

Ciber-Rato: Uma Competição Robótica num Ambiente Virtual

Nuno Lau

Artur Pereira

Andreia Melo

António Neves

João Figueiredo

Universidade de Aveiro - IEETA

Campo Universitário de Santiago, Aveiro

{[lau](mailto:lau@det.ua.pt), [artur](mailto:artur@det.ua.pt), [andreaia](mailto:andreaia@det.ua.pt)}@det.ua.pt, an@ieeta.pt, a13470@alunos.det.ua.pt

Sumário

Este artigo descreve a modalidade Ciber-Rato do Concurso Micro-Rato da Universidade de Aveiro. Nesta modalidade as provas decorrem num ambiente virtual criado por um simulador. O sistema de simulação é distribuído e baseado em tecnologias abertas. Os concorrentes desenvolvem um agente robótico, que funciona como o cérebro de um robot virtual, e que deve ser capaz de resolver um labirinto.

Palavras-chave

Simulação, Mundos Virtuais, Competições robóticas, Agentes, Visualização

1 INTRODUÇÃO

A utilização de competições como meio de promover a investigação ou facilitar a aprendizagem em áreas tecnológicas tem sido adoptada por várias instituições [Murphy01]. As competições estabelecem um objectivo claro para as equipas concorrentes que é vencer o concurso. No entanto, o ambiente criado pelo concurso tem diversos aspectos que o tornam interessante sob o ponto de vista pedagógico:

- A existência de regras claramente definidas e iguais para todos os concorrentes;
- A aplicação prática de conhecimentos adquiridos em ambiente real;
- O estímulo pela autonomia e pela criatividade dos elementos da equipa;
- O desenvolvimento de um sistema que integra muitas vezes conhecimentos multi-disciplinares;
- O trabalho em grupo, uma vez que em geral as equipas concorrentes são constituídas por mais do que 1 elemento;
- A necessidade de se apresentar no dia do concurso com algo que funcione.

Sendo, por todos os pontos apontados anteriormente, interessantes do ponto de vista pedagógico, este tipo de competições contém também alguns perigos. Tal como é referido em [Stone03], a competição pode tornar obsessiva a vontade de vencer o que pode conduzir a comportamentos pouco desportivos, ao desenvolvimento de soluções demasiado dependentes do domínio ou à exploração activa de falhas nas regras do concurso. Certos concursos que se repetem todos os anos podem tornar a entrada de novos concorrentes demasiado difícil.

O perigo associado à generalização incorrecta dos resultados também deve ser evitado pois a aferição científica da qualidade das soluções propostas por cada equipa não é o objectivo destas competições. A quantidade de provas em que cada equipa participa não é suficiente para determinar de forma absolutamente segura o seu valor relativamente aos seus concorrentes.

Existem actualmente diversas competições robóticas, decorrendo com elevado grau de regularidade e aplicando robots a diversas áreas temáticas. A nível internacional destaca-se o RoboCup [RoboCup04, Sklar02], onde o tema é o futebol e o desafio a construção de equipas de robots futebolistas. A nível nacional cabe destacar o Festival Nacional de Robótica, o Micro-Rato e o Robot Bombeiro. O Festival Nacional de Robótica tem apostado nos temas Condução Autónoma e Futebol Robótico, incluindo vários tipos de competições para cada um. O concurso Robot Bombeiro [Bom04] tem como objectivo construir robots que resolvam um labirinto no sentido de localizar e apagar uma vela acesa.

O concurso Micro-Rato da Universidade de Aveiro é um concurso de robótica que coloca a cada concorrentes o desafio de construir um robot móvel e autónomo que resolva um labirinto desconhecido, no sentido de localizar e alcançar um farol nele colocado. Uma vez alcançado o primeiro objectivo o robot deverá regressar ao seu ponto de partida. A modalidade Ciber (Ciber-Rato) deste concurso apresenta-se como a versão simulada e destina-se a concorrentes que pretendam apenas dedicar-se às componentes algorítmicas e de programação dos robots. Os problemas de hardware estão resolvidos pois os robots são virtuais, existindo apenas no interior de um simulador comum a todas as equipas. Os concorrentes participam com um programa de software, o agente robótico, que, com base em dados sensoriais recebidos do

simulador, deve comandar o movimento do robot no labirinto, no sentido de concretizar os objectivos do concurso. O concurso Ciber-Rato teve em 2004 a sua quarta edição, tendo vindo a crescer quer em termos de equipas inscritas quer em termos de intuições representadas. É de destacar o aumento de participação de equipas de Escolas Secundárias.

Neste artigo descrevem-se os passos de preparação, montagem e realização necessários à organização do concurso Ciber-Rato. A secção 2 apresenta a história breve do Ciber-Rato. A secção 3 apresenta uma visão geral do ambiente virtual de competição em que decorre o Ciber-Rato. A secção 4 inicia a apresentação das ferramentas de apoio ao concurso tratando dos aspectos gerais do sistema de simulação. A secção 5 apresenta o simulador utilizado pelo Ciber-Rato detalhando alguns dos aspectos da sua implementação. A secção 6 trata das questões associadas à visualização das provas do Ciber-Rato. A secção 7 apresenta alguns aspectos relacionados com o desenvolvimento de agentes robóticos. Finalmente, a secção 8 conclui o artigo.

2 HISTÓRIA DO CIBER-RATO

A primeira edição do Ciber-Rato decorreu em 2001, ia então o Concurso Micro-Rato já na sua 6ª edição. A introdução desta modalidade no concurso Micro-Rato surge como resultado indirecto da participação da equipa FC Portugal na Liga de Simulação do RoboCup [Reis01]. No decorrer desta participação foi verificado que a arquitectura geral do sistema de simulação usado pela Liga de Simulação do RoboCup [Noda98] podia ser utilizado para o desenvolvimento de um simulador para o Concurso Micro-Rato que permitisse a introdução, neste concurso, de uma modalidade de simulação.

Do simulador *soccerserver* foi aproveitada a ideia de um simulador comandado pelo tempo do tipo *cycle driven* e a arquitectura distribuída do sistema de simulação em que o simulador os agentes e o visualizador são aplicações independentes que comunicam através de *sockets* UDP. No entanto, a arquitectura interna e os modelos utilizados pelo simulador do Ciber-Rato são independentes do simulador do *soccerserver*.

Na edição de 2001 concorreram 5 equipas, 1 da Universidade de Aveiro e 5 da Universidade Fernando Pessoa. Nesta primeira edição do Ciber-Rato o desafio proposto aos concorrentes consistia apenas na ida até à área do farol.

Em 2002 realizou-se a 2ª edição, tendo concorrido 13 equipas de todo o país. Pela primeira vez o concurso teve a participação de equipas provenientes de escolas secundárias. Os robots mais bem classificados nesta edição foram convidados a publicar a sua estratégia em artigos que foram publicados na Revista Electrónica e Telecomunicações do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro [Ribeiro02,Reis02,Lomba02].

A 3ª edição do Ciber-Rato decorreu em Maio de 2003 e apresentaram-se a concurso 14 equipas. Neste ano o

desafio na modalidade Ciber-Rato foi estendido de modo a incluir também o desafio de regressar ao ponto de partida.

Na 4ª edição concorreram 13 equipas, sendo 6 provenientes de escolas secundárias e 5 de universidades. É de destacar o facto de esta edição ter sido ganha por uma equipa proveniente de uma Escola Secundária.

3 O AMBIENTE VIRTUAL DE COMPETIÇÃO

O ambiente virtual em que os robots estão inseridos é um mundo plano delimitado por uma parede exterior que forma um rectângulo de 28x14, que corresponde à área de jogo, no interior da qual existem diversas paredes que funcionam como obstáculos e limitam as trajectórias possíveis dos robots. As paredes são verticais relativamente ao chão e poligonais, enquanto os robots são circulares. Além dos obstáculos e dos robots existe no ambiente virtual uma área-alvo sinalizada por um farol. A altura dos obstáculos é variável, sendo que alguns são mais altos do que o farol e impedem a visão deste quando interpostos entre o robot e o farol.

O tempo é discreto e apenas avança no início de cada ciclo de simulação. As posições de todos os robots apenas são alteradas no início de cada ciclo de simulação permanecendo constantes durante este ciclo.

O ambiente é estruturado mas dinâmico, pois em cada prova concorre sempre mais do que 1 robot. O robot não conhece o labirinto antes da prova e a informação que lhe é fornecida pelos seus sensores apenas lhe dá uma perspectiva local e ruidosa do espaço à sua volta.

3.1 O robot virtual

O robot virtual é circular e constituído por vários sensores e actuadores (Figura 1).

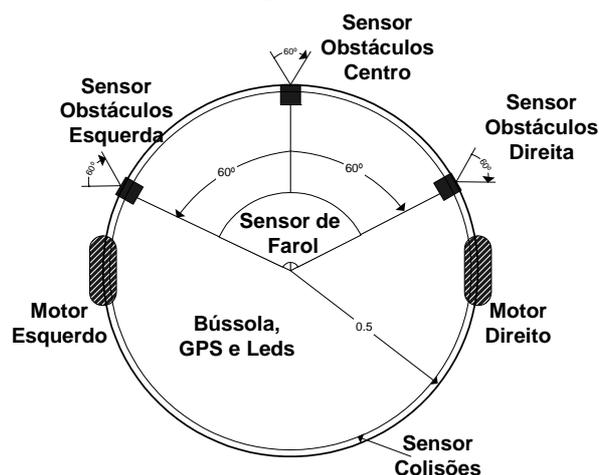


Figura 1: Robot Virtual

Cada robot está equipado com os seguintes sensores:

- 3 sensores de obstáculos – O valor medido por um sensor de obstáculos é inversamente proporcional à distância ao obstáculo mais próximo. O obstáculo pode ser outro robot ou uma parede, mas essa informação não é passada ao agente robótico. Este sensor tem ruído gaussiano.

- 1 sensor de farol – O valor medido indica a posição angular do farol relativamente ao eixo frontal do robot, sendo apenas detectado se o ângulo for menor ou igual a 60 graus. Este sensor tem ruído gaussiano.
- 1 bússola – O valor medido indica ao robot a direcção do norte virtual do labirinto. Este sensor tem ruído gaussiano.
- 1 sensor de área-alvo – É um sensor booleano que indica se o robot se encontra sobre a área-alvo.
- 1 sensor de colisões – É um sensor booleano que está activo sempre que o robot colide com outro robot ou com uma parede.
- 1 sensor GPS – Este sensor fornece ao robot a posição do seu centro e a sua orientação sem erros, apenas podendo ser utilizado para testes.

O robot dispõe ainda de alguns actuadores:

- 2 motores – Estes motores comandam o movimento de duas rodas, colocadas nas extremidades do diâmetro perpendicular ao eixo frontal do robot. É através do comando da potência destes motores que o robot se move. Os motores podem rodar para a frente ou para trás. Assim, por exemplo, para rodar sobre si próprio o robot deve colocar um dos motores a rodar para a frente e o outro a rodar para trás.
- 2 leds de sinalização – Estes leds permitem ao robot sinalizar a chegada à área-alvo e a finalização da prova.

3.2 O desafio

O desafio colocado às equipas participantes é o desenvolvimento de um agente robótico que, com base nas medidas obtidas pelos sensores, seja capaz de comandar um robot de forma a fazê-lo cumprir os seguintes objectivos em sequência:

1. Deslocar-se desde o ponto de partida até à área-alvo, assinalando o cumprimento desse objectivo.
2. Deslocar-se novamente até ao seu ponto de partida, assinalando a terminação da prova.

Os robots concorrem em labirintos desconhecidos. Em cada prova 3 robots tentam cumprir os mesmos objectivos, exceptuando as diferenças que resultam dos seus pontos de partida serem distintos.

Os robots são penalizados por cada colisão com paredes e pelas colisões com outros robots em que lhes seja determinada culpa.

3.3 O sistema de pontuações

A pontuação inicial de todos os robots é de 200 pontos. O objectivo do agente robótico é minimizar a sua pontuação.

Quando o robot cumpre o primeiro objectivo são subtraídos 100 pontos à sua pontuação. Ao mesmo tempo é determinada distância mínima, contornando obstáculos, entre o ponto de partida e o ponto em que assinalou a chegada ao farol. Durante o regresso a pontuação do robot é subtraída (ou aumentada) de 1 ponto sempre que

o robot se aproxime (ou se afaste) do seu ponto de partida de 1% daquela distância.

Por cada colisão a pontuação do robot é aumentada de 5 pontos.

Como factor de desempate foi utilizado, na edição de 2004, o tempo que o robot demora a regressar ao ponto de partida. Este factor de desempate pretende estimular as capacidades de aprendizagem e exploração do ambiente durante a fase de ida até ao farol de modo a que o regresso se faça tão rápido quanto possível.

O cumprimento de todas as regras e a resolução dos protestos apresentados pelas equipas são assegurados por um júri independente. As decisões do júri são soberanas. Só o júri pode aplicar as penas mais gravosas, tais como, a eliminação de um robot de uma prova.

4 ARQUITECTURA GERAL

A execução de uma prova do Ciber-Rato é apoiada por um sistema distribuído em que participam as seguintes aplicações:

- simulador;
- 1 ou mais visualizadores;
- 1 agente robótico por cada robot concorrente.

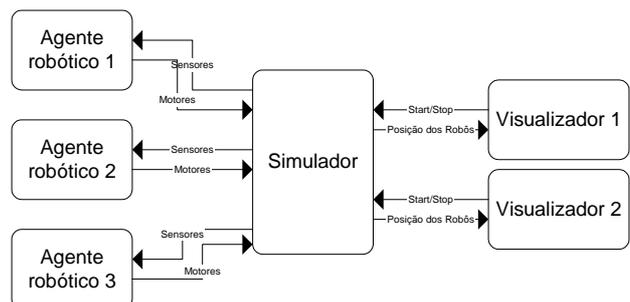


Figura 2: Arquitectura do sistema de Simulação

A arquitectura do sistema é do tipo cliente-servidor, em que o simulador funciona como servidor e os agentes robóticos e os visualizadores funcionam como clientes (ver figura 2).

A comunicação entre as aplicações é realizada através de mensagens XML enviadas através de sockets UDP.

A arquitectura do sistema permite que os agentes robóticos, o visualizador e o simulador possam residir em computadores distintos interligados em rede. Além disso, esta arquitectura permite que os agentes robóticos possam ser programados em qualquer linguagem de programação que permita a utilização de sockets, assim como utilizar um sistema operativo diferente daquele utilizado pelas outras aplicações.

A utilização de XML como formato base para a troca de mensagens pela rede torna o sistema mais aberto pois este é um formato padrão e de fácil manipulação.

O simulador e o visualizador são disponibilizados pela organização para os sistemas operativos Linux e Windows através do sítio do Concurso Micro-Rato

[MicroRato04]. A organização disponibiliza também um exemplo de um agente robótico reactivo para servir de modelo à construção de agentes. O simulador pode ser instruído para fazer a gravação de uma prova. A organização disponibiliza uma aplicação que permite reproduzir essas gravações, constituindo um óptimo auxiliar de teste.

5 O SIMULADOR

O simulador é a peça central das aplicações de suporte à execução de uma prova do Ciber-Rato. Funciona como servidor das restantes (agentes e visualizadores). As restantes aplicações ligam-se ao simulador enviando-lhe um pedido de conexão e aguardando autorização.

Antes de qualquer pedido de registo o simulador deve ser configurado para a prova que se vai iniciar. Esta configuração consiste na escolha do labirinto que vai ser utilizado, da grelha de partida dos robots, dos parâmetros de simulação (níveis de ruído, tempo total de prova, etc) e da gravação ou não da prova num ficheiro. O formato utilizado para a descrição dos labirintos, grelha de partida, parâmetros de simulação e ficheiros de gravação de prova é também baseado em XML.

Em cada ciclo de simulação o simulador actualiza as posições dos robots com base nos comandos recebidos pelos agentes e nas restrições impostas pelas paredes e outros robots, determina as pontuações, aplicando as penalizações apropriadas, determina quais os valores medidos por cada um dos sensores dos robots em prova e envia essa informação aos agentes robóticos, enviando ainda ao visualizador as novas posições e pontuações dos robots. Durante o ciclo de simulação o simulador recebe os comandos dos agentes, mas só no início do ciclo de simulação é que esses comandos são processados e as posições dos robots novamente alteradas (figura 3).

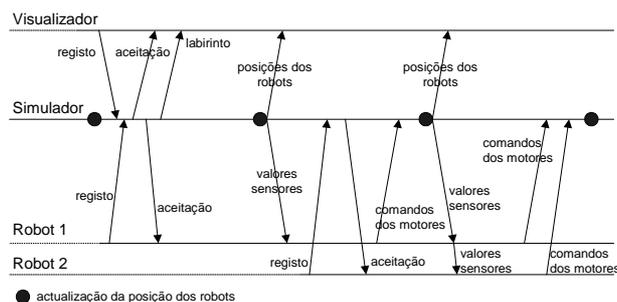


Figura 3: Mensagens trocadas entre simulador, agentes e visualizador

5.1 Funções

O simulador tem as seguintes funções:

- Implementa o ambiente virtual onde decorre a prova constituído pelo labirinto, o farol, a área-alvo, etc.
- Implementa o corpo virtual dos robots: todos os sensores e motores do robot “existem” no simulador, cabendo às equipas desenvolver apenas o “cérebro” ou algoritmo de controlo dos robots;

- Define e aplica as regras do modelo físico virtual, determinando as novas posições dos robots e as medidas de todos os seus sensores.
- Determina as pontuações dos robots, aplicando as penalizações quando for caso disso. O simulador funciona como árbitro automático que, por exemplo, decide de quem é a responsabilidade de uma colisão entre 2 robots.
- Envia aos visualizadores toda a informação necessária à visualização gráfica da prova e das pontuações dos robots.

5.2 Modelo físico

O simulador conhece internamente as posições de todos os robots e de todas as paredes. A passagem de um ciclo de simulação para o seguinte é activada por um Timer cujo período é um dos parâmetros da simulação.

A posição dos robots é determinada através da posição do seu centro e da orientação do seu eixo frontal. As paredes são armazenadas como sequências de segmentos de recta. A actualização da posição dos robots e a determinação de colisões é realizada através da aplicação de regras simples de geometria analítica.

O modelo de colisões é simplificado, pois a computação total de um choque entre mais do que 2 objectos é muito complexa. A abordagem seguida consiste em duas etapas. Na primeira, são determinadas as novas posições dos robots sem considerar a possibilidade de colisões. Na segunda fase, com base nas novas posições, são identificados os robots colidentes, e para estes são anuladas as componentes lineares do movimento, sendo, no entanto, mantidas as componentes de rotação.

5.3 Modelos dos Sensores

Os modelos dos sensores são em grande medida determinados através de cálculos de geometria analítica relativamente às posições dos robots e das paredes no interior do labirinto.

Nos casos em que é aplicado ruído gaussiano ao sensor, este é gerado através da forma polar da transformação de Box-Muller [Carter04] e alterando o resultado de forma a obter as médias e desvio padrão pretendidos.

5.3.1 Sensor de obstáculos

O sensor de obstáculos é um dos sensores mais importantes do robot. Estão colocados na periferia do robot e têm um ângulo de sensibilidade de 60 graus.

As medidas deste sensor são obtidas determinando a distância da parede mais próxima e do robot mais próximo que se encontra dentro do ângulo de sensibilidade do sensor. A medida do sensor é a distância mais curta invertida adicionada de ruído gaussiano.

5.3.2 Sensor de farol

O sensor de farol determina a direcção do farol relativamente ao eixo frontal do robot se este ângulo for

menor do que 60 graus e se não existirem paredes mais altas do que o farol entre este e o centro do robot.

A medida do sensor é obtida determinando o ângulo entre o farol e o centro do robot. Se o valor absoluto deste ângulo for maior do que 60 graus então o farol é considerado invisível. Caso contrário, é traçado um segmento entre o farol e o centro do robot e é verificado se esse segmento atravessa alguma parede de altura superior ao farol, caso em que a medida também se torna invisível. Se a medida é visível é-lhe aplicado ruído gaussiano.

5.3.3 Bússola

A bússola determina a orientação do robot relativamente a um norte virtual que está alinhado com o semi-eixo positivo dos x do labirinto.

A determinação desta medida é realizada simplesmente adicionando ao simétrico da orientação angular do robot ruído gaussiano.

5.3.4 Sensor de área-alvo

Este sensor é booleano. Conhecendo a posição do robot e a posição e raio da área-alvo, o simulador determina a medida deste sensor sem introduzir qualquer ruído.

5.3.5 Sensor de colisões

O sensor de colisões avisa o robot que este incorreu numa colisão. Os agentes robóticos sabem que nesse caso a componente linear do seu deslocamento foi anulada. É o simulador que determina todas as colisões e neste caso o valor determinado para cada robot é enviado ao seu agente robótico.

5.3.6 Sensor GPS

Este sensor apenas pode ser utilizado durante a fase de testes. Ele é activado por uma opção especial do simulador e nesse caso o simulador envia ao robot a posição do seu centro e a sua orientação sem erros. Durante as provas da modalidade Ciber-Rato este sensor está desligado.

5.4 Modelos dos actuadores

Os actuadores do robot são essencialmente os seus motores e os leds de sinalização de cumprimento do 1º objectivo e de terminação de prova.

5.4.1 Motores

O agente robótico pode aplicar a qualquer um dos motores uma potência que pode variar entre -0.15 e 0.15. A potência efectivamente aplicada à roda é determinada através de um filtro IIR com os seguintes parâmetros:

$$out_t = (in_t * 0.5 + out_{t-1} * 0.5) * noise$$

em que in_t é a potência enviada pelo agente robótico no ciclo t e out_t é a potência à saída do motor. Ao valor out_t é ainda aplicado ruído multiplicativo de distribuição gaussiana e média 1.

A actualização da posição do robot é efectuada através de uma aproximação de primeira ordem relativamente às potências de saída dos motores. O movimento do robot é dividido nos seus componentes linear lin e rotacional rot , sendo que:

$$lin = \frac{out_{right} + out_{left}}{2}$$
$$rot = \frac{out_{right} - out_{left}}{diam}$$

Ao robot é aplicado em primeiro lugar a componente linear deslocando-o segundo o sentido do seu eixo frontal e seguidamente a componente rotacional alterando a sua orientação.

5.4.2 Leds de Sinalização de eventos

Os leds de sinalização de eventos são comandados pelos agentes robóticos para a sinalização do cumprimento dos objectivos da prova. Ao receber estes comandos o simulador actua sobre a pontuação do robot e altera o seu estado interno no simulador. O estado interno do robot permite a gestão pelo simulador das pontuações e das acções possíveis do robot. Por exemplo, o calculo das distâncias mínimas apenas é necessário para os robots que já atingiram o primeiro objectivo.

Os agentes robóticos podem determinar se o seu comando de sinalização foi recebido pelo simulador pois na mensagem de medidas de sensores que é enviada ao robot em todos os ciclos estão incluídos o estado destes leds. O agente robótico pode determinar assim que a mensagem enviada não foi recebida, se o estado do led não for alterado, e reenviar a mensagem.

5.5 Determinação dos caminhos mínimos

A determinação dos caminhos mínimos entre o ponto de partida e a posição do robot ou o ponto de partida e a posição em que o robot assinalou o cumprimento do primeiro objectivo é algo complexa. Este calculo é realizado utilizando o algoritmo Dijkstra [Cherkassky94] para determinar caminhos mínimos sobre um grafo de visibilidades em que os nós são pontos do labirinto e os arcos indicam que é possível o trajecto entre essas posições, estando a cada arco associado um custo que é igual à distância a percorrer pelo robot para ligar os pontos origem e destino.

A construção do grafo de visibilidades é realizada considerando que todas as paredes foram inflacionadas de um valor igual ao raio do robot, definindo assim como zona livre todas as posições possíveis, sem colisões, de um robot no labirinto. Esta operação de expansão origina para cada canto de parede convexo a existência de um arco de circunferência, o que faz com que os nós do grafo de visibilidades associados sejam pontos tangentes a estas circunferências (Figura 4). Os cantos côncavos do labirinto não contribuem com pontos para o grafo de visibilidades.

Existe um grafo de visibilidades distinto para cada robot. Nesse grafo são incluídos todos os pontos determinados

pelas visibilidades tangentes referidas anteriormente e ainda os pontos de partida e a posição actual do robot. A inserção do ponto de partida e do ponto de chegada pode provocar a inserção no grafo de novos nós associados a novos pontos de visibilidade tangencial relativamente aos pontos inseridos. Como o robot se encontra em permanente movimento estes pontos adicionais têm de ser recalculados em cada ciclo de simulação.

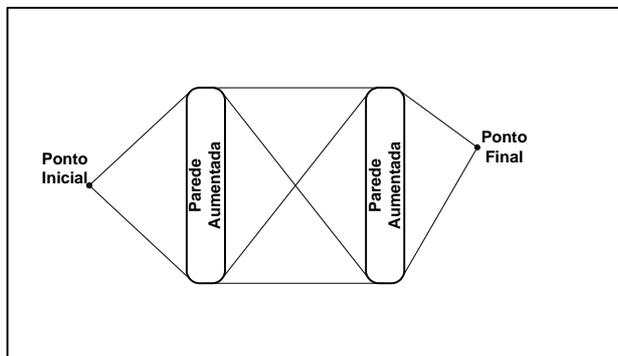


Figura 4: grafo de visibilidades, alguns nós são pontos tangentes aos arcos que resultam da expansão das paredes

A pesquisa no grafo utiliza uma lista ordenada dos nós com menor distância ao nó origem. A partir do elemento com menor distância são calculadas, ou recalculadas, as distâncias de todos os nós que são visíveis a partir deste. O processo é repetido até que o nó final seja atingido e que a distância mínima ao nó origem, que consta da lista de nós a tratar, seja maior ou igual à distância determinada até ao nó final.

5.6 Gravação de provas

O simulador permite a gravação da prova num ficheiro. Esse ficheiro pode ser utilizado por uma outra ferramenta denominada logplayer, que, lendo o ficheiro, emula um simulador e envia a todos os visualizadores registados a informação necessária à reprodução da prova.

O modelo utilizado para a reprodução da prova não necessita de qualquer alteração no código dos visualizadores.

5.7 Interface com o utilizador

A interface com o utilizador do simulador é realizada através da linha de comando, onde é possível especificar o labirinto, a grelha, os parâmetros de simulação e o ficheiro de gravação a utilizar, ou de forma interactiva através dos menus da aplicação simulador (figura 5). Estes menus permitem reiniciar uma prova, alterar o labirinto, a grelha ou os parâmetros de simulação. A aplicação do simulador inclui ainda na sua interface uma zona onde são escritas mensagens que indicam qual o sucesso ou insucesso dos comandos efectuados e a ocorrência de alguns eventos externos, tais como, o registo de um novo robot.

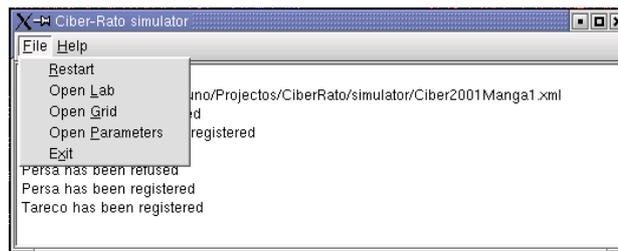


Figura 5: Janela do simulador

5.8 Implementação

O simulador é uma aplicação desenvolvida em C/C++, estando disponível em Linux e em Windows. No processamento das mensagens XML e na construção da interface gráfica foram utilizadas as bibliotecas Qt [TrollTech04].

6 A VISUALIZAÇÃO

A visualização das provas do Ciber-Rato é, por definição da arquitectura do sistema de simulação, realizada através de uma aplicação independente do simulador e cliente deste. O simulador permite o registo de vários visualizadores enviando a todos, sob pedido, o labirinto e a grelha de partida, e no início de cada ciclo de simulação as posições e pontuações de todos os robots.

Os visualizadores têm também a função de controlar a execução de prova. O simulador aceita os seguintes comandos dos visualizadores:

- Início da prova
- Paragem da prova
- Eliminação de um determinado robot

O comando de início da prova é executado a partir de um dos visualizadores registados após a verificação do correcto registo de todos os robots em prova. As paragens de prova podem acontecer, por exemplo, para consulta do júri sobre algum acontecimento estranho que ocorra durante a prova ou para resolver um protesto de concorrentes. A eliminação de um robot é uma pena que apenas pode ser aplicada pelo júri no caso de este julgar o comportamento de determinado robot como hostil.

6.1 Visualização 2D

A visualização das provas do Ciber-Rato é efectuada usando um visualizador, cujo desenvolvimento foi iniciado no âmbito da cadeira de Projecto da Licenciatura em Eng. Electrónica e Telecomunicações da Universidade de Aveiro [Neves02].

Este visualizador é constituído por 3 janelas:

- Área de jogo
- Painel de pontuações
- Painel de controlo

A área de jogo apresenta uma vista de cima da área de jogo. É possível visualizar os movimentos dos robots ao longo da prova, assim como a posição das paredes altas e baixas, da área-alvo e do farol (figura 6).



Figura 6: Visualizador 2D

O painel de pontuações apresenta características globais da prova, tais como o nome da prova e o tempo de simulação, e características específicas de cada robot, tais como o nome do robot, o ícone que lhe está associado, a fase de prova em que o robot se encontra (ida, regresso, terminou, etc), o tempo de regresso, o número de colisões e a pontuação global.

O painel de controlo permite a execução dos comandos de início, paragem e eliminação de um robot.

A organização de um concurso como o Ciber-Rato não pode ignorar um participante especial: o público. Nesse sentido é necessário não esquecer a espectacularidade em termos áudio-visuais. O visualizador 2-D permite a utilização de peles (*skins*) de modo a aumentar essa espectacularidade. Todos os anos é usada uma pele diferente. O tema da pele em 2004 foi os insectos, sendo o labirinto visualizado como um jardim onde existem muros (as paredes altas) e canteiros floridos (as paredes baixas) (ver figura 6). Além disso, o visualizador emite sons associados a eventos pré-determinados da prova, tais como o início, a chegada à área-alvo, a terminação de prova e a colisão. Imagens e sons associados a uma pele estão agrupados numa entidade coerente, de modo a permitir que o visualizador acomode facilmente novas peles.

O visualizador 2D está disponível em versões Linux e Windows. Esta aplicação foi desenvolvida em C/C++ utilizando extensivamente as bibliotecas Qt [TrollTech04] para a sua implementação. A utilização destas bibliotecas permitiu que o código seja compilável sem qualquer alteração em ambos os sistemas operativos.

6.2 Visualização 3D

Apesar de não fazer parte das ferramentas disponibilizadas aos concorrentes, foi desenvolvida uma aplicação, demonstrada durante o Ciber-Rato 2004, que permite a visualização da prova em 3 dimensões.

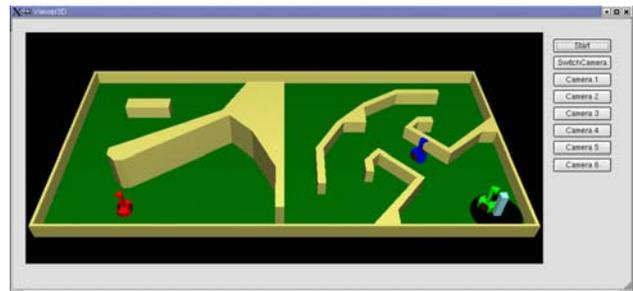


Figura 7: Visualizador 3D

Nesta aplicação os robots são modelados tal como aparece na figura 7, podendo o operador de visualização escolher uma de seis câmaras colocadas em posições diferentes relativamente ao labirinto e que permitem a visualização de prova a partir de perspectivas diferentes. Existem 4 câmaras fixas e 2 câmaras móveis. As câmaras fixas apresentam a visualização da prova a partir dos 4 lados da área de jogo. As câmaras móveis seguem um dos robots ao longo da execução da sua prova.

O visualizador 3D foi desenvolvido em C/C++ utilizando as bibliotecas Qt para a definição da sua interface e a biblioteca OpenGL para a visualização tridimensional da prova.

7 OS AGENTES ROBÓTICOS

A responsabilidade pelo desenvolvimento dos agentes robóticos é obviamente das equipas concorrentes. A arquitectura aberta do sistema de simulação e do protocolo de comunicações XML tem como objectivo estimular a criatividade e a autonomia das equipas concorrentes. No entanto e no sentido de facilitar a vida aos concorrentes a organização disponibiliza algumas ferramentas que são úteis para o desenvolvimento dos agentes.

7.1 Bibliotecas de comunicação

São fornecidas pela organização às equipas concorrentes bibliotecas de comunicação que permitem a programação do agente robótico em C/C++, quer em Linux quer em Windows, de forma transparente quanto aos mecanismos de comunicação. Através da utilização dessas bibliotecas de comunicação o registo do robot, a recepção dos dados sensoriais ou o comando dos motores são realizados simplesmente através de chamadas a funções.

Algumas das bibliotecas de comunicações fornecidas permitem a utilização pelo robot de uma interface gráfica utilizando as bibliotecas Qt.

A organização fornece ainda uma biblioteca que permite a programação do robot em Prolog. A biblioteca é lida pelo swi-prolog [Swi04] em Linux e a recepção e envio de mensagens passa a processar-se através da execução de predicados.

Está em desenvolvimento uma biblioteca de comunicações em Java.

Desenvolvidas pelas equipas concorrentes, em alguns casos com o apoio técnico da organização do Ciber-Rato

existem bibliotecas de comunicação em Visual Basic e Delphi.

7.2 O agente exemplo

O agente exemplo fornecido em conjunto com as ferramentas de apoio ao concurso pretende demonstrar a utilização da biblioteca de comunicações em C/C++ para a implementação de um robot reactivo simples.

O desenvolvimento de uma interface gráfica mínima para o agente robótico é demonstrado através de um dos agentes exemplo fornecidos.

7.3 O processo de decisão

As arquitecturas internas e os processos de decisão dos agentes concorrentes são muito variados, tal como é patente nos artigos publicados pelos concorrentes em 2002 [Ribeiro02,Reis02,Lomba02]. Existem agentes com características mais reactivas [Lomba02] ou mais deliberativas [Ribeiro02]. A utilização de um agente robótico concorrente ao Ciber-Rato como demonstrador dos princípios de funcionamento de um agente é também um dos pontos a destacar [Reis02],

7.4 Ferramentas de depuração

Algumas equipas concorrentes desenvolveram sistemas sofisticados de visualização do conhecimento do agente. A interface gráfica de um dos robots exemplo apresenta a visualização gráfica das medidas dos sensores. No entanto, esta visualização é algo básica e é frequente as equipas desenvolverem interfaces em que apresentam um mapa do mundo, dividido em zona conhecida, desconhecida e associarem às zonas conhecidas representações distintas conformem as julguem livres ou com obstáculos.

8 CONCLUSÕES

O Ciber-Rato é neste momento uma competição com um nível de notoriedade razoável. Ao longo dos seus 4 anos de existência, tem assistido ao aumento do número de equipas participantes. Todos os anos há equipas de anos anteriores que voltam e concorrer, mas existem também equipas que concorrem pela primeira vez.

No desenvolvimento de um agente robótico estão envolvidas tecnologias diversas, tais como: comunicação entre processos, inteligência artificial, construção de interfaces, desenvolvimento de ferramentas de depuração, engenharia de software, gestão de projecto, trabalho em grupo, etc. A quantidade de tecnologias envolvidas no desenvolvimento de um agente robótico, tornam o Ciber-Rato adaptável a muitos tipos de interesses.

É sabido que as boas competências de programação não se obtêm apenas dentro da sala de aula. É necessário que os alunos dediquem algum do seu tempo extracurricular a programar. O Ciber-Rato apresenta-se como um excelente veículo para potenciar esta actividade extracurricular. O aumento do número de Escolas Secundárias participantes mostra que já está a ser usado.

Desde 2003 é realizado na Ancorensis, cooperativa de ensino, em Vila Praia de Âncora, um concurso Ciber-Rato destinado a Escolas Secundárias que utiliza as ferramentas do Ciber-Rato da Universidade de Aveiro. Neste concurso a tarefa dos concorrentes é facilitada pois o sensor de GPS encontra-se activo durante a prova e o desafio consiste apenas em deslocar-se do ponto de partida até à área-alvo no menor tempo possível.

As ferramentas do Ciber-Rato têm também sido usadas na Universidade de Aveiro, Faculdade de Engenharia da Universidade do Porto, Universidade Fernando Pessoa e Escola Superior de Tecnologia de Castelo Branco no âmbito de trabalhos ou de docência de aulas práticas de cadeiras de Programação, Inteligência Artificial e Robótica.

9 AGRADECIMENTOS

Gostaríamos de agradecer em primeiro lugar a todos os que participaram como concorrentes, júris, ou simplesmente como assistentes nas edições anteriores do Ciber-Rato. Uma palavra especial de agradecimento para os outros elementos da Organização do Concurso Micro-Rato que contribuem decisivamente para o sucesso de ambas as modalidades. A equipa Ciber quer ainda agradecer à Reitoria e ao Departamento de Electrónica e Telecomunicações da Universidade de Aveiro pelo apoio prestado.

10 REFERÊNCIAS

- [Almeida00] Luís Almeida, Pedro Fonseca, José Luís Azevedo, Bernardo Cunha. The *Micro-Rato* Contest: Mobile Robotics for All. *Proc. Of CONTROLO 2000, The Portuguese Control Conference*, Guimarães (2000)
- [Bomb04] Concurso Robô Bombeiro.
<http://www.estg.ipg.pt/robobombeiro/>
- [Carter04] Everett Carter. Generating Gaussian Numbers.
<http://www.taygeta.com/random/gaussian.html>
- [Ciber04] Regras e especificações Técnicas da Modalidade Ciber. (2004)
<http://microrato.ua.pt/main/Historia/Docs/RegrasCiberRato2004.pdf>
- [Cherkassky94] B. V. Cherkassky, A. V. Goldberg, and T. Radzik. Shortest Paths Algorithms: Theory and Experimental Evaluation. *Proc. 5th ACM-SIAM Symposium on Discrete Algorithms*, (1994) 516-525
- [Lomba02] E. Lomba. R_ZERO: Um Agente Virtual “Reactivo Ma Non Troppo”. *Electrónica e Telecomunicações*, 3(7) (2002) 666-669
- [MicroRato04] Página oficial do Micro-Rato. (2004)
<http://microrato.ua.pt>
- [Murphy01] Robin Murphy. “Competing” for a Robotics Education. *IEEE robotics & Automation Magazine*, 8(2), (2002) 44-55.
- [Neves02] António Neves e João Figueiredo. Ferramentas de Apoio ao Concurso Ciber-Rato. Relatório da Cadeira de Projecto, Universidade de Aveiro (2002)
- [Noda98] I. Noda, H. Matsubara, K. Hiraki e I. Frank, SoccerServer: A Tool for Research in Multiagent Systems, *Applied Artificial Intelligence*, 12(3) (1998) 233-250

- [RoboCup04] RoboCup Federation. RoboCup official site.
<http://www.robocup.org>
- [Reis01] Luis Paulo Reis e Nuno Lau, FC Portugal Team
Description: RoboCup 2000 Simulation League Champion.
RoboCup-2000: Robot Soccer World Cup IV, LNAI
2019, Springer Verlag, Berlin (2001) 29-40
- [Reis02] Luís Paulo Reis. Ciber-FEUP – Um Agente para
Utilizar o Simulador Ciber-Rato no Ensino da Inteligência
Artificial e Robótica Inteligente. *Electrónica e
Telecomunicações*, 3(7) (2002) 655-658
- [Ribeiro02] Pedro Ribeiro. YAM (Yet Another Mouse) – Um
Robot Virtual com Planeamento de Caminho a Longo
Prazo. *Electrónica e Telecomunicações*, 3(7) (2002) 672-
674
- [Riley03] Patrick Riley e George Riley. SPADES --- A
Distributed Agent Simulation Environment with Software-
in-the-Loop Execution. *Winter Simulation Conference
Proceedings*, (2003) 817–825
- [Sklar02] Elizabeth Sklar and Simon Parsons.
RoboCupJunior: a vehicle for enhancing technical
literacy. *Proceedings of the AAAI-02 Mobile Robot
Workshop* (2002)
- [Stone03] P. Stone. Multiagent Competitions and Research :
Lessons from RoboCup and TAC. *RoboCup-2002: Robot
Soccer World Cup VI*, Springer Verlag, Berlin, (2003)
224–237
- [Smith03] Russel Smith. Open Dynamics Engine v0.039 User
Guide. (2003) [http://ode.org/ode-0.039-
userguide.html](http://ode.org/ode-0.039-userguide.html)
- [Swi04] SWI-Prolog's Home.
<http://www.swi-prolog.org/>
- [TrollTech04] TrollTech. <http://www.trolltech.com>