

O Simulador Ciber-Rato e o Concurso Micro-Rato

Luís Paulo Reis

lpreis@fe.up.pt

<http://www.fe.up.pt/~lpreis>

LIACC – Lab. Inteligência Artificial e Ciência de Computadores
FEUP – Faculdade de Engenharia da Universidade do Porto



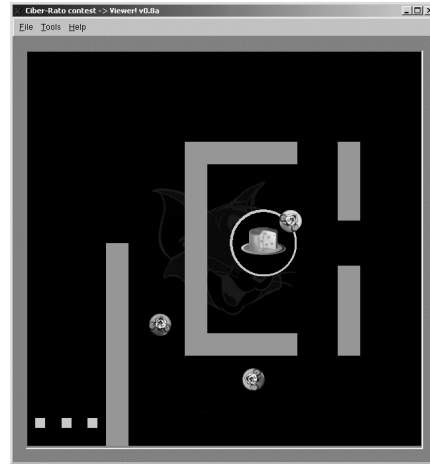
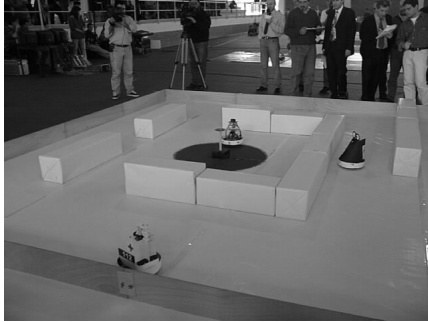
Estrutura da Apresentação

- Introdução
- Concurso Micro-Rato
- Descrição do Simulador Ciber-Rato
- Concurso Ciber-Rato
- Conclusões



Ciber-Rato

Robos Virtuais "Perdidos" num Labirinto



LIACC-FEUP

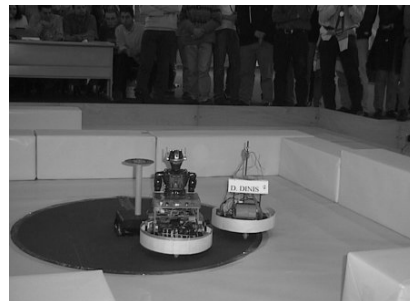
Luis Paulo Reis / 2003

Slide Nº 3



Micro-Rato

O que e ?



LIACC-FEUP

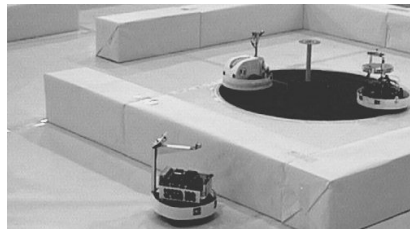
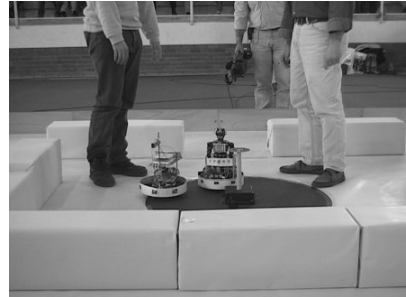
Luis Paulo Reis / 2003

Slide Nº 4



Micro-Rato

O que é? ◊



LIACC-FEUP

Luis Paulo Reis / 2003

Slide Nº 5



Micro-Rato

O que é? ◊

- Concurso de Robótica Móvel para robôs autónomos
- Organização da Universidade de Aveiro (DETUA)
- 7 Edições já realizadas (8ª edição no dia 23 de Maio de 2003) – (de 1995 a 2002)
- Pequenos robôs: 300x300x400mm (CxLxA)
- Mais Informação: <http://microrato.ua.pt>

LIACC-FEUP

Luis Paulo Reis / 2003

Slide Nº 6



Ciber-Rato

O que é? ▸



LIACC-FEUP

Luis Paulo Reis / 2003

Slide Nº 7

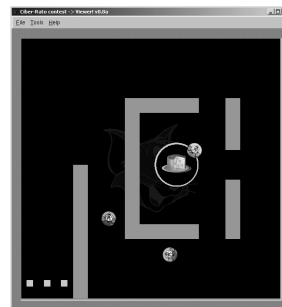


Ciber-Rato

O que é? ▸

**É a Modalidade de Simulação do
Concurso *Micro-Rato* da
Universidade de Aveiro**

**As equipas constroem o algoritmo
de controlo (*em C, C++ ou Delphi*) que
comanda um robô virtual**



Objectivo

Resolver um Labirinto

LIACC-FEUP

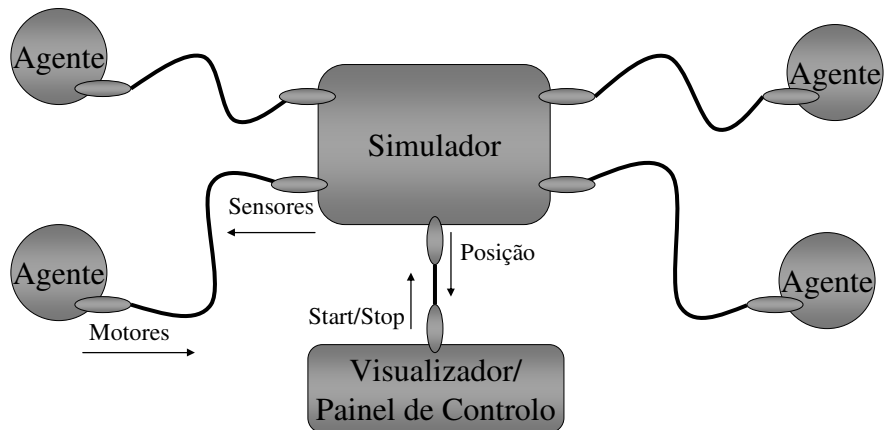
Luis Paulo Reis / 2003

Slide Nº 8



Ciber-Rato

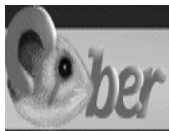
Como funciona ?



LIACC-FEUP

Luis Paulo Reis / 2003

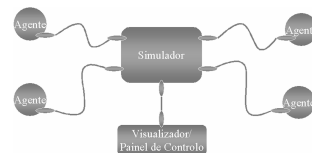
Slide Nº 9



Ciber-Rato

Simulador

- **Implementa labirinto virtual**
- **Implementa corpo virtual do robô**
- **Coordena movimento dos robôs no labirinto em articulação com agente**
- **Desempenha o papel de juiz virtual da prova:**
 - **Controlando tempo de prova**
 - **Aplicando penalizações**
 - **Calculando as pontuações**



LIACC-FEUP

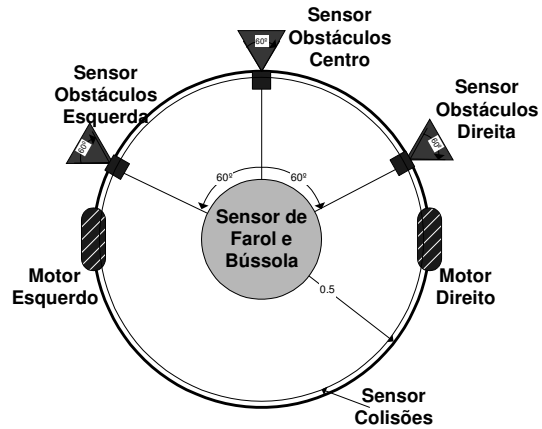
Luis Paulo Reis / 2003

Slide Nº 10



Ciber-Rato

O robo virtual



LIACC-FEUP

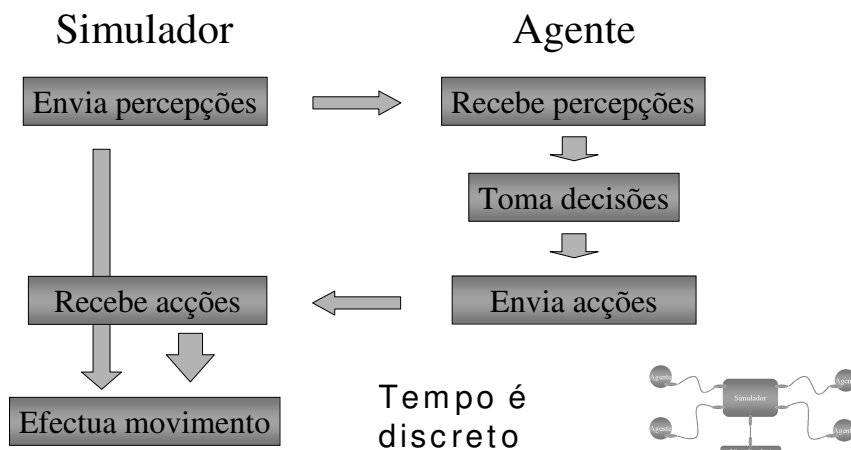
Luis Paulo Reis / 2003

Slide Nº 11



Ciber-Rato

Simulador <-> Agente



LIACC-FEUP

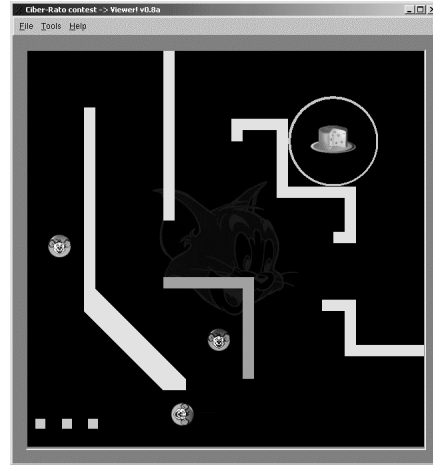
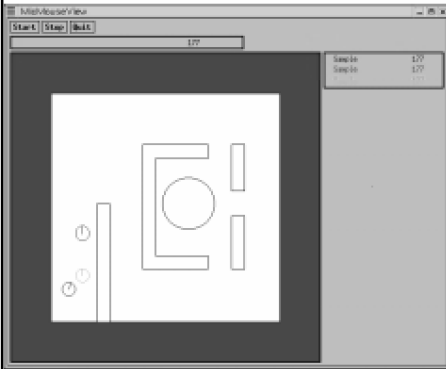
Luis Paulo Reis / 2003

Slide Nº 12



Ciber-Rato 2003

Visualizadores (2001 e 2002)



LIACC-FEUP

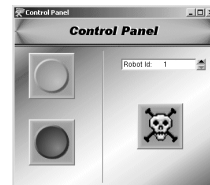
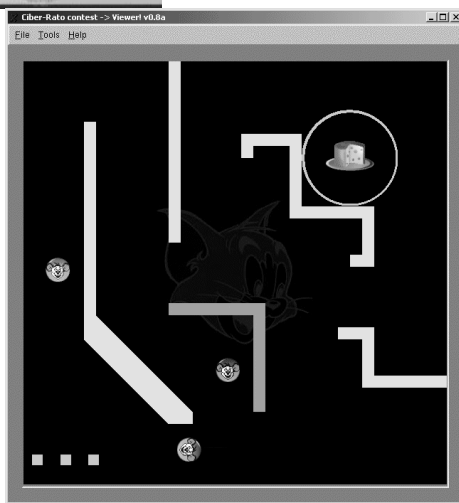
Luis Paulo Reis / 2003

Slide Nº 13



Ciber-Rato

Visualizador / Painel de Controllo



generated by sctchib

Time		Score	Collisions
Actual	1393		
Total	1800		
<hr/>			
	Tazmania	3793	24
	Speedy	2393	10
	Indo-Lazego	1393	0

Universidade de Aveiro Depart. Electrnica e Telecomunicaoes

LIACC-FEUP

Luis Paulo Reis / 2003

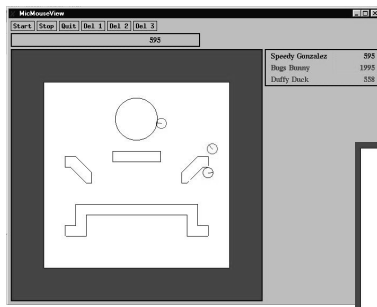
Slide Nº 14



Ciber-Rato

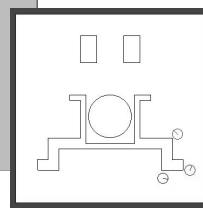
Provas realizadas (2001) 23 de Maio

Equipas: 4 a correr em Windows e 1 em Linux

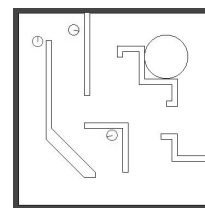


Manga 1

Manga 2



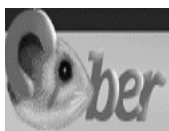
Final



LIACC-FEUP

Luís Paulo Reis / 2003

Slide Nº 15



Ciber-Rato

Provas realizadas (2002) - 26 de Abril

Equipas: 5 a correr em Windows e 8 em Linux

Proveniências: 2 de secundárias, 5 de politécnicos e 6 de universidades



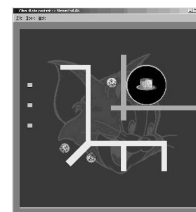
Manga 1



Manga 2



Manga 3



Final

LIACC-FEUP

Luís Paulo Reis / 2003

Slide Nº 16



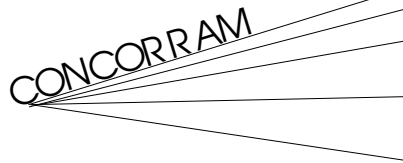
Ciber-Rato 2003

Novos desafios: labirintos maiores, regresso ao ponto de partida, erros maiores, eliminação da bússola

Novas ferramentas: simulador e visualizador em ambiente Windows

<http://microrato.ua.pt>

(<http://www.ieeta.pt/robocup>)



LIACC-FEUP

Luís Paulo Reis / 2003

Slide Nº 17



Ciber-Rato 2003

- Ferramentas
 - Simulador Linux / Windows
 - Visualizador Linux / Windows
 - Agente Reactivo Simples
 - Agentes Deliberativos
 - Editor de Mapas
 - Interface para Prolog

LIACC-FEUP

Luís Paulo Reis / 2003

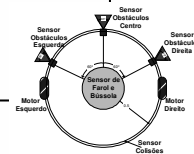
Slide Nº 18



Ciber-Rato 2003

Sensores dos Robôs Virtuais

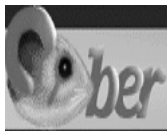
Tipo	Descrição	Ruído
obstáculos	medida inversamente proporcional à distância ao obstáculo mais próximo (dentro da área de sensibilidade do sensor). Varia entre 0.0 e 100.0 com uma resolução de 0.1.	Tem
farol	medida do angulo entre a frente do robot e a direcção do farol. Varia entre -180.0 e +180.0 com uma resolução de 1 grau.	Tem
colisão	sensor binário: valor igual a 1 quando o movimento do robot origina uma colisão.	Não tem
chão	sensor binário: valor 1 quando o robot se encontra totalmente dentro da área de chegada.	Não tem
bússola	indica a orientação do robot relativamente ao semi-eixo positivo das abcissas. A origem considera-se no canto inferior esquerdo do labirinto. Varia entre -180.0 e +180.0 com uma resolução de 1 grau.	Tem



LIACC-FEUP

Luís Paulo Reis / 2003

Slide Nº 19

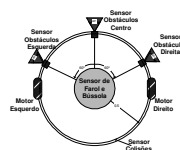


Ciber-Rato

Actuadores e Botões

Tipo	Descrição	Ruído
Motores	controle da potência do motor. A velocidade é proporcional à potência. Potência varia entre -0.1 e 0.1 com uma resolução de 0.001.	Tem
Led Fim	signalizador de fim de prova. Deve ser activado quando o robot se imobiliza na área de chegada.	Não tem

Botão	Descrição
Start	É utilizado no inicio da prova ou para reiniciar a prova após uma interrupção.
Stop	É utilizado no caso de ser necessária uma interrupção de prova.



LIACC-FEUP

Luís Paulo Reis / 2003

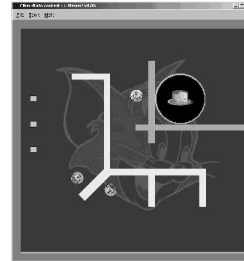
Slide Nº 20



Ciber-Rato 2003

Regras - Labirinto

- Área de jogo 17.5x17.5m
- Paredes Altas e Baixas com disposição desconhecida
- Posições de Partida Desconhecidas
- Área de chegada negra (raio mínimo de 1.5m) com farol no centro



LIACC-FEUP

Luís Paulo Reis / 2003

Slide Nº 21



Ciber-Rato 2003

Regras - Competição

- Objectivo: Chegar do ponto de partida à área de chegada
- Tempo limite: 3 minutos
- Mangas com 3 robôs
- Penalizações de 10 segundos por choques (com as paredes e com os outros robôs)

LIACC-FEUP

Luís Paulo Reis / 2003

Slide Nº 22



Ciber-Rato Instalação

```
tar xvfz MicMouseServer.tgz
```

e cria as seguintes directorias:

MicMouseServer - directoria onde se encontra o simulador `MicMouseServer`, o visualizador `MicMouseView` e alguns scripts de inicialização.

MicMouseServer/SampleRobot - directoria onde se encontra o código fonte de um robot reactivo simples e a biblioteca `libRobSock.a` que contém as primitivas C de comunicação dos robots com o simulador.



Ciber-Rato Modelo de Programação

```
void InitRobot(char *rob_name, char *host)
```

Realiza as operações de inicialização do robot, estabelecendo a ligação com o simulador. O simulador pode encontrar-se num outro computador (acessível através da rede) de nome `host`.

```
int ReadSensors(void)
```

Lê os novos valores dos sensores. Esta função espera até que o robot receba uma mensagem com os valores dos sensores, armazenando os valores medidos internamente. Retorna -1 em caso de erro.



Ciber-Rato Modelo de Programação

Leitura dos Valores dos Sensores:

```
unsigned int  GetTime(void)
float  GetObstacleSensor(int pos)
float  GetBeaconSensor(void)
float  GetCompassSensor(void)
bool   GetGroundSensor(void)
bool   GetBumperSensor(void)

bool   GetStartButton(void)
bool   GetStopButton(void)
```

LIACC-FEUP

Luis Paulo Reis / 2003

Slide Nº 25



Ciber-Rato Modelo de Programação

Actuação e Controlo da Prova:

```
bool GetFinished(void)
```

Função que permite ao robot verificar se a prova está terminada.

```
void DriveMotors(float LeftPower, float RightPower)
```

Função de comando dos motores. Envia ao simulador uma mensagem com os valores da potência nos motores esquerdo e direito.

```
void Finish(void)
```

Função de finalização de prova. Os robots devem chamar esta função quando estiverem imobilizados dentro da área de chegada.

LIACC-FEUP

Luis Paulo Reis / 2003

Slide Nº 26



Ciber-Rato Modelo de Programação

Envio de mensagem com a seguinte estrutura para a porta 6000 do simulador:

```
#define NEWROBOT_COMM      10
#define OK_COMM            25

typedef struct {
    unsigned short comm;
    union {
        char rob_name[20];
    } u;
} CommMessage;
```

LIACC-FEUP

Luis Paulo Reis / 2003

Slide Nº 27



Ciber-Rato Modelo de Programação

```
typedef struct {
    unsigned short time;
    unsigned short left,right,center;
    // (unsigned short) (medida*10.0 + 0.5)
    unsigned short light,compass;
    // (unsigned short) (medida_em_graus+360.5)
    unsigned short ground;
    unsigned short collision;
    unsigned short start;
    unsigned short stop;
    unsigned short endLed;
} SenseMessage;

typedef struct {
    unsigned short lPow, rPow;
    // (unsigned short) (potencia*1000.0 + 2000.5)
    unsigned short end;
    // activar para terminar a prova
} ActMessage;
```

Sensores e Actuadores:

LIACC-FEUP

Luis Paulo Reis / 2003

Slide Nº 28



Ciber-Rato Aproximações do Simulador

- Tempo Discreto
- Movimento Simples

```
Lin = (rPower * N1 + lPower * N2) / 2.0;  
Rot = ((rPower * N1 - lPower * N2) / 2.0) / ROBOT_RADIUS;
```

- Colisões Simples
- Valor dos Sensores inversamente proporcional à distância
- Ruído uniforme

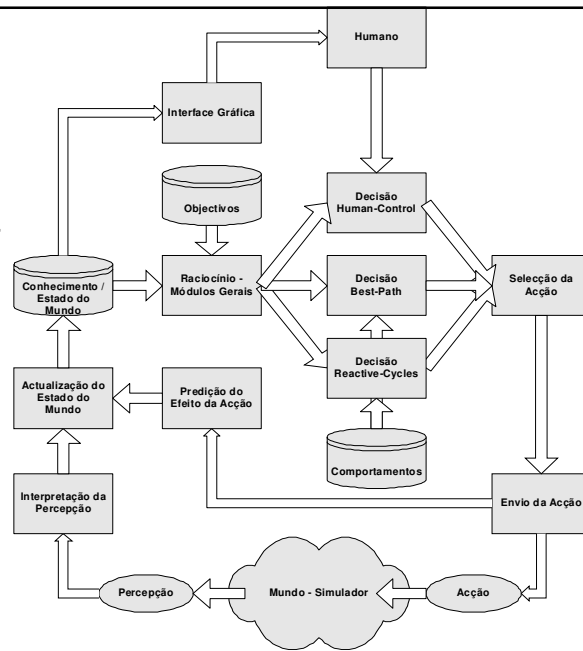


Ciber-Rato Alguns Desafios

- Como definir a arquitectura de controlo do robô?
- Comportamentos básicos?
 - Seguir parede? Seguir farol? Procurar farol? Vaguear? Iniciar? Passagem Estreita? Finalizar?
- Algoritmos de navegação em ambientes conhecidos/desconhecidos?
- Estratégia de mapeamento?
- Método de resolução do labirinto?
- Estratégia de controlo do robô?
- Tratamento sensorial (cuidado com os erros)?



- Arquitectura de um agente:
- Ciber-FEUP



LIACC-FEUP

Luís Paulo Reis / 2003

Slide Nº 31



Robot Name: Ciber-FEUP	Server IP Address: 192.168.102.32	Perception		
IP Address: 192.168.102.31	Server Initialization Port: 6000	Time: 560	Collision: No	
Robot Port: 6001	Server Communication Port: 1031	Left: 2	Center: 3	
520) LIGHT Right App (100, 58) 0	Sensors: 520 (4.1,13) [-21,-27] (No No)	Right: 9	Light: -8	
521) LIGHT Right App (100, 64) 0	Sensors: 521 (6.4,11) [-18,-28] (No No)	Compass: 48	Ground: Yes	
522) LIGHT Right App (100, 67) 0	Sensors: 522 (2.2,14) [-15,-33] (No No)	Start: No	Stop: No	
523) LIGHT Right App (100, 70) 0	Sensors: 523 (6.1,10) [-10,-37] (No No)	End Led: Yes	Action	
524) LIGHT Right App (100, 73) 0	Sensors: 524 (3.5,8) [-14,-33] (No No)	Command Mode: <input type="radio"/> Manual <input checked="" type="radio"/> Automatic	332	514
525) LIGHT Right App (100, 76) 0	Sensors: 525 (4.5,9) [-13,-43] (No No)	Automatic Command	Action in Execution: End	
526) LIGHT Right App (100, 79) 0	Sensors: 526 (1.5,8) [-5,-40] (No No)	Manual Command	Forward	Action: Light
527) LIGHT Right App (100, 82) 0	Sensors: 527 (3.5,6) [-9,-38] (No No)	Left Pow: Backward	Right Pow: End:	Left Stop Right End
528) LIGHT Right App (100, 85) 0	Sensors: 528 (3.3,10) [-9,-38] (No No)	Move: 0	0	1
529) LIGHT Right App (100, 85) 0	Sensors: 529 (3.3,8) [-9,-41] (No No)	World State		
530) End (0, 0) 1	Sensors: 530 (3.6,7) [-10,-49] (Yes No)	Left: 2	Center: 3	Right: 9

Send Init

Close

Restart

Best Path

Show Navigation

Clear Navigation

LIACC-FEUP

Luís Paulo Reis / 2003

Slide Nº 32