



Universidade do Porto

Faculdade de Engenharia

FEUP

AGENTES E INTELIGÊNCIA ARTIFICIAL DISTRIBUÍDA

4º ANO DO MESTRADO INTEGRADO EM ENGENHARIA
INFORMÁTICA E COMPUTAÇÃO

Simulação de Evacuação com Agentes

Relatório Intercalar

Estudantes:

Gil DOMINGUES

- up201304646@fe.up.pt

Pedro PONTES

- up201305367@fe.up.pt

6 de Novembro de 2016

Conteúdo

1	Introdução	3
2	Enunciado	3
2.1	Descrição	3
2.2	Objetivos	3
2.3	Resultados Esperados	4
3	Ferramentas	4
4	Especificação	4
4.1	Agentes	4
4.2	Interações	5
4.3	Planeamento	6
5	Conclusão	6
6	Recursos	6
6.1	Bibliografia	6
6.2	Software	6

1 Introdução

Uma evacuação implica mover pessoas de um dado local devido à ocorrência de uma situação de (potencial) catástrofe. Exemplos incluem a evacuação de um edifício em chamas ou de uma localidade, antes, durante ou após um desastre natural, como uma cheia ou terramoto.

Evacuar grandes multidões é um desafio, independentemente das circunstâncias. Tipicamente, de uma evacuação de emergência resultam feridos - ou mesmo mortes -, devido ao caos e pânico que se geram.

Com o aumento da frequência de situações que implicam a evacuação de um elevado número de pessoas num curto espaço de tempo, existe uma consciência acrescida da importância do planeamento dessas situações.

Com efeito, a gestão e organização de multidões em situações de emergência tornou-se uma importante área de estudo ao longo dos últimos anos e desempenha, hoje, um papel importante no planeamento de um edifício ou área.

Dados os desafios - quer de ordem prática, quer de ordem financeira - que a realização de simulacros coloca, é cada vez mais comum o uso de técnicas de simulação para estudar estas situações. De facto, existem já diversos tipos de sistemas, como as simulações baseadas na dinâmica de fluídos, as simulações baseadas em autómatos e as simulações baseadas em agentes.

2 Enunciado

2.1 Descrição

Ocorreu um incêndio, uma inundação, a libertação de um gás nocivo, um qualquer acidente que obriga à evacuação daqueles presentes num dado local. Esse local possui múltiplas saídas de emergência e também obstáculos. Os indivíduos encontram-se distribuídos pelo local, ocupados nas suas tarefas usuais. Aquando da deteção do acidente, todos os indivíduos procuram atingir uma das saídas de emergência, o mais rapidamente possível.

Alguns agentes poderão ser altruístas, no sentido de ajudarem acidentados a deslocarem-se até à saída, outros poderão simplesmente querer «salvar a pele», exibindo um comportamento mais egoísta, conforme se descreve adiante.

2.2 Objetivos

Realizado no âmbito da unidade curricular de Agentes e Inteligência Artificial Distribuída, com este projeto pretende desenvolver-se um programa que permita simular a interação de agentes confinados a um espaço concreto e limitado perante a necessidade de evacuar esse espaço, podendo o utilizador definir diferentes cenários, especificando, por um lado, o tipo, número e localização dos agentes a evacuar e, por outro, o número e localização de saídas de emergência e obstáculos.

2.3 Resultados Esperados

Como mencionado, será possível - e relevante - avaliar diferentes cenários, através da experimentação com:

- diferentes configurações para o local do acidente, variando o número e localização de saídas de emergência e obstáculos;
- diferentes combinações de agentes a evacuar, variando o seu tipo, número ou localização.

Deste modo, será possível observar-se como estas variações se refletem em métricas como o tempo médio e máximo de evacuação ou o número de feridos.

3 Ferramentas

A implementação do programa descrito será realizada usando *Repast*, uma *framework open-source* que permite criar, analisar e experimentar com mundos artificiais populados por agentes que interagem de forma não trivial.

Concretamente, irá utilizar-se a sua mais recente versão - *Repast Symphony* -, no *flavour RepastJ*, que permite programar em *Java* a estrutura espacial, a estrutura lógica e os comportamentos dos agentes.

Tendo sido amplamente utilizado em aplicações de simulação, considera-se de particular utilidade, por um lado, o foco em modelar o comportamento social e, por outro, a recolha de métricas associadas às simulações realizadas. Por último, tem-se a vantagem de poder acompanhar, de forma visual, o decorrer da simulação.

Adicionalmente, irá utilizar-se a *API SAJaS*, com vista a facilitar o desenvolvimento deste sistema multiagente, dado oferecer funcionalidades *JADE*. No caso, as funcionalidades de maior interesse serão as capacidades de comunicação entre agentes, visando simular as interações expectáveis num cenário de evacuação.

4 Especificação

4.1 Agentes

Podem distinguir-se dimensões distintas no comportamento exibido durante uma evacuação: por um lado, o espaço a evacuar e a sua configuração, e, por outro lado, as características psicológicas e sociais que afetam a resposta dos que participam na evacuação.

Assume-se que, em situações de emergência, os indivíduos entram em pânico e ficam, por isso, propensos a tomar decisões irracionais. Mais ainda, as pessoas tentam mover-se tão depressa quanto possível, devendo evitar obstáculos e ferimentos. Deste modo, tem-se que os agentes a implementar serão autónomos, proativos e reativos.

No caso, os modelos definidos assumem a existência de homens, mulheres e crianças, caracterizados por diversos atributos, conforme definido na Tabela 1.

Tabela 1: Atributos dos agentes a implementar.

Atributos	Tipo	Descrição
idade	int	[5, 65]
género	int	0: masculino 1: feminino
conhecimento da área	float	[0, 1] probabilidade de seguir um caminho até à saída mais próxima
independência	float	[0, 1] probabilidade de seguir (ou não) outros
altruísmo	float	[0, 1] probabilidade de ajudar outros
mobilidade	float	[0, 1] condiciona a velocidade a que se move
fadiga	float	[0, 1] >0,8: incapaz de se mover
estado de pânico	float	[0,1] condiciona a velocidade a que se move
velocidade inicial	float	condicionada pelo género e pela idade
velocidade máxima	float	condicionada pelo género e pela idade

4.2 Interações

Com vista a simular de forma mais fidedigna as condições de uma evacuação de emergência, prevê-se a implementação das seguintes interações entre agentes:

- Empurrar;

Uma pessoa pode empurrar uma pessoa no seu caminho, derrubando-a. Uma pessoa que seja derrubada verá a sua integridade física e, conseqüentemente, a sua mobilidade, reduzidas.

- Ajudar;

Uma pessoa pode ajudar outra, guiando-a até à saída, caso em que a mobilidade da pessoa altruísta fica diminuída (cansa-se mais rapidamente).

- Gritar. O facto de uma pessoa gritar pode aumentar o estado de pânico das pessoas em redor.

Além destas interações, deverá ser implementado outro tipo de interação, por modo a permitir a troca de informações entre as pessoas - nomeadamente, no que respeita à localização das saídas.

4.3 Planeamento

Para a implementação, definiram-se as seguintes etapas:

1. Especificação e planeamento;
2. Implementação de:
 - (a) Agente;
Teste e análise do comportamento de um agente num espaço.
 - (b) Espaço;
Teste e análise do comportamento de um agente num espaço.
 - (c) Interação entre agentes;
Teste e análise do comportamento de vários agentes num espaço.
3. Exploração de diferentes cenários e recolha e avaliação de métricas.

5 Conclusão

No final, consideram-se atingidos os objetivos definidos para esta primeira fase: foi feita a descrição do projeto e do seu objetivo - desenvolver um programa que permita simular a interação de agentes confinados a um espaço concreto e limitado perante a necessidade de evacuar esse espaço.

Após o estudo de diversas ferramentas, definiu-se a combinação *Repast* e *SAJaS* como a plataforma multiagente a utilizar no processo de desenvolvimento.

Caracterizaram-se os modelos de agente a implementar e definiram-se as interações entre eles.

Iniciou-se, ainda, a fase de implementação, com uma primeira definição dos agentes e espaço a evacuar.

6 Recursos

6.1 Bibliografia

[1] Almeida, João; Rosseti, Rosaldo; Coelho, António: *Crowd Simulation Modeling Applied to Emergency and Evacuation Simulations using Multi-Agent Systems*. (2011)

6.2 Software

- [1] *Plugin Repast Symphony* para *Eclipse IDE*;
- [2] *SAJaS*.