

# Simulação Multiagente Interativa no Ambiente SIMULA

Marcos Paulo Martins da Silva<sup>1</sup>, Daniela Bagatini<sup>1</sup>, Rejane Frozza<sup>1</sup>

<sup>1</sup>UNISC - Universidade de Santa Cruz do Sul  
Departamento de Informática  
Av. Independência, 2293 – Santa Cruz do Sul - RS

marcospaulomartinsdasilva@yahoo.com.br; {bagatini,frozza}@unisc.br

**Abstract.** *To facilitate the development of multiagent systems, was implemented an environment to develop reactive agents, called SIMULA. This environment aims reduce the programming effort in creating user applications using a graphical interface that allows specify the agents and how they will act in resolution of a problem. This paper presents aspects of interactivity in the SIMULA environment, allowing users to analyze different situations in the simulation process.*

**Resumo.** *Para facilitar o desenvolvimento de sistemas multiagentes, foi concebido um ambiente de desenvolvimento de agentes reativos com propósito didático chamado SIMULA. Este ambiente tem como objetivo principal diminuir o esforço de programação do usuário na criação de suas aplicações, por meio de uma interface gráfica que permite especificar os agentes e como eles agirão na resolução de um determinado problema. O objetivo principal deste trabalho foi desenvolver aspectos de interatividade no ambiente SIMULA, permitindo aos usuários analisarem diferentes situações no processo de simulação da modelagem realizada.*

## 1. Introdução

O avanço da tecnologia de construção de agentes inteligentes tornou ainda mais atraente a área de simulação. É possível simular situações como: um jogo de futebol, um robô coletor de materiais em Marte, tráfego aéreo e urbano, catástrofes naturais, situações ambientais diversas, entre outras, ou seja, qualquer problema real, simples ou complexo, com o intuito de melhorar o processo decisório em relação aos resultados atingidos (Rezende, 2003).

Como existem problemas complexos que estão além das capacidades individuais dos agentes, foram concebidos os Sistemas Multiagentes. Segundo (Rabelo, 2009), estes sistemas consistem de uma aplicação distribuída, composta por um conjunto de agentes que cooperam entre si para a solução de um problema. Geralmente, são utilizados os agentes reativos em sistemas de simulação, por serem capazes de perceber seu ambiente e responder rapidamente às mudanças que ocorrem nele, tendo como princípio satisfazer seus objetivos (Wooldridge, 2002).

Os ambientes de desenvolvimento de sistemas multiagentes reativos permitem a simulação de eventos reais. Entre os ambientes disponíveis, destaca-se o Sistema SIMULA (Frozza, 1997), desenvolvido para fins didáticos, sendo utilizado para simulação de problemas reais, onde o usuário define a situação inicial do problema e os

possíveis comportamentos dos agentes. O ambiente executa a simulação e apresenta uma situação final atingida pela atuação dos agentes.

O objetivo principal deste trabalho foi tornar o SIMULA um ambiente de simulação multiagente que possibilita ao usuário interagir nos resultados das simulações, ou seja, permitir alterações/manipulações diretamente no cenário apresentado pelo SIMULA, o que não era possível em versão anterior do sistema. Desta forma, evitam-se problemas como o aglomerado de agentes, por exemplo, modificando o resultado da simulação. Outras alterações foram: retirar agentes, impedir agentes de passar por um determinado caminho, deixar pista em determinadas posições, entre outras ações. A intervenção do usuário durante a simulação permite uma análise mais realista da atuação dos agentes no domínio modelado no ambiente.

O SIMULA é um ambiente utilizado pelos pesquisadores da área de Sistemas Multiagentes em suas disciplinas de Inteligência Artificial e projetos, visto que é um ambiente desenvolvido para fins educacionais e que trabalha muito bem com a questão de simulação multiagente para agentes reativos. Existem outros ambientes de simulação, como o NetLogo, que é um ambiente de simulação social. Mas que exigem dos usuários a programação em alguma linguagem. No SIMULA, o usuário fica livre da programação em linguagens, ele faz todas as especificações e o próprio ambiente gera código fonte (importante contribuição do SIMULA em relação a outros ambientes). Desta forma, o SIMULA como um ambiente de simulação interativo é uma contribuição para a área, que irá propiciar novas pesquisas.

Este artigo está organizado da seguinte forma: na seção 2, apresenta-se um breve panorama sobre ambientes de simulação multiagente; na seção 3, aborda-se sobre as novas funcionalidades que possibilitam simulação multiagente interativa no ambiente SIMULA; na seção 4 relata os resultados dos testes e detalhes de implementação e, por fim, são apresentadas as considerações finais.

## **2. Simulação Multiagente**

Agentes e sistemas multiagentes estão sendo cada vez mais utilizados nas aplicações do dia a dia. Diretamente ou indiretamente, possibilitam a resolução de problemas complexos, como exemplo, no campo de simulação, com sistemas multiagentes reativos e cognitivos, na qual é possível uma simulação muito próxima da realidade, desde uma colônia de formigas até catástrofes ambientais.

Os modelos baseados em agentes consistem em interações entre os agentes e o ambiente. Mesmo em um modelo simples percebe-se um comportamento complexo que resulta destas interações com importantes informações sobre o funcionamento do sistema. Isto torna possível a exploração das conexões entre os comportamentos de indivíduos em seu micro nível e o macro nível de modelos que emergem da interação de uma sociedade destes indivíduos (Azevedo & Menezes, 2006). Entende-se por micro nível o sistema ou indivíduo isolado que possui um problema a solucionar de acordo com seu conhecimento e capacidade para resolvê-lo; e no macro nível, o sistema é considerado como uma sociedade de agentes que possuem conhecimento, capacidade de resolução limitada e interação (Werner & Demazeau, 1991).

### **2.1. Requisitos para um Ambiente de Simulação Multiagente**

Em ambientes de simulação multiagente deve ser possível descrever o sistema a ser

modelado, a simulação computacional do sistema e os recursos para a análise e monitoramento do comportamento dos agentes. E ao definir-se a sociedade de agentes, devem ser utilizados os seguintes elementos (Azevedo & Menezes, 2006):

- A definição dos agentes: as características do agente, como sua aparência e forma, tamanho, quantidade, posição (pré-fixada ou aleatória) e cor para melhorar a visualização.
- A definição do comportamento: os elementos que definem o comportamento dos agentes são: movimento (como o agente se desloca no ambiente); percepção (como o agente percebe outros agentes e objetos no ambiente); colisão (o que o agente fará quando mais de um agente quiser ocupar a mesma posição do ambiente).
- A definição do ambiente: é o local onde o agente será posicionado e onde realizará suas ações e interações.
- A identificação dos agentes e definição das variáveis: as regras de comportamento são estabelecidas na identificação do agente, sendo possível a formação de conjuntos e classes de agentes. Ao descrever o modelo são utilizadas variáveis que estabelecem parâmetros entre os agentes e o modelo, sendo classificadas em tipos como: lógicas, globais, locais e vinculadas ao agente.

Com a definição destas características dos agentes e do ambiente a serem modelados, são necessárias ferramentas (ambientes de simulação) para analisar o modelo e seus dados comportamentais, facilitando a compreensão do processo.

## 2.2. Ambientes de Simulação Multiagente

Quase todos os ambientes de simulação multiagente são compostos de um conjunto de conceitos para o projeto e descrição de uma modelagem multiagente. Alguns ambientes de simulação existentes oferecem uma padronização dos projetos de *software* e ferramentas para simulação sem limitar a complexidade ou o tipo de modelo, porém são limitados pela sua dificuldade de uso, ou seja, o usuário deve ter conhecimento de programação; e também possuem ferramentas insuficientes para a construção, execução, observação e documentação dos modelos.

Foram escolhidos três ambientes multiagentes para serem apresentados neste trabalho: SWARM (Minar et al, 1996), NetLogo (Netlogo, 2010) e SIMULA (Frozza, 1997), a fim de contribuírem para análise de características a serem abordadas na proposta de melhorias do Ambiente SIMULA, para torná-lo um ambiente interativo para desenvolvimento de aplicações com agentes reativos.

Pelos estudos realizados sobre o ambiente SWARM, foi possível verificar que apesar de ser uma adequada ferramenta para simulações multiagente, destoa dos outros ambientes por não ter uma *interface* muito amigável e, principalmente, os usuários da ferramenta devem ter um bom conhecimento em linguagem de programação para criar seus modelos de simulação. Já o *NetLogo* e o SIMULA têm uma *interface* amigável e fazem uso de recursos que simplificam o desenvolvimento de modelos e visualização da simulação. O *NetLogo* possui como destaque uma grande biblioteca de modelos de simulação prontos para testes nas mais diversas áreas, porém o desenvolvimento de modelos é bastante trabalhoso, pois o usuário além de desenvolver o modelo de simulação, deve montar toda a interface de visualização da simulação; e o SIMULA por ser voltado mais para uma conceitualização didática, não possui tantos recursos como o *NetLogo*, mas possui uma interface muito simples e prática, bem como recursos de

geração de código que possibilitam ao usuário rapidez e praticidade no desenvolvimento de modelos de simulação.

### **2.3. Trabalhos Relacionados**

Atualmente o campo de modelagem baseada em agentes tem sido amplamente utilizado em simulações das mais diversas áreas de domínio. A seguir serão descritos alguns trabalhos relacionados à simulação para mostrar as diferentes áreas nas quais a simulação pode ser utilizada.

O trabalho de Sistema de Simulação e Controle de Tráfego (Cancian & França, 2007) é composto de um sistema para simulação e controle de tráfego urbano, que utiliza sistemas multiagentes e sistemas especialistas organizando uma estrutura hierárquica de agentes, onde um agente principal comanda os outros agentes que são os veículos. Entre os veículos há ambulâncias que atendem a chamados de emergência. Os agentes utilizados nesta simulação basicamente são agentes reativos. Somente o agente gerente poderia ser considerado híbrido, devido a sua atuação. Os sistemas especialistas são exclusivamente utilizados na definição das regras para os agentes.

O trabalho de Sistema de Simulação da Criminalidade (Furtado, 2008) tem o objetivo de fortalecer a hipótese de que o estudo e o desenvolvimento dos sistemas multiagentes inspirados em sistemas biológicos possui uma estratégia adequada para modelar o comportamento criminal. A modelagem computacional de crimes tem o objetivo de produzir simulações para se obter uma melhor compreensão da criminalidade.

O objetivo do trabalho de Sistema de Simulação de Futebol (Bagatini, 2001) foi o desenvolvimento de um time de futebol para o simulador *Soccerserver*, ou seja, desenvolver agentes jogadores que demonstrem um nível considerável de competência na realização de tarefas, como: percepção, ação, cooperação, estratégias pré-definidas, decisão e previsão.

Através destes trabalhos foi possível analisar como foram construídas as simulações, quais foram seus objetivos, como se comportam os agentes, que benefícios trouxeram para os pesquisadores, a fim de aplicar simulações em domínios reais.

Os trabalhos relacionados foram escolhidos por utilizarem agentes reativos, que foram o foco da pesquisa.

## **3. Simulação Multiagente Interativa no Ambiente SIMULA**

Como abordado na Introdução, a área de simulação multiagente está em expansão em diversos domínios, como, por exemplo, simulação de catástrofes naturais e de situações do cotidiano de grandes cidades. E para estas simulações serem desenvolvidas, existem ambientes de simulação (apresentados na seção 2).

O SIMULA é um ambiente utilizado como plataforma na área de sistemas multiagentes em disciplinas como Inteligência Artificial, ou seja, é um ambiente desenvolvido para fins educacionais que trabalha muito bem com a questão de simulação multiagente para agentes reativos. Desta forma, um dos propósitos deste trabalho foi torná-lo um ambiente de simulação interativo, como uma nova contribuição para a área, propiciando novas pesquisas.

Devido a estes motivos e através deste trabalho, possibilitou-se a interação do usuário nos resultados das simulações realizadas pelo SIMULA, permitindo alterações/manipulações diretamente na simulação sem alterar as definições prévias do modelo como, por exemplo: retirar agentes, impedir agentes de passar por um determinado caminho, não deixar pista em determinadas posições, entre outras ações.

### 3.1. Novas Funcionalidades

A seguir, descrevem-se as novas funcionalidades que foram desenvolvidas para o SIMULA.

#### 3.1.1. Log da Simulação

Todas as simulações executadas no SIMULA são visualizadas na tela de “Execução” do ambiente, onde o usuário pode acompanhar e analisar o comportamento dos agentes no ambiente. Para que este usuário possa analisar a simulação com diferentes parâmetros e com diferentes enfoques das atuações dos agentes, foi criado um “log” que armazena dados da simulação. Desta forma, o usuário tem a possibilidade de analisar as simulações geradas e chegar a uma simulação próxima do modelo pretendido.

O arquivo de “log” do SIMULA é um arquivo de texto criado em tempo de execução, onde são gravadas informações da simulação em andamento como (Figura 1):

- Horário de início da simulação: quando o usuário clicar no botão “Iniciar” pela primeira vez, o horário do sistema será gravado no início do arquivo de “log”.
- Horário de término da simulação: quando o usuário clicar nos botões “Log” e “Sair”, a simulação será terminada e o horário do sistema será gravado no final do arquivo de “log”.
- Tempo total da simulação em ciclos: como a simulação é baseada em ciclos, é feita uma contagem do total de ciclos percorridos pela simulação e gravado no final do arquivo de “log”.
- Horário de cada ciclo: para ter-se uma ideia de quanto tempo cada ciclo leva, é gravado no arquivo de “log” o horário do sistema em que cada ciclo inicia.
- Ações e reações de cada agente em cada ciclo: são gravadas, no arquivo de “log”, todas as ações e reações que cada agente inserido na simulação executa em cada ciclo.

Com base nestas informações que constituem o arquivo de “log”, após o término de cada simulação, o usuário pode ter um histórico da simulação e arquivá-la para futuras comparações e estudos, pois através deste arquivo o usuário pode visualizar em modo texto a simulação sem ter visto a sua execução. Outra contribuição do arquivo de “log” do SIMULA é de possibilitar ao usuário medir o desempenho dos agentes em suas tarefas, sendo útil para alocar determinadas tarefas a determinados agentes e, desta forma, trabalhar este aspecto de tempo nas ações dos agentes, tornando possível mensurar a capacidade dos agentes executarem tarefas em determinado tempo.

```
Início da simulação: 15:40:13.229.

azul percebe vermelho
azul executa movimento randômico
vermelho percebe azul
vermelho executa movimento randômico
Ciclo 1: 15:40:13.432.

azul percebe vermelho
azul executa movimento randômico
vermelho percebe azul
vermelho executa movimento randômico
Ciclo 2: 15:40:13.634.

azul percebe vermelho
azul executa movimento randômico
vermelho percebe azul
vermelho executa movimento randômico
Ciclo 3: 15:40:13.837.

azul percebe vermelho
azul executa movimento randômico
vermelho percebe azul
vermelho executa movimento randômico
Ciclo 4: 15:40:14.40.

Fim da simulação: 15:40:37.768.

Tempo da simulação: 4 ciclos.
```

**Figura 1. Arquivo de Log.**

Foi criado um novo botão na tela de execução do SIMULA chamado de “Log”, onde, ao clicar, a simulação é terminada e é aberta uma caixa de “Salvar como” para que o usuário possa salvar o arquivo de “log” em qualquer pasta de seu computador e com o nome do arquivo que desejar.

### 3.1.2. Evitando o Embolo

No ambiente SIMULA, após o usuário definir os agentes, ele parte para a etapa de definição das regras de comportamento destes agentes, onde o ambiente fornece uma biblioteca de funções para a montagem e definição das ações dos agentes. Dentre estas funções, há uma denominada “movimento\_randomico”, onde o agente realiza um movimento para uma direção (verticalmente, horizontalmente ou diagonalmente), definida por uma função randômica interna. Esta função foi verificada, pois parecia não funcionar muito bem, já que muitas vezes os agentes ficavam “embolados” ou intercalando entre duas posições, por exemplo.

O que ocorria era que estava sendo utilizada uma função randômica do Java chamada “*Math.random()*” que randomizava as direções possíveis do agente se movimentar pelo ambiente. Esta função tem uma falha que faz com que ela vicie, ou seja, gere a mesma sequência de direção para os agentes por várias vezes. Para corrigir este problema foi utilizada a mesma função, porém com a condição que armazena a direção anterior e compara com a nova direção gerada. Caso esta direção seja a mesma, ela gera outra até ser diferente, fazendo com que o agente não repita os mesmos movimentos, permitindo uma melhor movimentação dos agentes no ambiente, favorecendo as suas ações na simulação.

### 3.1.3. Movendo Agentes

Quando a simulação é inicializada, é possível visualizá-la e, agora com este trabalho, interferir durante o processo da simulação. O processo de interatividade se dá na fase de

execução da simulação, onde o usuário pode mover os agentes de e para qualquer parte do ambiente, alterando a simulação. Para isto, é necessário que o usuário pare a simulação, clique com o botão direito do *mouse* sobre o agente escolhido, onde aparecerá um menu *popup* com algumas opções. Ao selecionar a opção “Mover” (Figura 2), o cursor do *mouse* mudará para o formato de uma mão e aguardará que o usuário selecione o novo local para o agente, movimentando o cursor para o local desejado no ambiente e clicando com o botão esquerdo do *mouse*, fazendo com que o agente vá para este local e o cursor do *mouse* volte para seu estado normal.

### 3.1.4. Excluindo Agentes

Além da movimentação dos agentes, também é possível a sua exclusão. Para isto, deve-se fazer o mesmo procedimento anterior, para abrir o menu *popup*, e em seguida selecionar a opção “Excluir” para excluir o agente da simulação, conforme Figura 2.

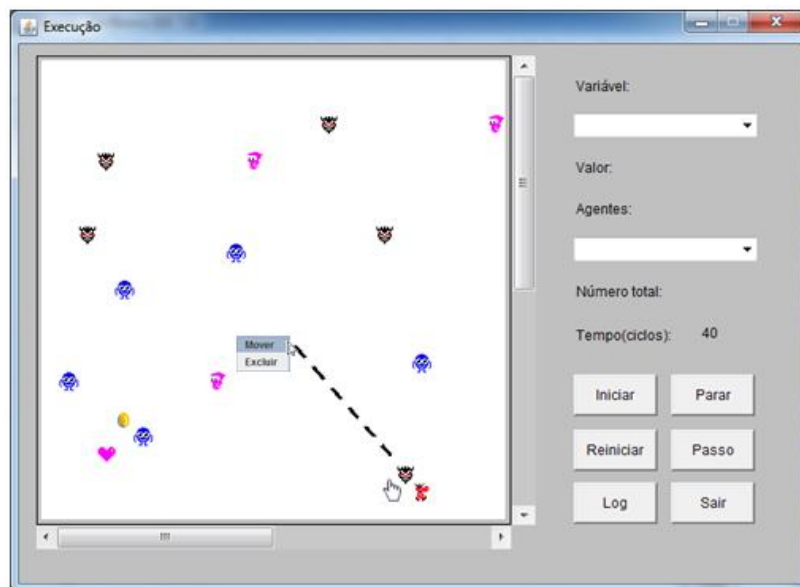


Figura 2. Interação no SIMULA.

### 3.1.5. Deixando Pistas

Outro processo de interatividade no SIMULA, baseia-se nas pistas, onde o usuário pode colocar e retirar pistas no ambiente de simulação, pistas estas necessárias em algumas simulações para que agentes as utilizem como meio de localização no ambiente. Para isto, é necessário que o usuário pare a simulação, clique com o botão direito do *mouse* sobre algum ponto vago do ambiente, ou seja, um lugar que não esteja ocupado por um agente ou obstáculo, onde aparecerá um menu *popup* com a opção “Pista”. Após esta opção ser selecionada e clicada, deixará uma pista naquele local apontado. Para a exclusão de uma pista é necessário fazer o mesmo procedimento anterior, porém deve-se clicar sobre uma pista e ao aparecer o menu *popup* selecionar e clicar na opção “Exclui”, opção esta que apagará a pista do ambiente.

### 3.1.6. Evitando o Bolo

Em algumas circunstâncias (na execução da simulação), determinados grupos de agentes aglomeram-se em volta de algo (por exemplo: um obstáculo), trancando o

sistema e causando o chamado “bolo”, “anomalia” esta, corrigida, com a intervenção do usuário, não necessitando da reinicialização da simulação para tentar contornar o problema.

Agora que o ambiente permite estas alterações/manipulações diretamente na simulação sem alterações no modelo de simulação como: “tirar um agente”; “não deixar um agente passar por um caminho”; “aqui não deixar pista”; entre outras, possibilita ao usuário fazer descobertas que não eram possíveis anteriormente, como por exemplo: retirar um agente de um determinado local e colocá-lo em outro, qual será a reação deste agente e dos outros? Com esta interatividade, o usuário terá novas possibilidades, deixando sua simulação cada vez próxima do modelo pretendido. Para a área de simulação multiagente, a característica de interatividade é o ponto principal para gerar diferentes situações e configurações dos agentes e sua atuação, a fim de aproximar a simulação da realidade modelada.

A cada início da execução da simulação, se os agentes não tiverem sua posição inicial pré-definida pelo usuário, o SIMULA randomicamente os posicionará no ambiente. Em alguns casos, alguns agentes não aparecem no ambiente, porque talvez tenham sido colocados na mesma posição de outros agentes. Esta situação ainda será verificada, a fim de não ocasionar prejuízos pela falta de agentes no processo de simulação.

#### **4. Implementação e Testes**

Como o ambiente SIMULA foi desenvolvido em Java e o trabalho em si teve foco na parte de execução do ambiente, a linguagem de programação permaneceu em Java e o IDE utilizado foi o *NetBeans 7.0*. O SIMULA é composto de vários arquivos “.java”. Os arquivos que sofreram alteração foram:

- *SIMULA.java*: Que foi alterado somente na saída do programa, ou seja, agora quando o usuário decidir sair do ambiente, abrirá uma caixa de diálogo perguntando se ele deseja salvar a simulação.
- *Metodos.java*: Possui todos os tipos de ações que os agentes podem executar na simulação. Em cada função, que agrega cada tipo de ação dos agentes, foi inserida uma chamada à função “GravaS”, que encontra-se no arquivo “Executar.java”, que extrai da função o agente, o alvo (se houver) e a ação executada pelo agente, e grava no arquivo de *log*.
- *Executar.java*: Arquivo principal da tela de execução do ambiente. Neste arquivo foram desenvolvidas todas as funções para transformar o simula em um ambiente interativo.

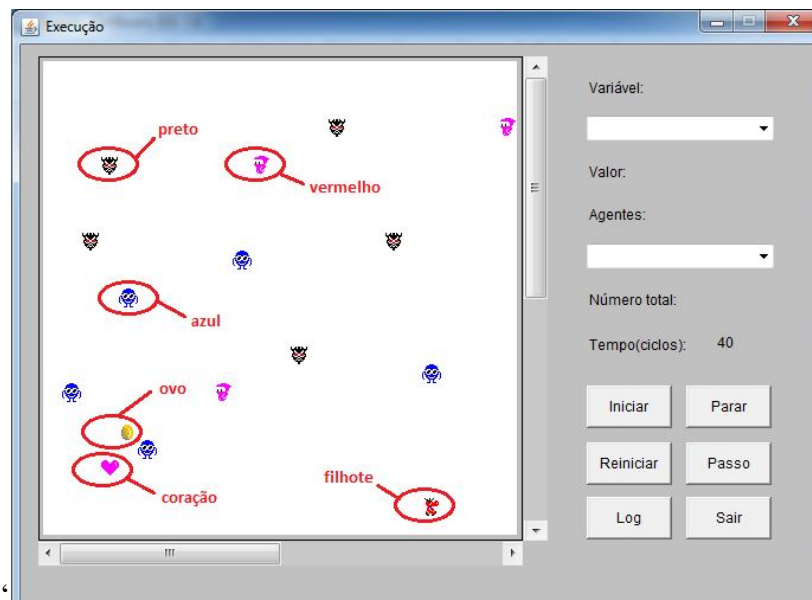
Para o arquivo de *log* foi utilizado um arquivo de texto, que é criado em tempo de execução, sofre várias alterações (escritas) e é finalizado. O arquivo é criado assim que o botão “Iniciar” é pressionado na tela de execução do ambiente e finalizado ao clicar nos botões “Log” ou “Sair”. Foi criada uma função “Hora” para buscar a hora do sistema, sendo utilizada para gravar no arquivo de *log* os horários de início e fim da simulação, bem como o tempo de cada ciclo de simulação. Já existe uma variável “tempo” que contém o número de ciclos da simulação, a qual é utilizada também para mensurar o tempo da execução em ciclos da simulação e o início de cada ciclo.

Para o usuário interagir com o ambiente, seja selecionando um agente para



movê-lo, ou uma pista a ser inserida em determinado local, foram utilizados os eventos de cliques do mouse para registrar sua posição na tela e uma função que retornava o que havia na posição clicada, para que um menu *popup* exibisse as opções possíveis ao que a função havia retornado. Para fazer a exclusão de um agente, foi utilizada a função “morteDeAgente”, que já existia no código original, porém todo o código “novo” foi desenvolvido com um prévio e minucioso estudo do código original em conjunto com as variáveis e funções já existentes no código.

Como exemplo para testes foi utilizada uma simulação constituída pelos agentes azul, vermelho, preto e filhote (Figura 3). Os agentes azul e vermelho iniciam com movimento randômico e quando percebem um ao outro, movem-se para encontrarem-se, e ao encontrarem-se formam um coração, que por sua vez transforma-se em um ovo, que em certo tempo dá origem a um filhote. Caso o filhote perceba o agente preto, foge. O agente preto por ter sua área de percepção reduzida em relação a dos outros agentes, tem o auxílio de pistas, que podem ser inseridas pelo usuário durante a simulação, pois os agentes pretos ao perceberem os filhotes, os perseguem e os matam.



**Figura 3. Apresentação dos agentes.**

Na versão anterior do SIMULA para que o usuário pudesse assistir a morte de um filhote, ele teria que aguardar até que um filhote nascesse perto de um agente preto, o que em alguns casos poderia demorar pouco ou muito tempo, dependendo do encontro dos agentes azuis e vermelhos. Porém, na nova versão interativa do SIMULA, o usuário tem opções como: colocar uma pista entre um agente preto e o filhote, mover o filhote próximo a um agente preto ou mover um agente preto próximo ao filhote. Assim, o usuário pode intervir na simulação acelerando ou retardando algum acontecimento, movendo (Figura 4) o filhote ou o agente preto, ou ainda, colocando uma pista (Figura 5) na direção do filhote, para que os agentes pretos a percebam.

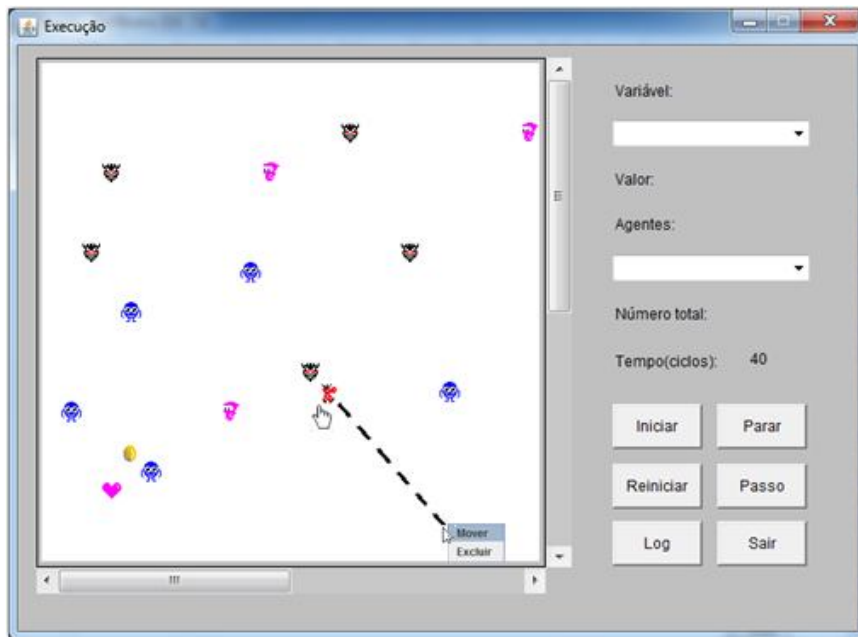


Figura 4. Movendo agente filhote.

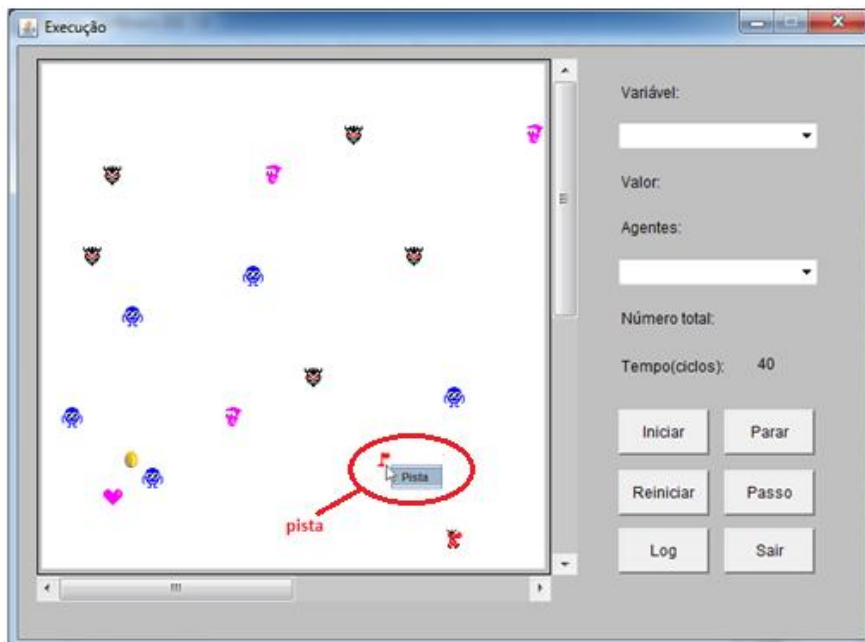


Figura 5. Inserção de pista.

A partir dos testes realizados com a simulação utilizada no exemplo apresentado, pode-se ter uma ideia de uma série de possibilidades que os usuários terão em relação à versão anterior do SIMULA, com a utilização da interatividade durante o processo de simulação. Como visualizado na simulação, além de acelerar processos da execução natural da simulação, como a morte do agente filhote, pode-se evitar vários problemas ocorridos anteriormente como o bolo e embolo de agentes e novas funcionalidades

como deixar pistas, excluir e mover agentes.

## 5. Conclusões

Com base nos estudos realizados com agentes e sistemas multiagentes, verificou-se que a área de simulação destina-se à solução de diversos problemas e, principalmente, a imitar a realidade de uma forma computacional buscando analisar situações diversas.

No trabalho, atingiram-se os objetivos da proposta destinados à interatividade, contribuindo para a área de inteligência artificial com o aprimoramento do ambiente SIMULA e principalmente na inserção da interatividade do usuário no ambiente, permitindo-lhe chegar a novas descobertas através desta ferramenta de fácil manuseio.

Em relação aos ambientes citados no artigo, o SIMULA tornou-se ainda mais distante do ambiente SWARM em termos de praticidade no desenvolvimento e execução da simulação, pois o SWARM além de não ter uma *interface* muito amigável, os usuários da ferramenta devem ter um bom conhecimento em linguagem de programação. Já em comparação com o *NetLogo*, o SIMULA por ter uma adequada *interface* e fazer uso de recursos que simplificam o desenvolvimento de modelos, bem como a visualização e interatividade do usuário no decorrer da simulação, aproxima-se um pouco do *NetLogo* em recursos, mas sempre mantendo a praticidade no desenvolvimento de modelos de simulação, por ser voltado mais para a área acadêmica.

Como trabalhos futuros, sugere-se o desenvolvimento de um “*parser*”, pois ao realizar todos os passos de configuração da simulação, o usuário deve gerar o código da aplicação antes de executá-la e há casos em que em um dos passos anteriores, pode ter havido a montagem de uma regra ou a definição de um agente de forma incorreta, causando erros na geração do código, que não são detalhados ao usuário. Portanto, é interessante o desenvolvimento de um “*parser*”, ou seja, um indicador de erros, que identifique ao usuário onde se encontra o erro, facilitando a correção de erros no processo de modelagem da simulação. Também, considerar a inserção de regras de comportamento para os agentes no momento da execução da simulação, diferentes das já definidas na modelagem.

## REFERÊNCIAS

- Azevedo, L. L.; Menezes, C. S. (2006) AProSiMA: Ambiente para Resolução Cooperativa de Problemas Baseado em Simulação Multiagente, [http://www.rc.unesp.br/serp/trabalhos\\_completos/completo2.pdf](http://www.rc.unesp.br/serp/trabalhos_completos/completo2.pdf).
- Bagatini, Daniela D. S. (2001). Um Sistema Multiagente para o Simulador Soccerserver, <http://www.lume.ufrgs.br/handle/10183/1650>.
- Cancian, M. H.; França, R. B.. (2007) Uma Abordagem de Simulação e Controle de Tráfego Utilizando Sistemas Multiagente. Florianópolis – SC, [http://www.das.ufsc.br/~gb/pg-ia/Jess07/relatorio\\_sma\\_maiara\\_ricardo.pdf](http://www.das.ufsc.br/~gb/pg-ia/Jess07/relatorio_sma_maiara_ricardo.pdf).
- Frozza, R. (1997) SIMULA: Ambiente para Desenvolvimento de Sistemas Multiagentes Reativos, [www.lume.ufrgs.br/bitstream/handle/10183/17923/000100248.pdf](http://www.lume.ufrgs.br/bitstream/handle/10183/17923/000100248.pdf).
- Furtado, V. (2008) Modelagem e Simulação Multiagente da Criminalidade, <http://funcapciencia.funcap.ce.gov.br/artigos/artigos/anexos/Sistemas%20Complexos%20e%20Simulacao.pdf>.

Minar, N.; Burkhart, R.; Langton, C.; Askenazi, M. (1996) The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations, <http://www.swarm.org/images/b/bb/MinarEtAl96.pdf>.

Netlogo. (2010) NetLogo User Manual, <http://ccl.northwestern.edu/netlogo/docs/>.

Rabelo, R. J. (2009) Projeto de Sistemas Multiagentes, <http://www.das.ufsc.br/~rabelo/Ensino/DAS6607/>

Rezende, S. O. (Org.). (2003) Sistemas inteligentes: fundamentos e aplicações. Barueri: Manole.

Werner, E.; Demazeu, Y. (1991) European Workshop on Modeling Autonomous Agents in a Multi-agent World, 3., Kaiserslautern. Proceedings, Germany.

Wooldridge, M. J. (2002) An introduction to multiagent systems. Chichester: John Wiley & Sons.