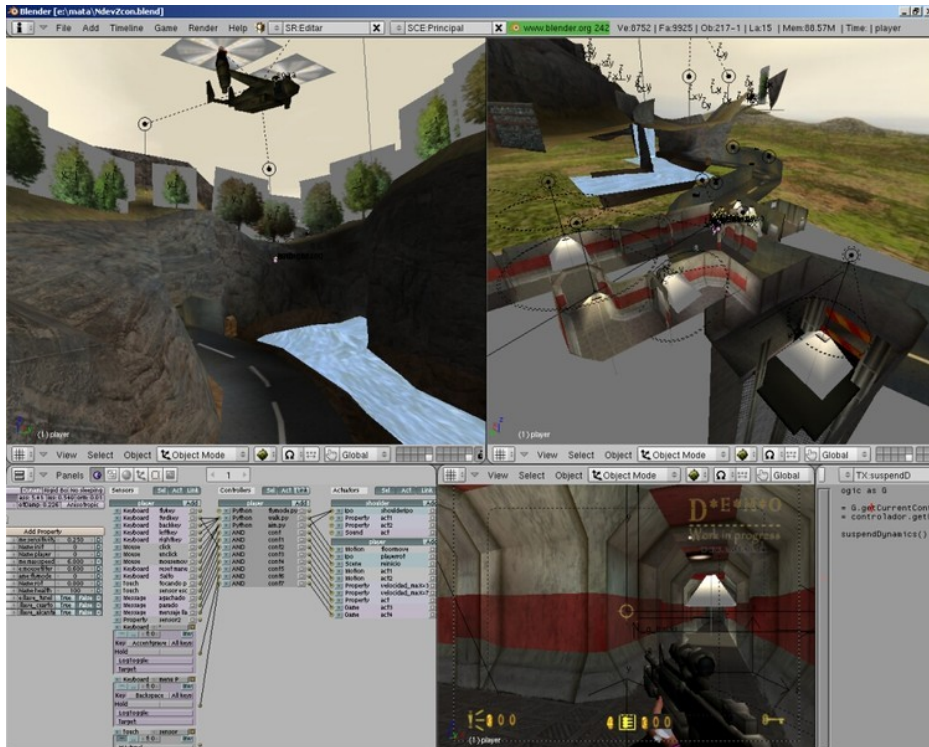
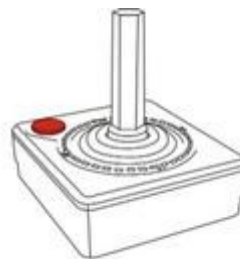


Desenvolvimento de Jogos 3D com o Blender



Por
Marcos Romero
Outubro / 2008

www.blender.org

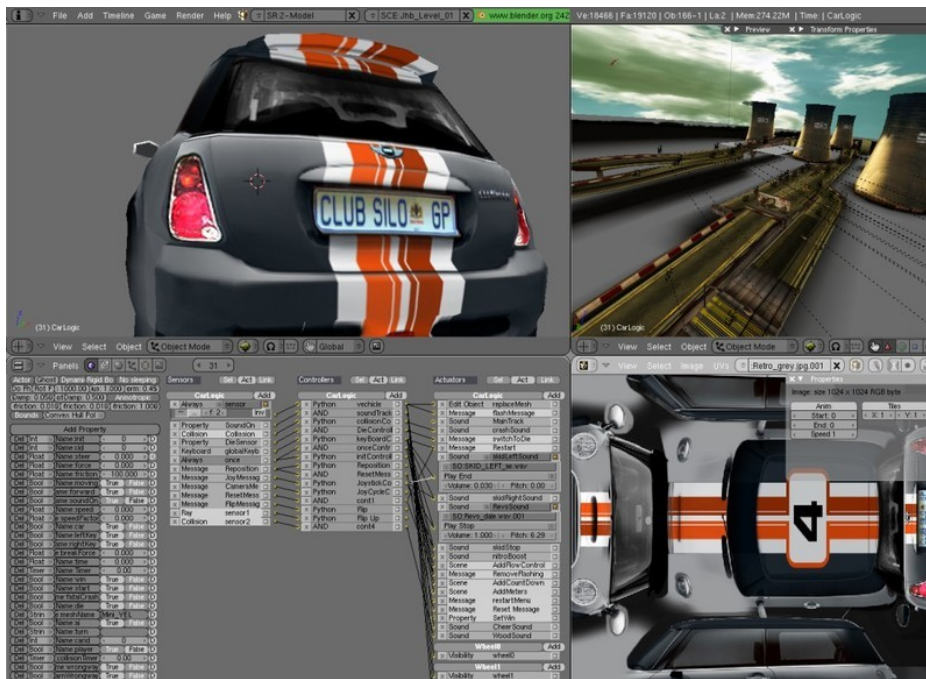


Big Buck Bunny



- O Blender já conquistou seu espaço na área de animações em 3D.
- Projetos como “Elephant Dreams” e o “Big Buck Bunny” mostraram o que o Blender é capaz de fazer.
- O Blender também tem sido usado na produção de jogos.

Formas de Uso do Blender



Club Silo

- Blender já tem sido muito usado na área de jogos para a modelagem 3D de personagens e cenários.
- O Blender também possui uma game engine que permite a criação de um jogo completo dentro da própria ferramenta.

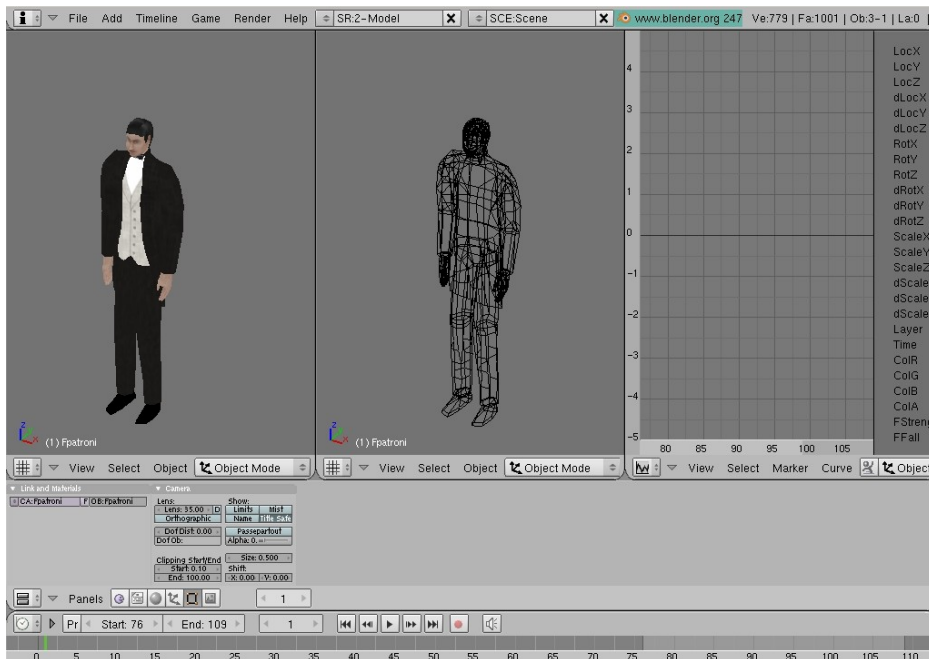
GP Brasil 1.0



- GP Brasil 1.0 foi o primeiro fliperama 100% brasileiro desenvolvido com ferramentas livres.
- Tanto o software quanto as cabines foram feitos pela Escola Politécnica da USP e pela empresa Matic Entretenimento.
- O Blender foi usado para a modelagem 3D do jogo.

Revolta da Cabanagem

- Outro projeto nacional feito somente com ferramentas livres é o jogo “Revolta da Cabanagem”.
- Desenvolvido pelo Laboratório de Realidade Virtual da Universidade Federal do Pará.
- O Blender foi usado na modelagem 3D dos personagens e objetos do jogo.



Blender Game Expose



Me and My Airship

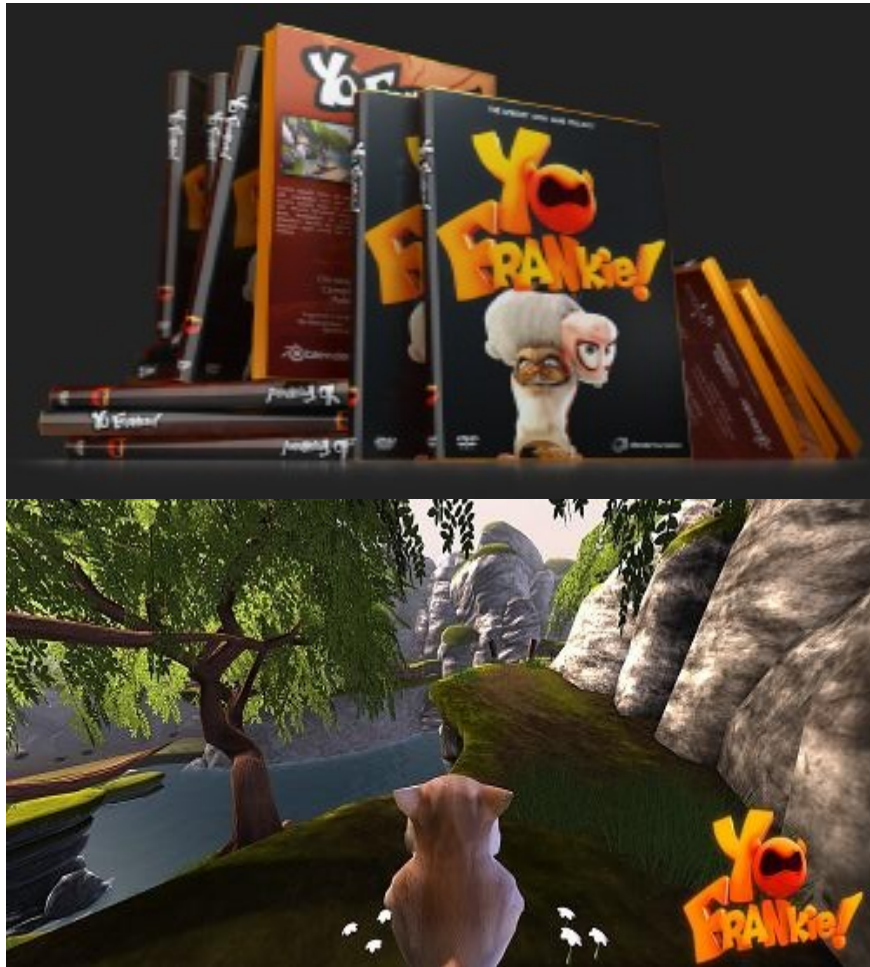
- Existe uma exposição de jogos anual chamado “Blender Game Expose”.
- São apresentados diversos projetos que foram feitos com a Game Engine do Blender.
- Exemplo de alguns jogos:
“Eternal Odyssey”;
“Sinister Plot”;
“Me and My Airship”;

Monkey Game Project



- Este foi um dos projetos exibidos no Blender Games Expose.
- Ele surgiu com propósito de chamar pessoas iniciantes a participarem e aprenderem em um projeto Blender.
- O estilo do jogo é de aventura.

Yo Frankie!



- Yo Frankie é um jogo feito no Blender baseado na estória do Big Buck Bunny.
- Existem duas versões do jogo. Uma feita totalmente na Blender Game Engine e a outra que usa a engine Crystal Space para a renderização.
- O DVD possui as duas versões, além de diversos tutoriais sobre a produção do jogo.

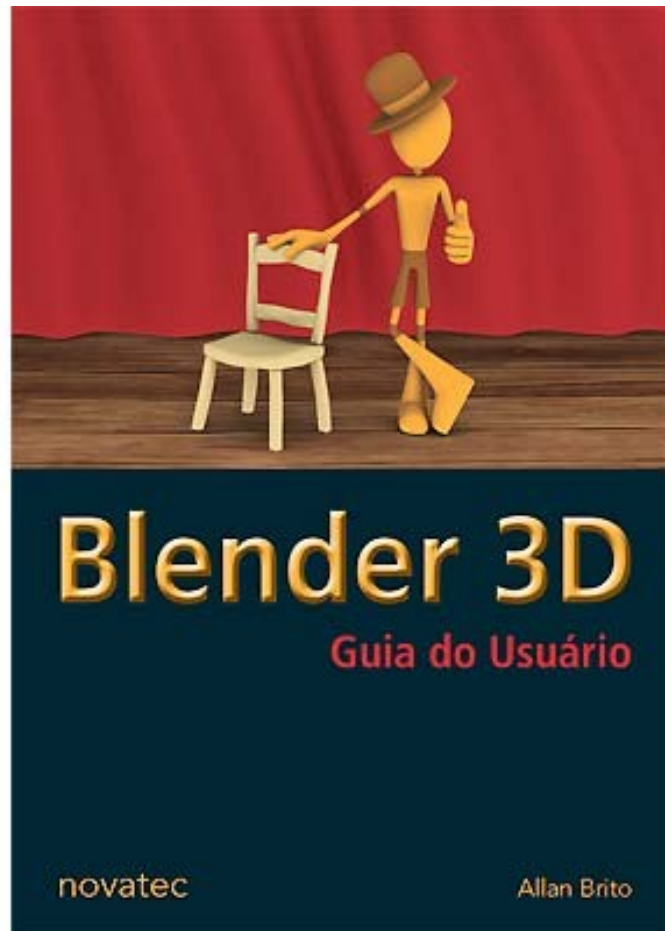
Blender Gamekit



- Existe um livro em inglês que foi lançado em 2003 que aborda a criação de jogos no Blender.
- O livro traz 10 jogos de demonstração e ele está disponível para download.
- Uma segunda edição do livro está sendo produzida para a versão atual do Blender.

<http://download.blender.org/documentation/gamekit1/>

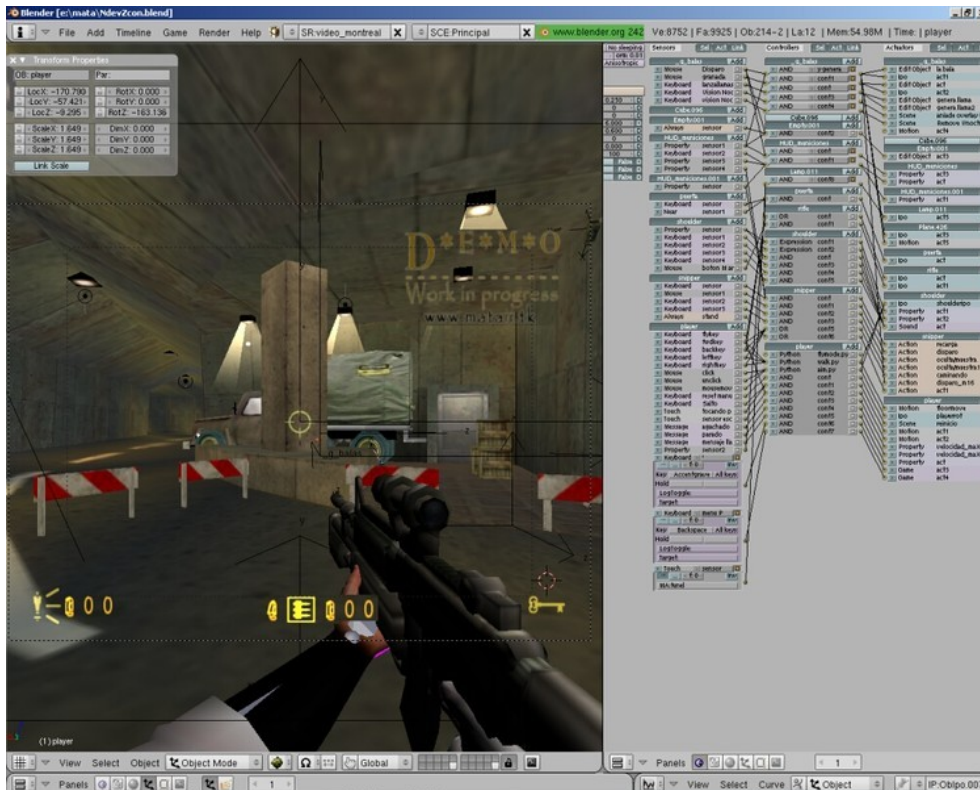
Livro em Português



- O livro “Blender 3D: Guia do Usuário” de Allan Brito é uma ótima referência para o uso geral da ferramenta.
- O Autor também mantém um blog com bastante informações sobre o Blender e sua Game Engine.

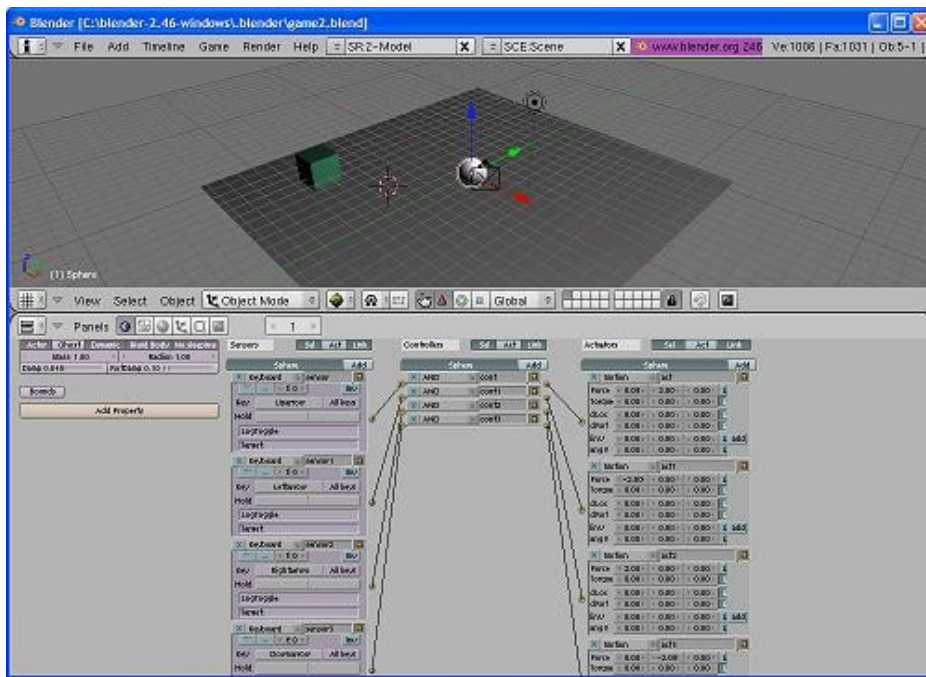
www.allanbrito.com

Noções da Game Engine



- A Game Engine do Blender já existe a algum tempo mas agora tem recebido atenção especial por causa do projeto “Yo Frankie!”
- Ela permite a criação de animações interativas ou de protótipos de jogos de uma forma bem simples.

Painel de Lógica



- Neste painel que será especificado o que acontece dentro do jogo.
- Ele reúne os Sensores, Controladores, Atuadores e Propriedades de determinado objeto do jogo.

Sensores



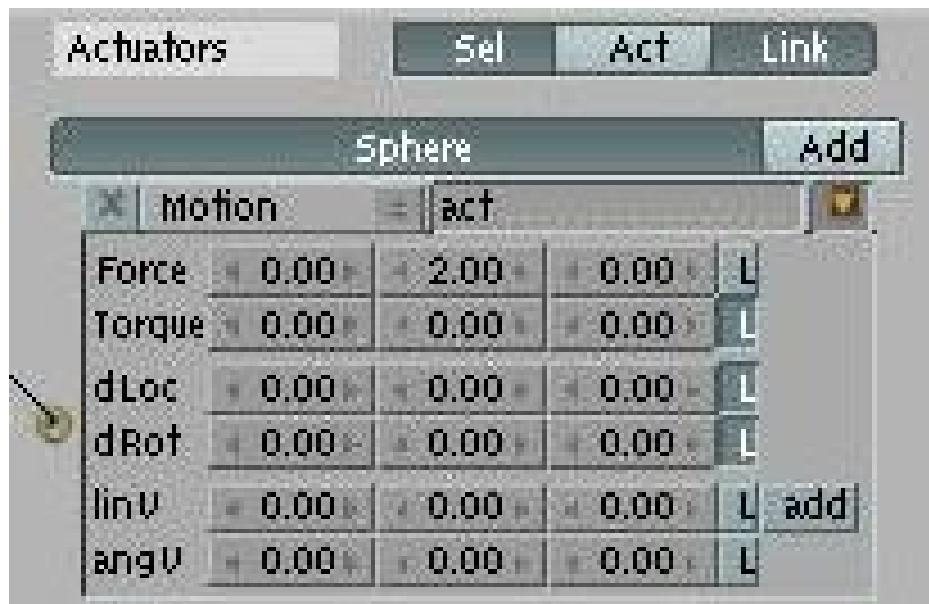
- Um sensor irá detectar alguma forma de entrada ou evento.
- Ex: Teclado, Mouse, Joystick, Colisão, Proximidade, Propriedade, Always.

Controladores



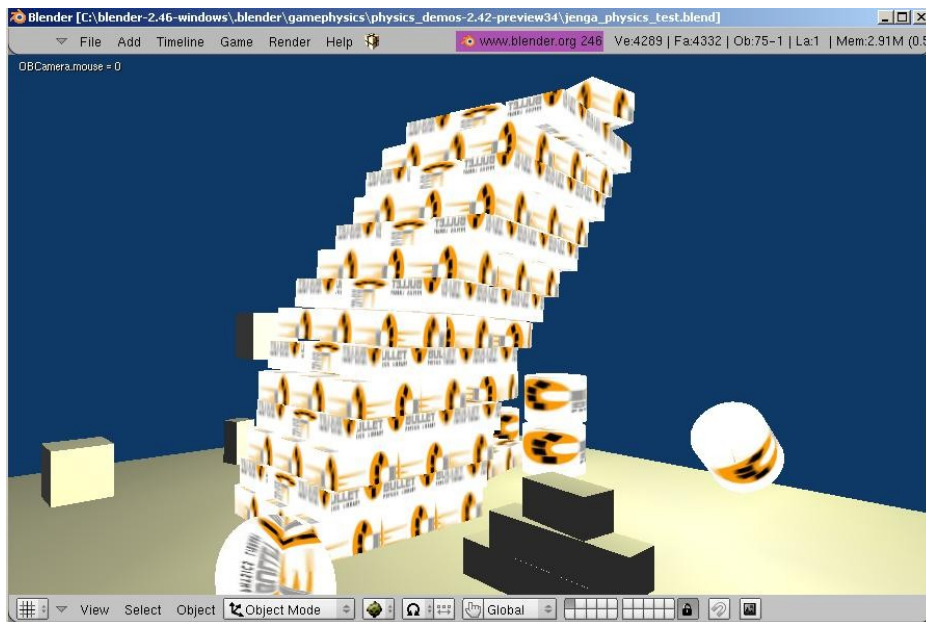
- São usados para ligar os Sensores aos Atuadores, permitindo um controle complexo sobre essa interação.
- Ex: AND, OR, Expressão, Script Python.

Atuadores



- Um atuador irá executar uma ação dentro do jogo, em decorrência dos eventos e controladores.
- Alguns Exemplos:
 - Motion, para movimentar um objeto na cena.
 - Game, para reiniciar ou sair do nível atual. Também para carregar uma cena.
 - Sound, para música e efeitos sonoros.

Engine de Física



- O Blender utiliza uma engine de física chamada Bullet, que manipula as forças atuantes nos objetos e suas interações.
- Essa engine é simples de ser utilizada e permite a criação de ambientes bem interativos.

Atores

- Atores são os objetos nos quais a engine de física atua. Existem alguns atributos já criados:
 - Ghost: evita a colisão com outros objetos.
 - Dynamic: indica um objeto físico e abre outras opções.
 - Rigid Body: a rotação é feita de forma automática.
 - No sleeping: indica se o objeto pode ser desativado.

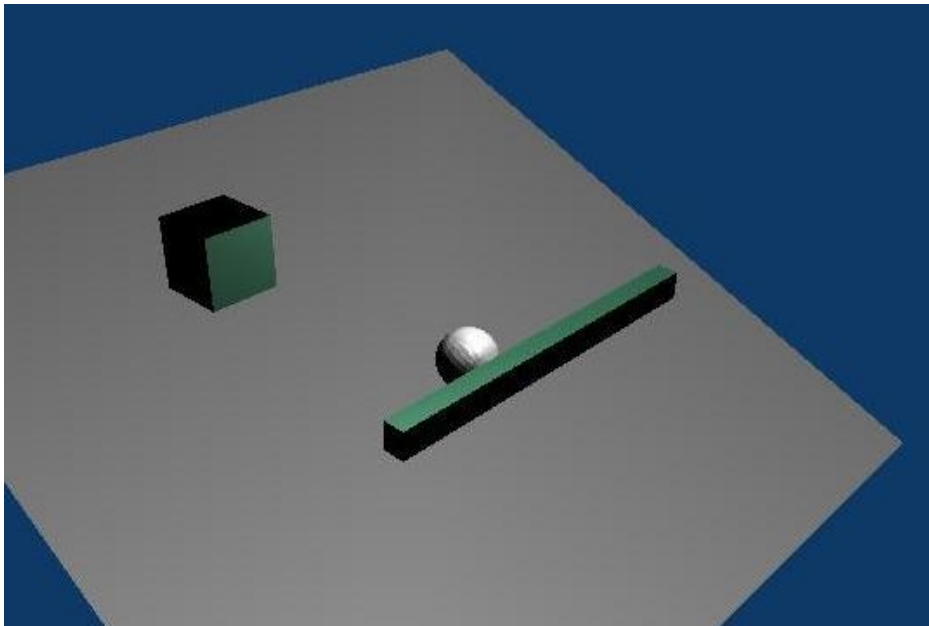


Propriedades

Add Property					
Del	Float	⇌	Name:velocidade	◀ 4.000 ▶	D
Del	Int	⇌	Name:vidas	◀ 3 ▶	D
Del	Bool	⇌	Name:poder	True False	D
Del	String	⇌	Name:nome	Jogador	D
Del	Timer	⇌	Name:espera	◀ 15.000 ▶	D

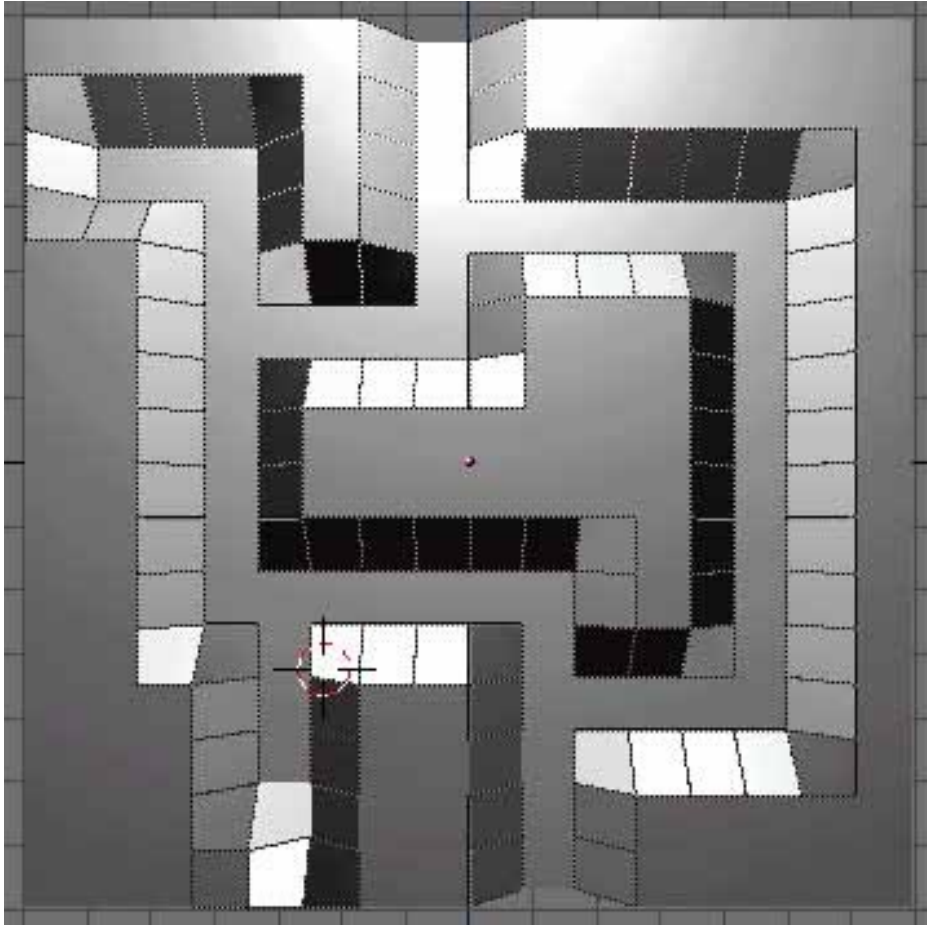
- O Blender permite a definições de propriedades para os objetos.
- Elas podem ser dos tipos: Float, Int, Bool, String e Timer.
- Essas propriedades é que irão guardar o estado atual do jogo.

Colisões



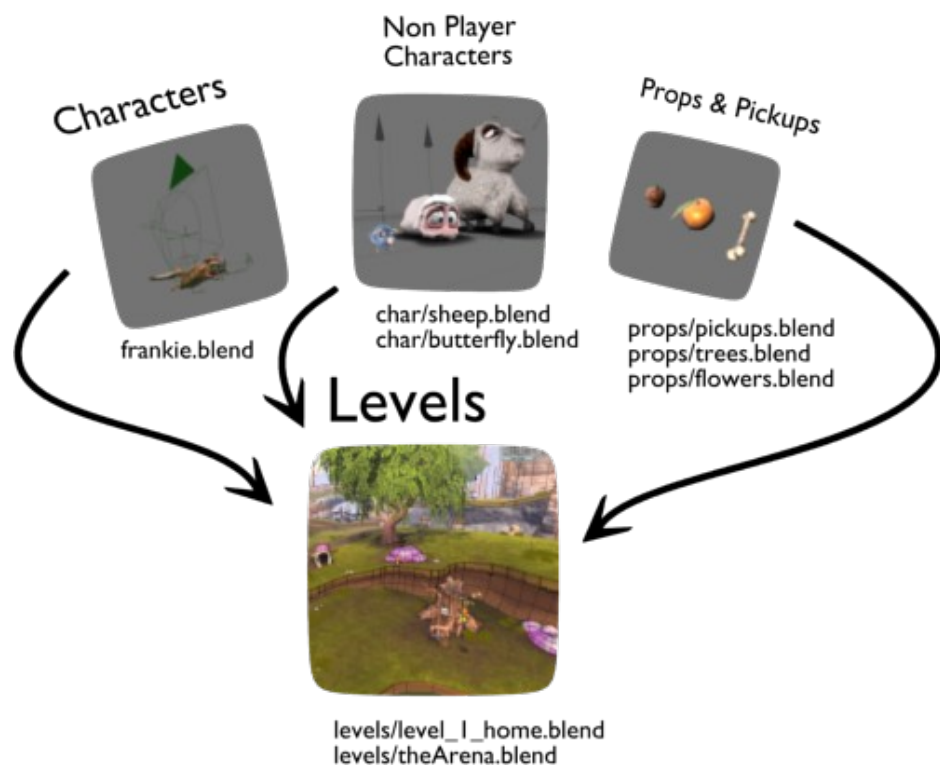
- As colisões podem ser detectadas a partir de uma forma geométrica simples ao redor do objeto, como caixas ou esferas.
- Ou podem usar a forma precisa do objeto, mas que requer um maior tempo de processamento.
- Essas colisões e suas reações são detectadas pela engine de física automaticamente.

Labirinto



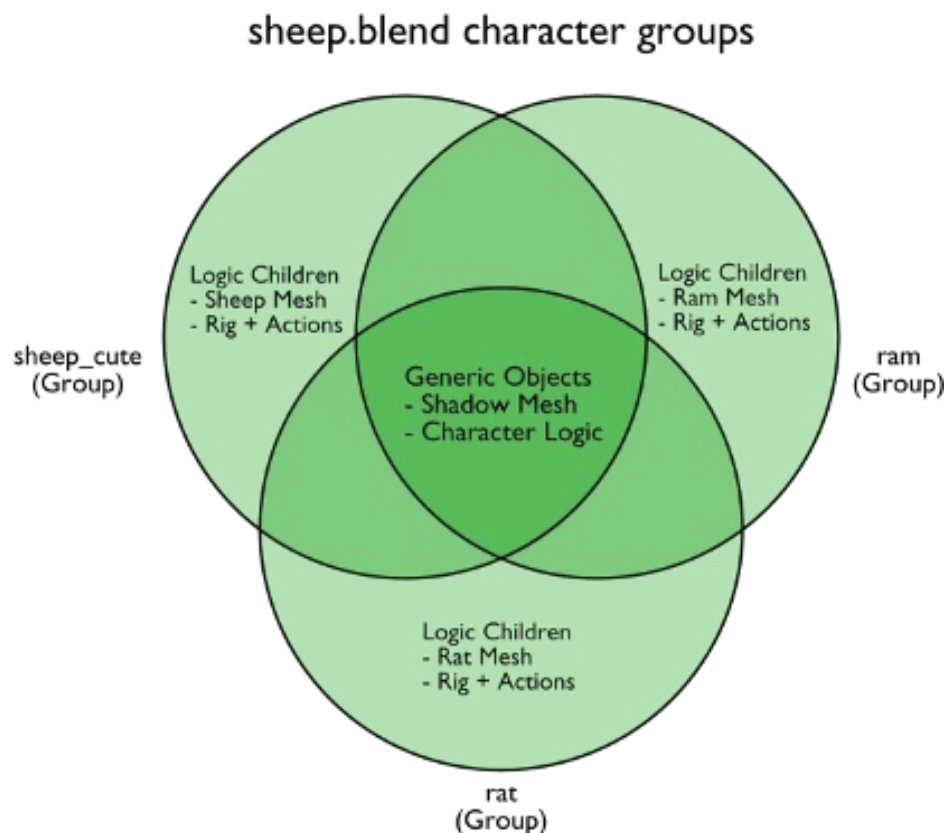
- Um labirinto é um cenário bastante comum em jogos.
- No Blender ele pode ser criado a partir da subdivisão de um plano em diversas partes menores.
- Após isso, basta alterar a altura dessas partes menores.

A Lógica de Yo Frankie!



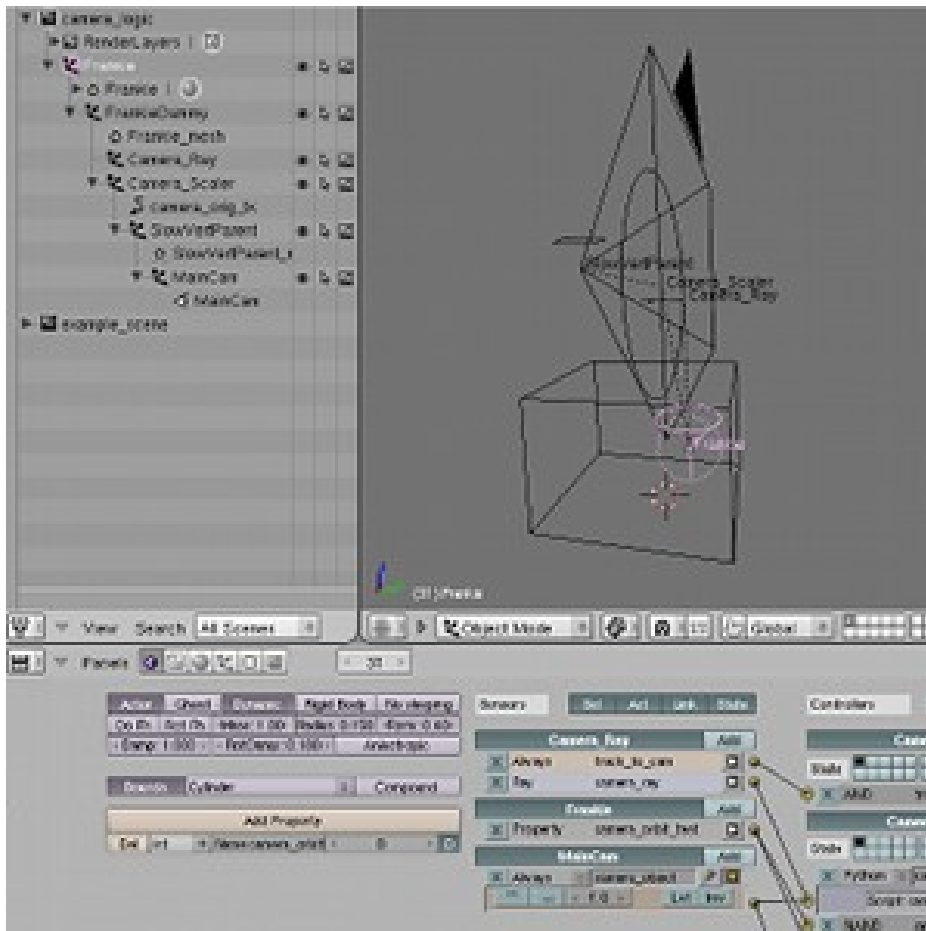
- Foi publicado um artigo comentando sobre a construção do jogo Yo Frankie! No Blender.
- O jogo é separado em diversos arquivos `.blend` para os personagens, níveis e outros objetos 3D.
- Os arquivos de níveis contém essencialmente cenários estáticos e os links para os outros objetos.

Lógica Compartilhada



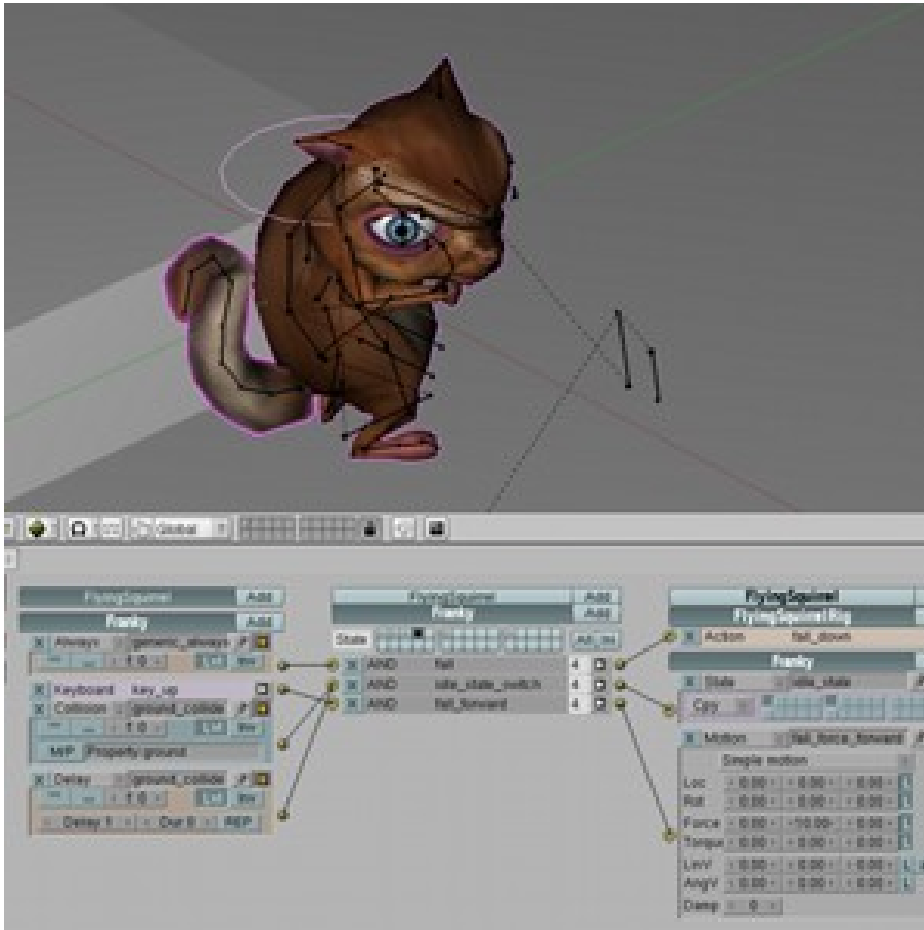
- O Blender permite o uso de grupos que compartilham determinados objetos.
- Com isso é possível definir a lógica dos personagens do jogo em um objeto genérico que será usado pelos outros objetos que contêm a parte visual.

Câmera



- A câmera do jogo é em terceira pessoa, ou seja, o personagem do jogo Frankie é exibido na tela.
- Há uma lógica que detecta a existência de algum obstáculo entre a câmera e o Frankie, fazendo o ajuste na posição dela se necessário.

Controle do Frankie



- Os controles padrões são as setas para o movimento e a barra de espaço para o pulo.
- Os estados do Frankie determinam os tipos de movimento que ele pode fazer.
- As propriedades do Nível determinam a interação com o Frankie, indicando por onde ele pode andar.

Portais



- Os portais permitem a ligação de um cena do Blender à outra cena ou arquivo do Blender, mantendo o estado de determinados objetos.
- Eles são usados na transição de níveis ou de cenários dentro do jogo.

Tela Dividida



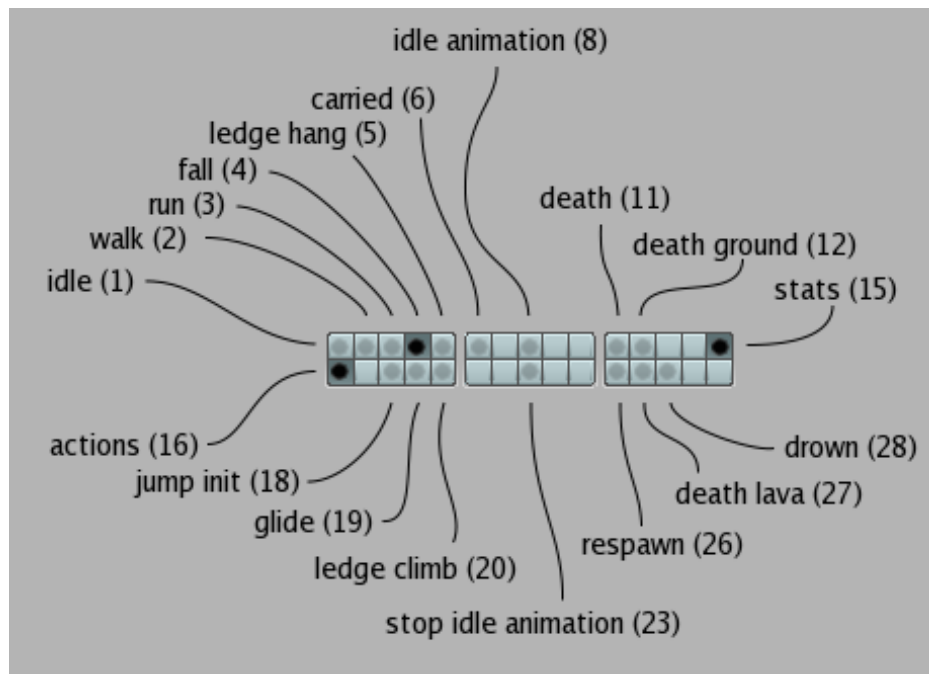
- O jogo permite dois jogadores simultâneos através da divisão de tela.
- Cada jogador recebe um ID que irá definir durante o processamento qual espaço de tela será usado e quais teclas executarão os comandos.

Menu



- O menu demonstra o uso de alguns componentes de interface de usuário no Blender, tais como itens de menu, radio buttons e checkboxes.
- Os Portais são usados para carregar as cenas (fases).
- As configurações são salvas em um arquivo externo.

Estados do Frankie



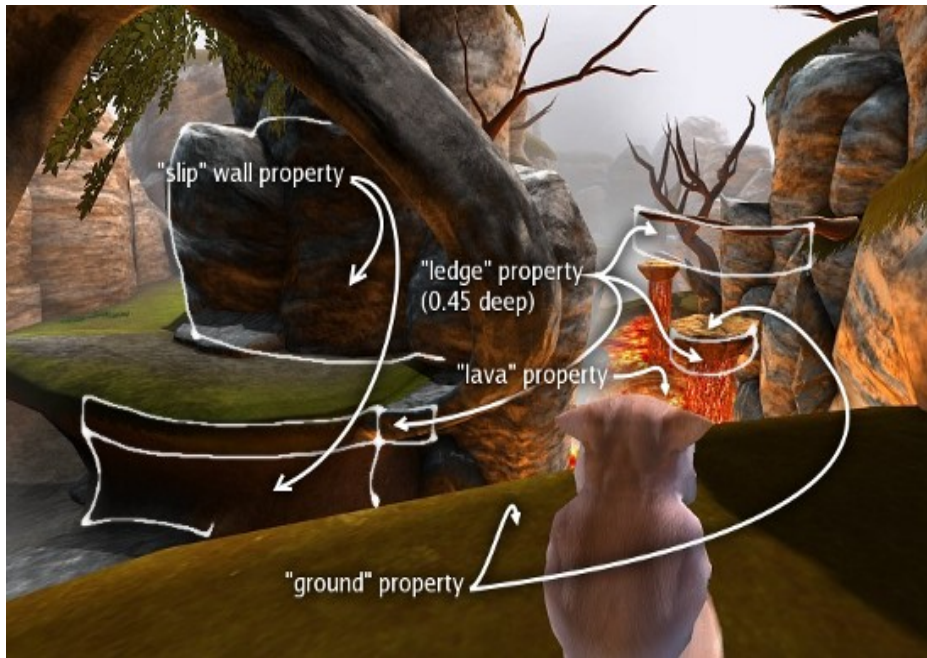
- Os estados ajudam a organizar o controle de Frankie.
- Alguns comandos podem ser processados somente em alguns estados.
- A partir do estado “Walk” pode-se ir para vários outros estados tipo: Corrida, Pulo, Morte.
- No estado “Morte” os comandos de movimento não são processados.

Propriedades

Int	Name:id	-1	D	
Int	Name:grounded	0	D	
Bool	Name:predator	True	False	D
Bool	Name:kickable	True	False	D
Int	Name:hit	0	D	
String	Name:orig_pos		D	
String	Name:ground_pos		D	
String	Name:ground_pos_old		D	
Timer	Name:water_touch_time	1.000	D	
Int	Name:camera_orbit	0	D	
Timer	Name:idle_anim_timer	-8.000	D	
Int	Name:carrying	0	D	
Int	Name:carried	0	D	
Timer	Name:force_walk	0.000	D	

- Existe um grande número de propriedades necessárias para guardar informações do jogo e definir as interações do objeto.
- Alguns exemplos: orig_pos, hit, carrying, grounded, life, life_max, can_climb.

Edição dos Níveis



- Para criar os níveis do jogo é preciso definir as propriedades que indicam a interação com o jogador.
- Essas propriedades são armazenadas em objetos invisíveis.
- Ex: ground, slip, ledge, bounce, kill, water, lava.