

## **SBIE 2000**

### **MÚLTIPLAS PERSPECTIVAS DE OBJETOS NO CONTEXTO EDUCACIONAL**

**André Santanchè**

santanche@unifacs.br

<http://www.geocities.com/santanche>

**Cesar Augusto Camillo Teixeira**

cesar@unifacs.br

## **Interfaces, Linguagens e Computadores**

As primeiras linguagens de programação e interfaces com sistemas operacionais tomaram como base para seu modelo de concepção a própria arquitetura do computador, ou seja, o centro era o computador.

## Interface e Novas Metáforas

“Metáfora é para muita gente um dispositivo da imaginação poética e da alegoria retórica – um assunto de linguagem extraordinária ao invés de ordinária. Além disto, metáfora é tipicamente vista como característica só da linguagem, um corpo de palavras ao invés de pensamento e ação. Por esta razão, muitas pessoas pensam que elas podem avançar perfeitamente bem sem metáforas. Nós percebemos, ao contrário, que essa metáfora impregna cada dia da vida, não somente na linguagem mas no pensamento e ação. Nosso sistema conceitual normal, em termos do que nós pensamos e agimos, é fundamentalmente metafórico por natureza”. [LAK80]

Mapeamento Metafórico

## Um pouco de história

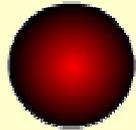
- SIMULA 67
  - Primeira Linguagem Orientada a Objetos
- Smalltalk
  - Projeto Dynabook
    - “Este ‘Dynabook’ foi baseado na visão de computadores pessoais baratos do tamanho de um caderno, tanto para adultos quanto crianças, com a capacidade de lidar com todas as suas respectivas necessidades de informação”. [KRE98]

## Princípios do Paradigma



### Objeto

Os objetos, nos sistemas baseados em objetos, são caracterizados por três aspectos: identidade, características (estado interno) e comportamento.



**peso:** 200 g

**tamanho (raio):** 60 cm

**elasticidade:** alta

**cor:** vermelha

## Princípios do Paradigma



### Mensagem

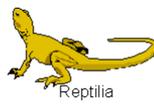
A comunicação entre os objetos é realizada através de mensagens. É através delas que um objeto solicita a outro que realize determinada tarefa.



## Princípios do Paradigma



### Classe



Quando realizamos uma classificação de objetos, identificamos o seu comportamento e as características que eles possuem em comum. Por este motivo, as classes definem:

- os atributos que irão descrever o objeto;
- os métodos que definem o comportamento dos mesmos

Classe	Objeto	Objeto	Objeto
			
peso	peso: 200 g	peso: 200 g	peso: 50 g
raio	raio: 60 cm	raio: 60 cm	raio: 30 cm
cor	cor: vermelha	cor: azul	cor: verde

## Princípios do Paradigma

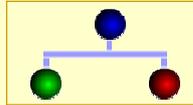


### Encapsulamento

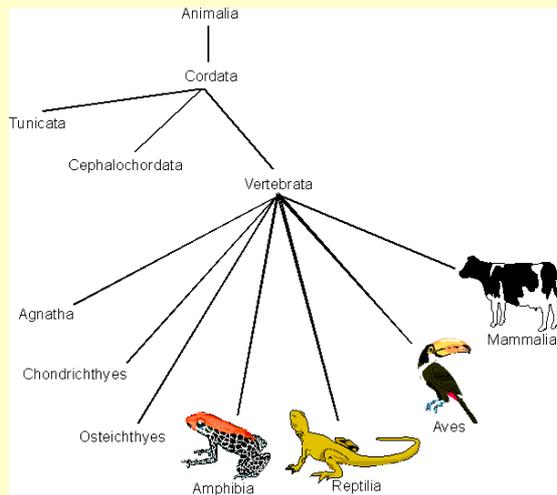
“Na programação orientada a objetos, os objetos comunicam-se entre si através de mensagens. A única coisa que um objeto conhece sobre outro objeto é a sua interface de comunicação. Os dados e a lógica de cada objeto são mantidos escondidos dos outros objetos. Em outras palavras, a interface encapsula o código e os dados do objeto”. [IBM]

- **Interface:** partes do objeto que se relacionam com o exterior;
- **Implementação:** dados e código que implementam o comportamento do objeto

# Princípios do Paradigma

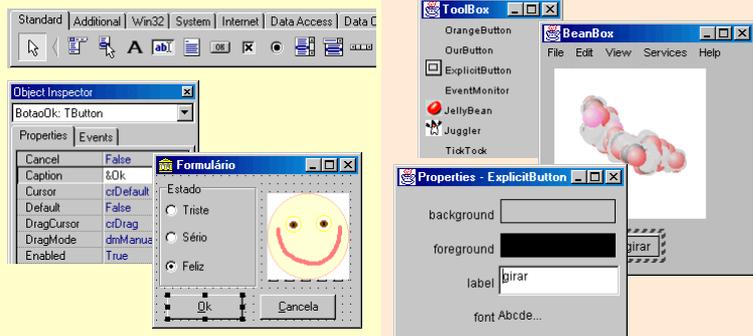


## Herança



A herança é o mecanismo utilizado para se definir uma nova classe de objetos a partir de uma “imitação, refinamento ou combinação” [MEY97]

# Componentes



- Têm a capacidade de encapsular dentro de um módulo uma coleção de programas interligados entre si, formando um elemento autônomo.
- Facilmente interligáveis para a construção de um modelo mais complexo

“Somente poucas pessoas podem construir seu próprio sistema estéreo interligando transistores e outros componentes no nível do circuito, mas muitas pessoas podem montar um sistema estéreo adequado a suas necessidades, ligando cabos ao seu amplificador, receiver, disc player, prato e alto-falantes preferidos” [ROS98]

# Agentes

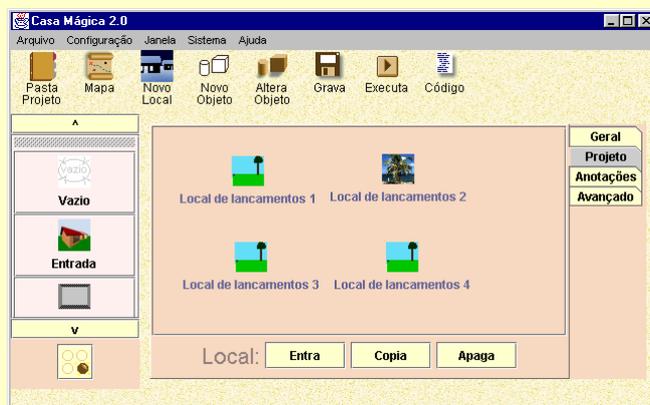
## Perspectivas diferentes:

“Cada agente mental, individualmente, só pode realizar tarefas simples, que não exijam, de modo algum, qualquer mente ou pensamento. Contudo, ao agruparmos esses agentes em sociedades – de maneiras muito especiais – acabamos nos deparando com a verdadeira inteligência”. [MIN87]

“A autonomia de um agente requer inteligência devido à necessidade de sobreviver em um ambiente real, dinâmico e nem sempre benigno. Está intimamente relacionada com o tipo de arquitetura do sistema; assim, podemos ter agentes com diferentes graus de autonomia”. [GIR99]

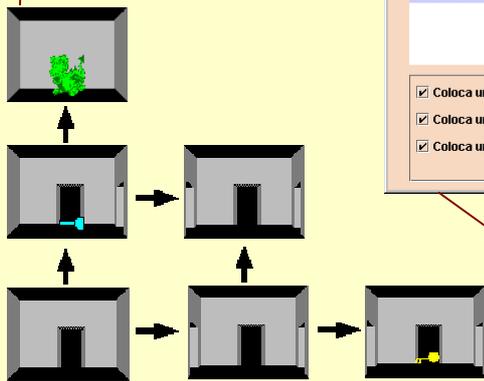
# Casa Mágica

Constitui-se em um ambiente para construção de aplicações educacionais que combina os recursos de uma ferramenta de autoria com recursos que permitem a construção e exploração de modelos de estudo.



# Casa Mágica

Locais de um Mundo interligados por Passagens



Construção de Unidade Nova

Sala

Descrição  
Sala de Entrada

Identificação  
SalaEntrada

Escolha um tipo  
Sala desenhada a mão  
Sala com efeito tridimensional

Coloca uma porta a esquerda  
 Coloca uma porta frontal  
 Coloca uma porta a direita

Ok Cancela

Assistente para construção de Locais

Visor

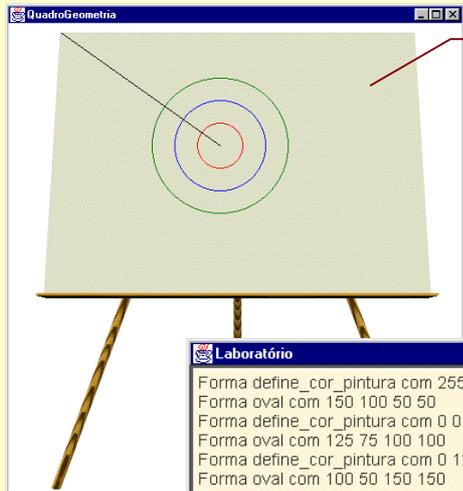
# Casa Mágica

Objetos do Local

Local

Classes

## Casa Mágica



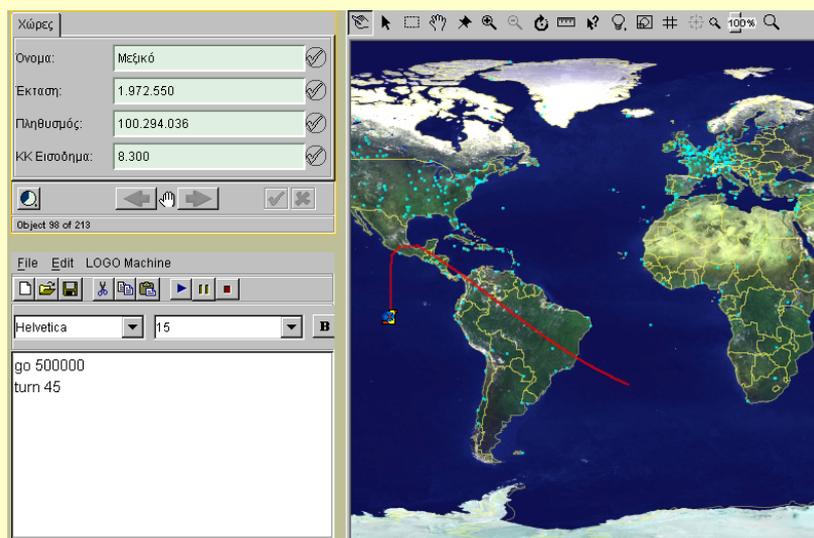
Componente Educacional que permite a construção dinâmica de formas geométricas.

Instruções emitidas para o componente (laboratório).

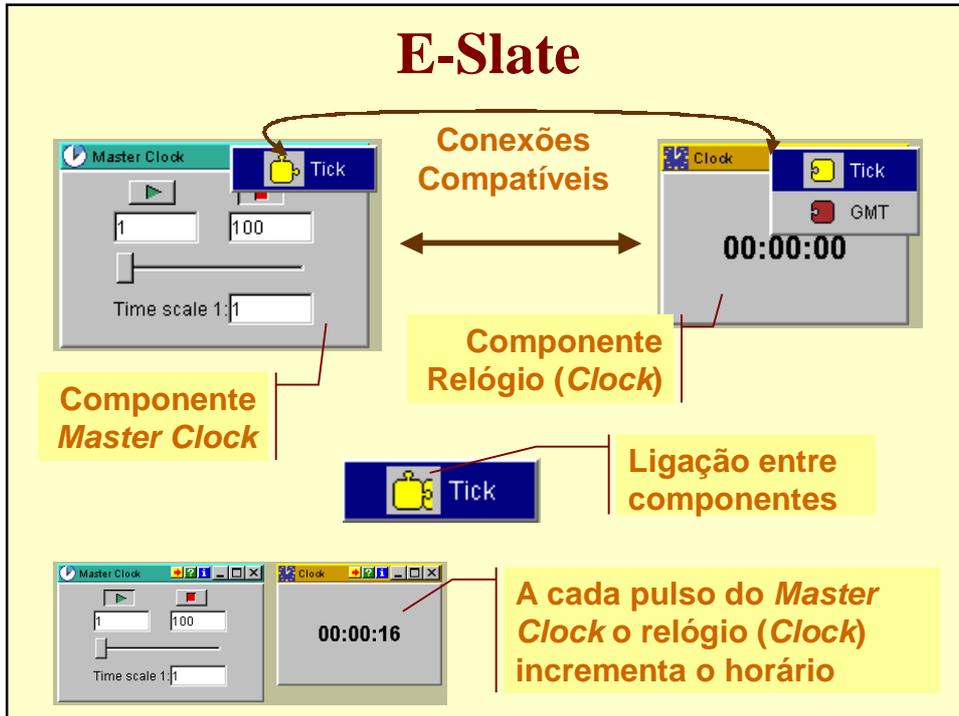


## E-Slate

O E-Slate permite a construção de micro-mundos utilizando uma biblioteca de componentes educacionais.



## E-Slate

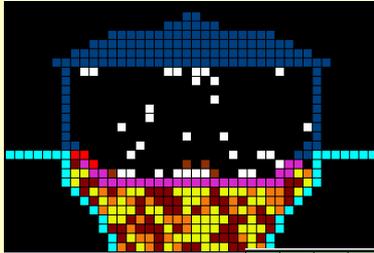


## E-Slate

Os componentes podem ser diretamente combinados em páginas Web e seu comportamento pode ser descrito através de uma linguagem de script baseada no LOGO.

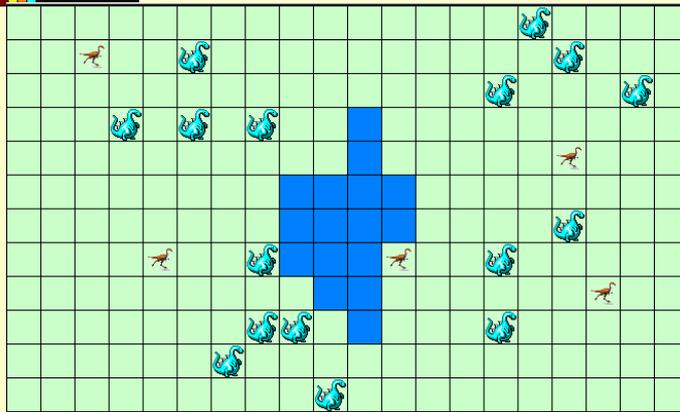


# Homos



Ferramenta de software para modelagem e simulação baseado em Objetos e Regras.

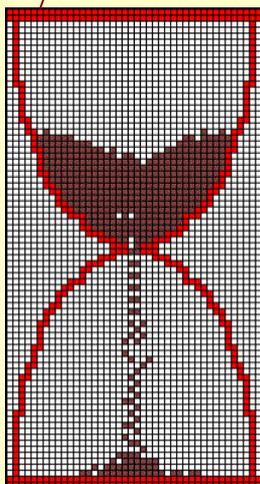
Combina princípios de Autômatos Celulares e Tecnologia de Objetos.



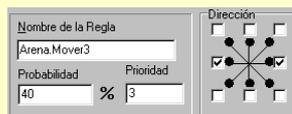
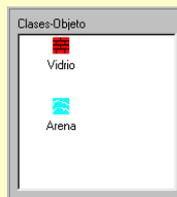
Objetos e Simulação

# Homos

Classes



Regras de Movimento da Areia



# Homos

**Clases Objeto**

- Conejo
- Lobo

**Reglas**

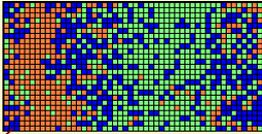
- Conejo.Mover
- Lobo.Mover
- Conejo.Reproducir
- Lobo.Reproducir
- Conejo.Morrir
- Lobo.Morrir
- Lobo.Comer

**Clases**

**Reglas**

**Objetos e Simulação**

**Gráfico da Simulação**



**Diagrama de Relação**

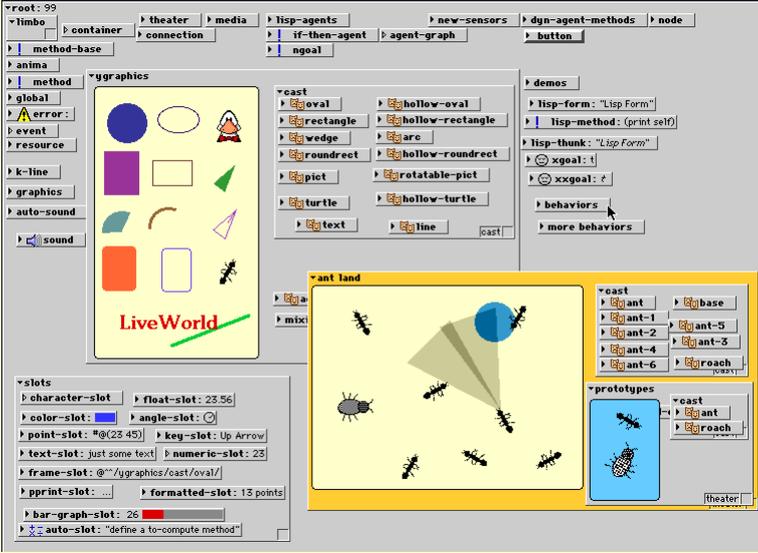


**Gráfica de la Simulación**



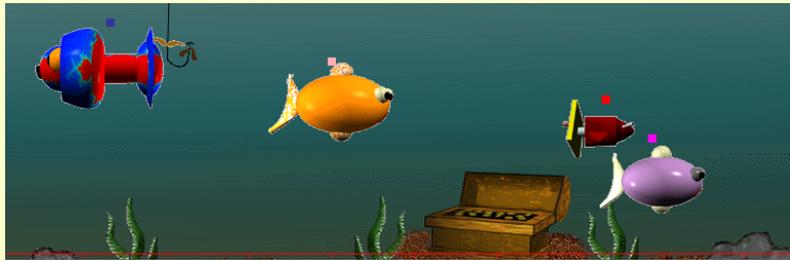
# LiveWorld

Ambiente de programação para sistemas animados baseados em agentes.



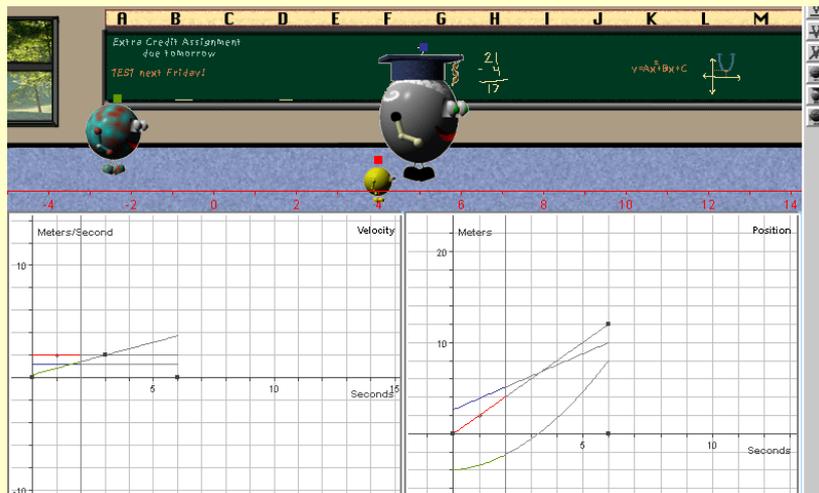
## MathWorlds

“[...] provê uma coleção de componentes de software, incluindo um conjunto de mundos de animação e uma variedade de gráficos. Atores nos mundos (como um palhaço, ou um pato) movem-se de acordo com funções matemáticas.”. [RSC98]



## MathWorlds

“Gráficos mostram estas funções matemáticas e permitem aos alunos editarem diretamente as funções”. [RSC98]



# MathWorlds

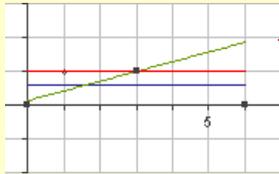


Gráfico  
Velocidade x Tempo

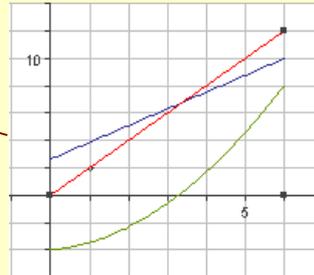


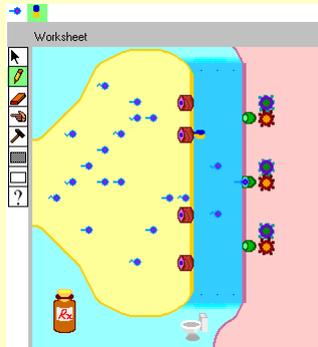
Gráfico  
Distância x Tempo



Atores animados  
(deslocamento  
conforme o gráfico)

# AgentSheets

“AgentSheets é um ambiente de autoria baseado em agentes destinado a educadores para construir simulações que podem se transformar em applets Java e JavaBeans”. [ROS99]

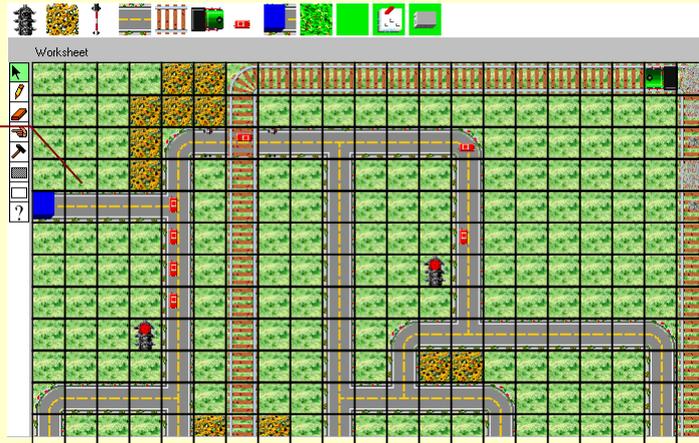


# AgentSheets

Agentes



Espaço Discreto



# AgentSheets

Visual AgentTalk

Comportamento

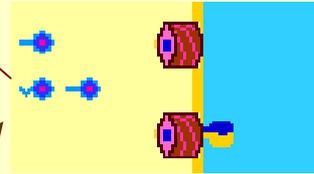


Agentes e Simulação

# AgentSheets

Visual AgentTalk

Agentes e Simulação



Comportamento

# StageCast

Baseados em agentes cuja atuação se desenvolve em espaço e tempo discretos. Sua abordagem é especialmente adequada para a realização de atividades pedagógicas que usam simulação.



# StageCast

**Agentes e Simulação**

**Bee**

do first  
Move onto flower

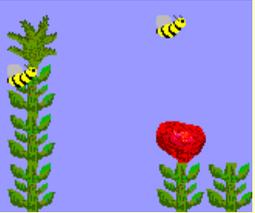
- From left
- From lower left
- From upper left

do random  
Move randomly

**Stalk**

do random  
Grow

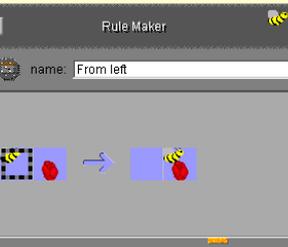
- Stalk
- Flower
-



**Comportamento**

**Rule Maker**

name: From left



**Rule Maker**

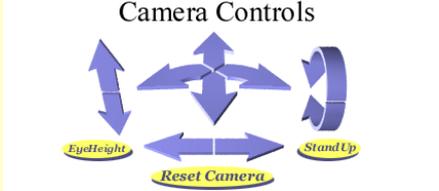
name: Stalk



# Alice



**Camera Controls**



**Ambiente tridimensional para a construção de animações com propósitos educacionais.**

**Utiliza a Tecnologia de Objetos.**

```

AliceLiddell
├── .arm
│   ├── .rshouldr
│   ├── .l4arm
│   └── .rhand
├── .larm
│   ├── .l4arm
│   ├── .lhand
│   └── .lshouldr
├── .neck
├── .head
├── .hair
├── .dress
│   ├── .rtigh
│   └── .lhigh
    
```

# Alice

The screenshot shows the Alice software interface with several components:

- Objetos da Cena:** A list of objects in the scene, including camera, Light, Ground, Pterodactyl (RightWing, LeftWing), island, palmtree, and three coconut objects.
- Cena:** A 3D view of the scene showing a palm tree on an island with coconuts.
- Navegação no Espaço:** Camera controls including directional arrows, 'Reset Camera', 'Stand Up', and 'Eye Height' buttons.
- Operações:** A list of mouse actions: Moves object, Raises/lowers, Turns left/right, Turns forward/back, Tumbles, and Orbits object.
- Transformações com o Objeto:** Buttons for 'Undo', 'Teach Me', 'Add Object', 'Make Object', 'Add Sound', and 'Add 3D Text'.

# Alice

The screenshot shows the Alice software interface with scripts and an event table:

- Scripts associados a Objetos (comportamento):** A list of scripts for the Pterodactyl object, including 'DesceAsa = DoTogether', 'SobeAsa = DoTogether', and 'Movimento = DoInOrder'.
- Codificação "arrastando e soltando":** A visual representation of the scripts using colored buttons (Undo Action, Teach Me, Make Event, Reset, Start).
- Tabela de Eventos:** A table defining the conditions and actions for the scripts.

When	Happens To	Do Animation
World Start	---	Movimento
LeftMouseButtonDown	Pterodactyl	DesceAsa
RightMouseButtonDown	Pterodactyl	DesceAsa