida, assumindo portanto a forma do volume que o contém
- Enquadram-se nessa categoria os líquidos e gases (como a fumaça e o fogo)

# Fluídos

---

- Dá-se o nome de **fluídos** para aqueles elementos que não apresentam forma geométrica definida, assumindo portanto a forma do volume que o contém
- Enquadram-se nessa categoria os líquidos e gases (como a fumaça e o fogo)
- Blender implementa um conjunto de algoritmos para simulação de partículas que se comportam como fluídos

# Fluídos

---

- Dá-se o nome de **fluídos** para aqueles elementos que não apresentam forma geométrica definida, assumindo portanto a forma do volume que o contém
- Enquadram-se nessa categoria os líquidos e gases (como a fumaça e o fogo)
- Blender implementa um conjunto de algoritmos para simulação de partículas que se comportam como fluídos
- Simulação da dinâmica de fluídos envolve um processamento denominado *cálculo volumétrico*, ou seja, as partículas são processadas em uma região limitada do espaço denominada **domínio**

# Fluídos

---

- Um domínio nada mais é do que um objeto tridimensional (malha poligonal). P.ex.: cubo

# Fluídos

---

- Um domínio nada mais é do que um objeto tridimensional (malha poligonal). P.ex.: cubo
- Dentro do domínio serão posicionados outros dois elementos necessários para a determinação do comportamento das partículas:

# Fluídos

---

- Um domínio nada mais é do que um objeto tridimensional (malha poligonal). P.ex.: cubo
- Dentro do domínio serão posicionados outros dois elementos necessários para a determinação do comportamento das partículas:
  - Inflow representa o ponto de entrada do fluido no domínio da simulação

# Fluídos

---

- Um domínio nada mais é do que um objeto tridimensional (malha poligonal). P.ex.: cubo
- Dentro do domínio serão posicionados outros dois elementos necessários para a determinação do comportamento das partículas:
  - Inflow** representa o ponto de entrada do fluido no domínio da simulação
  - Outflow** representa o ponto de saída do fluido no domínio da simulação

# Fluídos

---

- Um domínio nada mais é do que um objeto tridimensional (malha poligonal). P.ex.: cubo
- Dentro do domínio serão posicionados outros dois elementos necessários para a determinação do comportamento das partículas:
  - Inflow** representa o ponto de entrada do fluido no domínio da simulação
  - Outflow** representa o ponto de saída do fluido no domínio da simulação
  - Fluido** representa o fluido posicionado inicialmente no domínio da simulação

# Fluídos

---

- Um domínio nada mais é do que um objeto tridimensional (malha poligonal). P.ex.: cubo
- Dentro do domínio serão posicionados outros dois elementos necessários para a determinação do comportamento das partículas:
  - Inflow** representa o ponto de entrada do fluido no domínio da simulação
  - Outflow** representa o ponto de saída do fluido no domínio da simulação
  - Fluido** representa o fluido posicionado inicialmente no domínio da simulação
  - Obstáculo** obstáculos ao deslocamento das partículas. P.ex.: vasilhame que conterá o líquido

# Fluídos

---

Processo de construção da simulação de fluídos:

# Fluídos

---

Processo de construção da simulação de fluídos:

1. Modelagem do espaço no qual a simulação será realizada, e no painel *Physics* configurar os parâmetros para **Fluid | Domain**

# Fluídos

---

Processo de construção da simulação de fluídos:

1. Modelagem do espaço no qual a simulação será realizada, e no painel *Physics* configurar os parâmetros para **Fluid | Domain**
2. Modelagem do objeto emissor de fluido (configurado para o tipo **Inflow**)

# Fluídos

---

Processo de construção da simulação de fluídos:

1. Modelagem do espaço no qual a simulação será realizada, e no painel *Physics* configurar os parâmetros para **Fluid | Domain**
2. Modelagem do objeto emissor de fluido (configurado para o tipo **Inflow**)
3. Modelagem e configuração dos objetos obstáculos

# Fluídos

---

Processo de construção da simulação de fluídos:

1. Modelagem do espaço no qual a simulação será realizada, e no painel *Physics* configurar os parâmetros para **Fluid | Domain**
2. Modelagem do objeto emissor de fluido (configurado para o tipo **Inflow**)
3. Modelagem e configuração dos objetos obstáculos
4. A simulação de fluidos deve sempre ser previamente computada via comando **bake**

# Fluídos

---

Processo de construção da simulação de fluídos:

1. Modelagem do espaço no qual a simulação será realizada, e no painel *Physics* configurar os parâmetros para **Fluid | Domain**
  2. Modelagem do objeto emissor de fluido (configurado para o tipo **Inflow**)
  3. Modelagem e configuração dos objetos obstáculos
  4. A simulação de fluidos deve sempre ser previamente computada via comando **bake**
- 
- **Lembrete:** tanto o objeto *inflow* quanto os obstáculos da simulações podem ser previamente animados, tanto via quadros-chave quanto via simulações físicas. Tais movimentos são levados em considerados durante a simulação de fluidos.

# Fluídos

---

- O objeto domínio é quem deve conter os parâmetros de renderização do líquido = material

# Fluídos

---

- O objeto domínio é quem deve conter os parâmetros de renderização do líquido = material
- Blender tem suporte tanto a *Blender Render* quanto a *Blender Cycles* para líquidos

# Fluídos

---

- O objeto domínio é quem deve conter os parâmetros de renderização do líquido = material
- Blender tem suporte tanto a *Blender Render* quanto a *Blender Cycles* para líquidos
- **Lembrete:** o uso do plugin **Network Render** dá suporte tanto ao processo de bake quanto de render de simulações, podendo representar uma redução significativa no tempo de produção

# Fluídos

---



# Fogo + Fumaça

---

- Fogo e fumaça são casos particulares de fluídos

## Fogo + Fumaça

---

- Fogo e fumaça são casos particulares de fluídos
- Blender inclui *templates* para gerar fluídos e fumaça

## Fogo + Fumaça

---

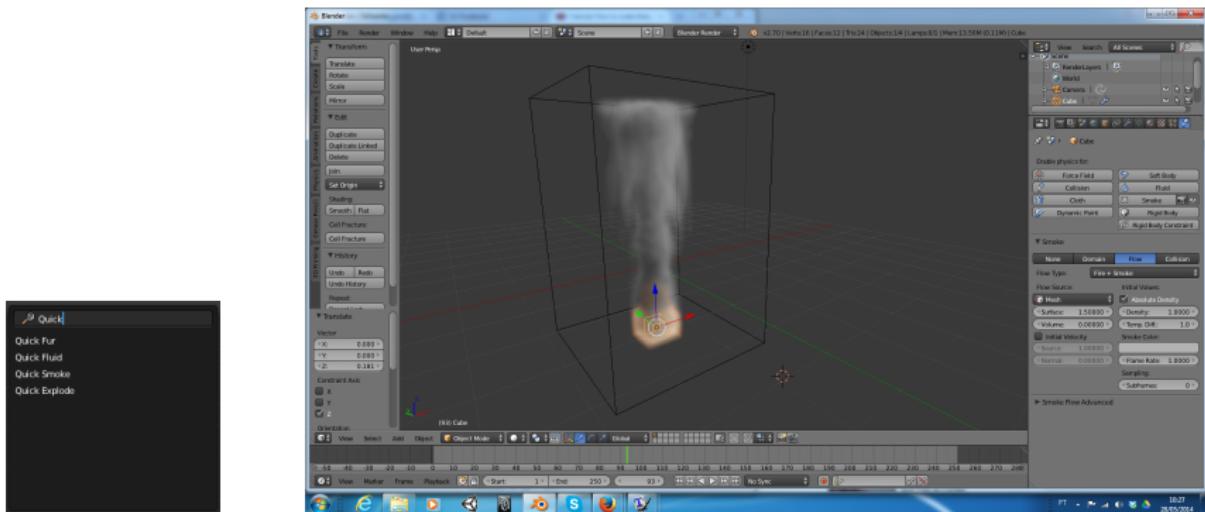
- Fogo e fumaça são casos particulares de fluídos
- Blender inclui *templates* para gerar fluídos e fumaça
- O template é acessível, no 3D View pela barra de espaço, buscando por “**Quick Smoke**”

## Fogo + Fumaça

---

- Fogo e fumaça são casos particulares de fluídos
- Blender inclui *templates* para gerar fluídos e fumaça
- O template é acessível, no 3D View pela barra de espaço, buscando por “**Quick Smoke**”
- Já é construído um domínio e um emissor de fumaça, o qual pode ser alterado o tipo para *Fire+Smoke* na aba Physics

# Fogo + Fumaça



## Fogo + Fumaça

---

A fim de se obter resultados mais realistas, deve-se modificar diversos parâmetros:

# Fogo + Fumaça

---

A fim de se obter resultados mais realistas, deve-se modificar diversos parâmetros:

- Para o domínio:

**Resolution** aumentar o número de divisões (p.ex. 100)

# Fogo + Fumaça

---

A fim de se obter resultados mais realistas, deve-se modificar diversos parâmetros:

- Para o domínio:

- Resolution aumentar o número de divisões (p.ex. 100)

- Smoke Adaptive Domain habilitar o uso de domínio adaptativo

# Fogo + Fumaça

---

A fim de se obter resultados mais realistas, deve-se modificar diversos parâmetros:

- Para o domínio:

- Resolution aumentar o número de divisões (p.ex. 100)

- Smoke Adaptive Domain habilitar o uso de domínio adaptativo

- Smoke High Resolution habilitar o uso de alta resolução e alterar o número de divisões (p.ex. 2)

# Fogo + Fumaça

---

A fim de se obter resultados mais realistas, deve-se modificar diversos parâmetros:

- Para o domínio:

- Resolution** aumentar o número de divisões (p.ex. 100)

- Smoke Adaptive Domain** habilitar o uso de domínio adaptativo

- Smoke High Resolution** habilitar o uso de alta resolução e alterar o número de divisões (p.ex. 2)

- Para o emissor:

- Flame Rate** alterar para um valor mais alto a fim de aumentar as chamas (p.ex. 5)

# Fogo + Fumaça

---

A fim de se obter resultados mais realistas, deve-se modificar diversos parâmetros:

- Para o domínio:

- Resolution** aumentar o número de divisões (p.ex. 100)

- Smoke Adaptive Domain** habilitar o uso de domínio adaptativo

- Smoke High Resolution** habilitar o uso de alta resolução e alterar o número de divisões (p.ex. 2)

- Para o emissor:

- Flame Rate** alterar para um valor mais alto a fim de aumentar as chamas (p.ex. 5)

- Dica:** para dar um efeito mais dramático, pode-se acrescentar um objeto **Force Field** do tipo **Wind**

## Fogo + Fumaça

---

- Ao se trabalhar com fluídos, nós abandonamos a ideia de que objetos são representados por malhas poligonais. Ao invés disso, temos uma **representação volumétrica** de objetos a partir de uma matriz tridimensional de voxels

## Fogo + Fumaça

---

- Ao se trabalhar com fluídos, nós abandonamos a ideia de que objetos são representados por malhas poligonais. Ao invés disso, temos uma **representação volumétrica** de objetos a partir de uma matriz tridimensional de voxels
- Por este motivo, objetos volumétricos não são renderizados da mesma forma que polígonos e irão requerer materiais diferentes para isso

## Fogo + Fumaça

---

- Ao se trabalhar com fluídos, nós abandonamos a ideia de que objetos são representados por malhas poligonais. Ao invés disso, temos uma **representação volumétrica** de objetos a partir de uma matriz tridimensional de voxels
- Por este motivo, objetos volumétricos não são renderizados da mesma forma que polígonos e irão requerer materiais diferentes para isso
- Na aba *Materials*, o template terá já criado também um modelo padrão para materiais do tipo “**Volume**”. Cabe ao animador apenas configurá-lo apropriadamente às suas necessidades

# Fogo + Fumaça

---

Parâmetros a configurar para o material volumétrico:

## Fogo + Fumaça

---

Parâmetros a configurar para o material volumétrico:

**Density Scale** como a maior parte da informação de densidade provém da simulação de partículas (e temos 2 tipos de partículas: fogo e fumaça), aconselha-se usar valores baixos (p.ex. 2)

## Fogo + Fumaça

---

Parâmetros a configurar para o material volumétrico:

**Density Scale** como a maior parte da informação de densidade provém da simulação de partículas (e temos 2 tipos de partículas: fogo e fumaça), aconselha-se usar valores baixos (p.ex. 2)

**Integration** determina como diferentes voxels serão compostos para formar um pixel. Valores menores para o passo (**Step Size**) indicam maior precisão (porém mais processamento). Valor sugerido: 0.05

## Partículas Estáticas

---

- Dá-se o nome de partículas estáticas a um tipo especial de partícula que está associada a algum outro objeto, sendo portanto fixado a este

## Partículas Estáticas

---

- Dá-se o nome de partículas estáticas a um tipo especial de partícula que está associada a algum outro objeto, sendo portanto fixado a este
- Em muitos casos, as partículas estáticas serem utilizadas apenas como técnica de modelagem: pedras no solo marciano, grãos de uma espiga de milho, prédios de uma cidade, etc.

# Partículas Estáticas

---

- Dá-se o nome de partículas estáticas a um tipo especial de partícula que está associada a algum outro objeto, sendo portanto fixado a este
- Em muitos casos, as partículas estáticas serem utilizadas apenas como técnica de modelagem: pedras no solo marciano, grãos de uma espiga de milho, prédios de uma cidade, etc.
- A sua abordagem mais tradicional (e de interesse para as animações computacionais), está na sua utilização para simulação física do tipo **“cabelo”**: cabelo, pêlo, grama, etc.

# Partículas Estáticas

---

- Em Blender, a simulação de cabelo (*Hair*) se dá no painel **Particles**

# Partículas Estáticas

---

- Em Blender, a simulação de cabelo (*Hair*) se dá no painel **Particles**
- Principais parâmetros:

# Partículas Estáticas

---

- Em Blender, a simulação de cabelo (*Hair*) se dá no painel **Particles**
- Principais parâmetros:

**Emission** número de partículas (*Number*) e comprimento (*Length*)

# Partículas Estáticas

---

- Em Blender, a simulação de cabelo (*Hair*) se dá no painel **Particles**
- Principais parâmetros:

**Emission** número de partículas (*Number*) e comprimento (*Length*)

**Render** material e *Strand render* (no caso de fios como cabelo e grama)

# Partículas Estáticas

---

- Em Blender, a simulação de cabelo (*Hair*) se dá no painel **Particles**
- Principais parâmetros:

**Emission** número de partículas (*Number*) e comprimento (*Length*)

**Render** material e *Strand render* (no caso de fios como cabelo e grama)

**Children** tipo de sub-divisão recursiva da simulação

# Partículas Estáticas



## Partículas Estáticas - Exercício

---

- Exercício: Construir uma cena de um gramado foto-realista balançando ao vento



# Exercício Prático Individual - EPI #5

---

**Nome:** *Animação de Grupos*

**Objetivo:** construção de uma animação que aplique as diversas técnicas de animação de grupos estudadas

- Enunciado:**
1. Imaginar uma estória que utilize os diversos tipos de partículas numa mesma cena
  2. produzir uma sequência de quadros através de *straight-ahead* e armazenar os quadros-chaves resultantes em uma timeline, à qual possa ser posteriormente editada
  3. compilar um filme resultante

- Requisitos:**
- A cena final deve conter os seguintes tipos de animações: partículas (halo, linha, etc.), grupos de objetos geométricos como partículas (cubos, esferas, suzanne, etc.), partículas estáticas (cabelo, grama, etc.) e boids
  - Sugestão: animar elementos como luzes e câmeras
  - Pode conter trilha sonora e/ou efeitos sonoros
  - Duração mínima: 30 segundos
  - Deve apresentar tela de créditos iniciais e finais (o tempo de apresentação dos créditos não contabiliza para o tempo da animação)

**Prazo para Entrega:** 2 semanas (filme + estória + slides ilustrando o processo)

**Peso:** 15% da média semestral

# Projeto Final

---

**Objetivo:** avaliar a capacidade do aluno de combinar diferentes técnicas a fim de produzir um filme de animação completo

- Enunciado:**
1. Selecionar uma estória ou cena de filme para ser utilizado como roteiro inicial
  2. Extrair o áudio da cena para ser utilizado como referência
  3. Alterar o roteiro selecionado para acrescentar tantos elementos quanto necessários, a fim de cobrir todas as técnicas estudadas durante o semestre
  4. compilar um filme resultante

- Requisitos:**
- A cena final deve conter uma compilação de todas as técnicas vistas nos EPI #1 a EPI #5
  - Duração mínima: 70 segundos
  - Deve apresentar tela de créditos iniciais e finais (o tempo de apresentação dos créditos não contabiliza para o tempo da animação)

**Sugestão:** criar uma vinheta de abertura para o vídeo final

- Os slides da apresentação devem abordar: a estória original escolhida, as alterações propostas a fim de cobrir todos os EPIs, um *checklist* indicando onde cada técnica foi utilizada na cena

**Prazo para Entrega:** ? (filme + estória + slides ilustrando o processo)

**Peso:** 25% da média semestral

[Parent, 2002, Reinolds, 1999, Brito, 2011, Mullen, 2007, Osipa, 2003, Roberts, 2007, Kundert-Gibbs and Kundert-Gibbs, 2009]