



Universidade do Porto

FEUP Faculdade de
Engenharia

MIEIC

**Agentes e Inteligência
Artificial Distribuída 2012**

T01 - Exploração de Caminhos

João Lopes | ei10009@fe.up.pt
Ricardo Pinho | ei09045@fe.up.pt

9 de Dezembro de 2012

Índice

Objectivo

Cenário

Uma companhia do exército encontra-se dispersa numa floresta, acompanhada de um grupo de robots. Existem estradas por onde todos se movem, mas algumas encontram-se danificadas e são intransitáveis (não têm saída). Há apenas uma saída da floresta. Os militares cooperam de modo a localiza-la, comunicando entre si para partilhar a informação obtida sobre as regiões já exploradas.

Objectivos do trabalho

Pretende-se criar uma aplicação que simula o cenário acima descrito, com algumas restrições, utilizando uma arquitetura de inteligência artificial distribuída com agentes. Cada agente representa uma das entidades do problema: robot, soldado ou capitão. Os agentes devem cooperar na exploração do terreno até todos os soldados e capitães terem chegado à saída da floresta.

A aplicação deverá permitir variar alguns parâmetros, nomeadamente o número de agentes de cada tipo, o mapa da floresta, a frequência de comunicação, a longevidade das baterias dos rádios e dos robots, alcance do rádio e campo de visão dos agentes. O objetivo é fazer experiências de modo a tirar conclusões sobre como a cooperação entre os agentes afecta o desempenho do sistema e de de que forma estes parâmetros alteram a eficiência da exploração dos caminhos.

Resultados esperados

Especificação

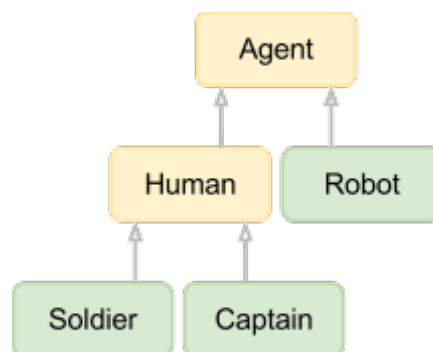


Figura 1: Hierarquia de classes de agentes

A figura 1 mostra o modelo concetual criado para representar o SMA. Os agentes são do tipo *Robot*, *Soldier* ou *Captain*, sendo que os agentes humanos partilham o seu objectivo principal - encontrar a saída dirigir-se até ela, e todos os agentes partilham um conjunto de comportamentos quanto à navegação pelos caminhos.

Caracterização de agentes

Agente

As características descritas nesta secção são comuns a todos os agentes do sistema. As secções seguintes descrevem as características específicas de cada classe de agentes.

Arquitectura

O agente tem a capacidade de se mover pelo espaço e possui, ainda, memória sobre todo o caminho percorrido previamente. A qualquer momento, o agente conhece a sua posição no mapa (GPS).

Comportamento

O agente navega pelo espaço tomando decisões quanto a mudanças de direcção quando encontra uma intersecção ou quando o caminho em frente encontra-se bloqueado (devido a uma curva ou caminho interrompido). O agente mantém não só um registo das regiões do mapa já exploradas, mas também do número de vezes que percorreu cada caminho.

Soldado

Arquitectura

O soldado é um agente deliberativo e colaborativo e tem a capacidade de comunicar a médias distancias (comunicação por rádio). Este agente tem um campo de visão que lhe permite conhecer o espaço em seu redor até uma determinada distância. O soldado pode juntar informação recolhida pelos outros agentes (recebida pelo rádio) à sua.

Comportamento

O soldado utiliza o rádio regularmente para procurar outros agentes nas imediações.

Quando a comunicação se inicia, todos os soldados no raio de alcance do rádio partilham informação sobre as regiões da floresta já conhecidas e no final todos os soldados que comunicaram terão o mesmo conhecimento sobre a exploração da floresta.

Quando o soldado toma conhecimento da localização da saída, tenta calcular a rota que lhe permite sair da floresta rapidamente tendo em conta apenas os caminhos já conhecidos. Conhecer a localização da saída não significa conhecer o caminho para lá chegar já que essa posição por ter sido comunicada a longas distâncias.

Estratégias

O objectivo do soldado é encontrar a saída e descobrir uma rota que o leve até ela. No entanto, o soldado dá prioridade à comunicação com outros agentes em relação à exploração do ambiente, uma vez que esta interacção é potencialmente mais benéfica para todos os agentes - mais informação será obtida pela partilha do que pela exploração no mesmo intervalo de tempo.

Para explorar os caminhos da floresta de forma eficiente, o soldado evita percorrer caminhos percorridos anteriormente. Para tal, regista quantas vezes percorreu um determinado caminho e, quando se depara com uma intersecção, vai escolher seguir por um dos caminhos menos visitados.

Capitão

Arquitectura

O capitão tem as mesmas características que um soldado e ainda a capacidade de comunicações a longas distâncias em todo para qualquer ponto da floresta (telemóvel).

Comportamento

O comportamento do capitão é muito semelhante ao soldado e ambos têm o mesmo objectivo. A estreia adoptada difere ligeiramente, como é explicado no ponto seguinte.

Estratégias

A estratégia do capitão é semelhante à do soldado, com duas diferenças principais.

Sempre que o capitão obtém novas informações pelo rádio comunica-as de imediato aos demais capitães. Esta accção tem, também, prioridade sobre a exploração.

Enquanto que o soldado e o robot resolvem empates (duas opções possíveis com igual utilidade) escolhendo uma das opções aleatoriamente, o capitão escolhe a opção que o afasta do capitão mais próximo. O ganho desta estratégia prende-se ao facto de, por o telemóvel permitir comunicar para qualquer ponto do espaço, a colaboração ser mais eficaz se os capitães cobrirem áreas distintas com os seus rádios.

Robot

Arquitetura

O robot é um agente reactivo. Tem mobilidade para percorrer o espaço e mantém um registo do caminho percorrido. O alcance de “visão” é mínimo tendo apenas conhecimento dos caminhos ou obstáculos imediatamente em seu redor.

Comportamento

Não tem um objetivo definido. Enquanto tem carga na bateria limita-se a mover em frente e a escolher aleatoriamente uma direcção nas intersecções.

Comunicação

Existem três tipos de comunicação válida:

- de um capitão para outro capitão, a qualquer distância;
- de um capitão ou soldado para outro capitão ou soldado que esteja dentro do raio de alcance do rádio;
- de um robot para um soldado ou capitão que esteja dentro do raio de visão (trata-se mais de uma leitura de informação do que um envio do robot para o soldado).

Todo o tipo de comunicação é restringida a uma certa frequência de utilização, visto que tanto os robots, os telemóveis e os rádios tem uma carga de bateria limitada.

Comunicação por telemóvel

1. Capitão recebe comunicação rádio com novas informações;
2. Para cada capitão ainda na floresta, o primeiro partilha a informação que recolheu mais a que acabou de receber por rádio.

Tempo total gasto em comunicação: um turno por cada troca de informação.

Comunicação por rádio

1. Agente verifica quem está no alcance do rádio;
2. Para cada agente no seu alcance, o primeiro partilha a informação que recolheu até então;
3. Prossegue com a exploração de caminhos e repete o protocolo num intervalo fixo

Desenvolvimento

O trabalho descrito neste relatório foi desenvolvido em Java, utilizando o IDE Eclipse em Windows 7. Foi utilizada a ferramenta RepastJ para criar a simulação. O Repast é uma plataforma útil para criar simulações baseadas em agentes mas não para criar sistemas de agentes reais. Fornece um grupo de ferramentas que permitem visualizar o progresso da simulação, a interacção entre os agentes e os resultados acumulados da simulação com recurso a grelhas, mapas e gráficos, além de diferentes tipos de representação dos espaços (*grid*, *torus*).

De modo a simplificar a criação de mapas, que são lidos de um ficheiro de texto, o espaço utilizado no repast tem uma estrutura em grid.

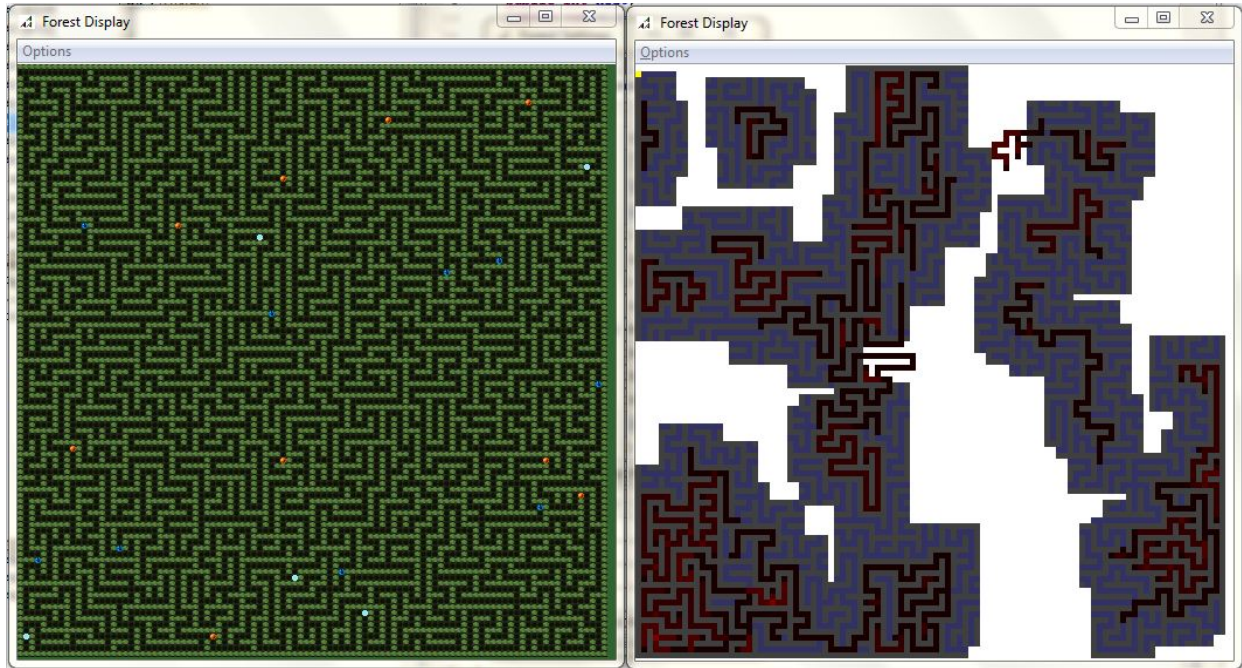


Figura 2a: Representação do mapa da floresta.

Figura 2b: Representação da informação obtida por um agente.

O Repast utiliza *displaySurfaces* para representar graficamente o espaço. Esta simulação utiliza dois destes elementos: um que representa o mapa com todos os agentes a moverem-se, os obstáculos e os caminhos (Figura 2a), e outro que permite visualizar a informação recolhida por um agente durante o seu trajecto sob a forma de um *heat map* em que zonas transitadas repetidamente aparecem mais realçadas a vermelho (Figura 2b).

A arquitectura do programa desenvolvido passa muito pela definição dos agentes, do seu comportamento e da comunicação entre eles. Existem outras classes, como mostrado na Figura 3, que cumprem funções relevantes no funcionamento do sistema, tais como as descritas em seguida.

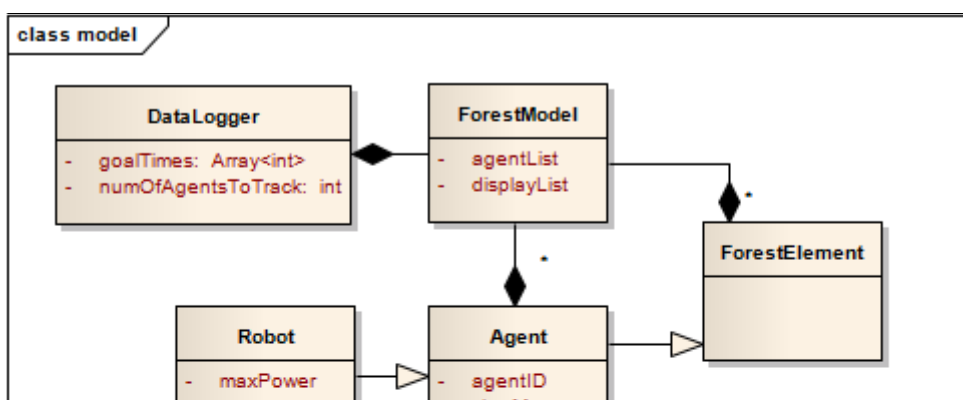


Figura 3: Modelo Concetual

ForestModel (extends SimModelImpl)

Classe central do sistema multi agente, responsável pela apresentação da interface gráfica do Repast, pelo escalonamento das acções dos agentes e pela gestão de dados e parâmetros.

DataLogger

Os métodos desta classe agregam dados úteis à análise do desempenho do sistema. Esta classe foi criada para criar registos em ficheiros de texto do tempo médio de chegada à saída da floresta e de todos os parâmetros associados à simulação efetuada.

ForestElement

Todos os elementos contidos no espaço do Repast são instancias desta classe. Não só agentes, como também os obstáculos (árvores).

Experiências

O objectivo dos testes é verificar qual o efeito da manipulação dos parâmetros da simulação no desempenho geral da equipa. Este desempenho é avaliado pelo tempo que os agentes levam a terminar o seu trabalho. Neste caso em concreto, quando mais rápido (menos turnos) todos os soldados e capitães conseguirem encontrar e chegar à saída, melhor foi o desempenho do sistema. O número de iterações usado em cada um dos testes foi de 10.

Experiência 1: Variação do alcance do rádio

- Objectivo
 - Controlar a performance com o alcance do rádio
- Resultados
 - Média de turnos com alcance de 40: 95.2
 - Média de turnos com alcance de 1:376.2
- Parametros
 - RadioCommInterval: 40
 - MapFile: 2
 - NumCaptains: 20
 - NumRobots: 100
 - RobotMaxPower: 350
 - RadioMaxPower: 400
 - VisionRange: 40
 - EfficientHuman: true
 - EfficientComm: true
 - UseIcons: false
 - NumSoldiers: 50
 - CellMaxPower: 150
 - DebugMode: false
 - WithGraphics: true

Experiência 2: Variação da frequência de utilização do rádio

- Objectivo
 - Controlar a performance com a frequência do rádio
- Resultados
 - Média de turnos com frequência de 40: 30.4
 - Média de turnos com frequência de 1:99.7
- Parametros
 - MapFile: 1
 - NumCaptains: 20
 - RadioMaxRange: 100
 - NumRobots: 100
 - RobotMaxPower: 350

- RadioMaxPower: 400
- VisionRange: 1
- EfficientHuman: true
- EfficientComm: true
- UseIcons: false
- NumSoldiers: 50
- CellMaxPower: 150
- DebugMode: false
- WithGraphics: true

Experiência 3: Variação da longevidade dos robots

- Objectivo
 - Controlar a performance com a longevidade dos robots
- Resultados
 - Média de turnos com bateria de 350: 94.1
 - Média de turnos com bateria de 1: 98.8
- Parâmetros
 - RadioCommInterval: 40
 - MapFile: 2
 - NumCaptains: 20
 - RadioMaxRange: 40
 - NumRobots: 100
 - RobotMaxPower: 1
 - RadioMaxPower: 400
 - VisionRange: 40
 - EfficientHuman: true
 - EfficientComm: true
 - UseIcons: false
 - NumSoldiers: 50
 - CellMaxPower: 150
 - DebugMode: false
 - WithGraphics: true

Experiência 4: Variação da longevidade do rádio

- Objectivo
 - Controlar a performance com o longevidade do rádio
- Resultados
 - Média de turnos com longevidade de 400: 96.8
 - Média de turnos com longevidade de 1: 98.8
- Parâmetros
 - RadioCommInterval: 40
 - MapFile: 2

- NumCaptains: 20
- RadioMaxRange: 40
- NumRobots: 100
- RobotMaxPower: 350
- VisionRange: 40
- EfficientHuman: true
- EfficientComm: true
- Uselcons: false
- NumSoldiers: 50
- CellMaxPower: 150
- DebugMode: false
- WithGraphics: true

Experiência 5: Variação do número de capitães

- Objectivo
 - Avaliar o efeito dos capitães no desempenho do sistema
- Resultados
 - Média de turnos com 20 capitães: 188.8
 - Média de turnos com 1 capitão: 95.8
- Parâmetros
 - RadioCommInterval: 40
 - MapFile: 2
 - RadioMaxRange: 40
 - NumRobots: 100
 - RobotMaxPower: 350
 - RadioMaxPower: 1
 - VisionRange: 20
 - EfficientHuman: true
 - EfficientComm: true
 - Uselcons: false
 - NumSoldiers: 50
 - CellMaxPower: 150
 - DebugMode: false
 - WithGraphics: true

Experiência 6: Variação do número de robots

- Objectivo
 - Avaliar o efeito dos robots no desempenho do sistema
- Resultados
 - Média de turnos com alcance de 100: 115
 - Média de turnos com alcance de 0: 118.4
- Parâmetros
 - RadioCommInterval: 40

- MapFile: 2
- NumCaptains: 1
- RadioMaxRange: 40
- RobotMaxPower: 350
- RadioMaxPower: 500
- VisionRange: 10
- EfficientHuman: true
- EfficientComm: true
- UseIcons: false
- NumSoldiers: 50
- CellMaxPower: 550
- DebugMode: false
- WithGraphics: true

Experiência 7: Variação da duração do telemovel

- Objectivo
 - Avaliar o efeito dos soldados no desempenho do sistema
- Resultados
 - Média de turnos com duração de 1000: 122.1
 - Média de turnos com duração de 0: 105.7
- Parâmetros
 - RadioCommInterval: 40
 - MapFile: 2
 - NumCaptains: 100
 - RadioMaxRange: 20
 - NumRobots: 0
 - RobotMaxPower: 350
 - RadioMaxPower: 100
 - VisionRange: 5
 - EfficientHuman: true
 - EfficientComm: true
 - UseIcons: false
 - NumSoldiers: 50
 - DebugMode: false
 - WithGraphics: true

Conclusões

Em relação aos resultados das experiências, ao variar os parâmetros usados, conseguimos verificar com exactidão o que cada parâmetro contribui para o SMA.

Entre os resultados, é importante salientar a diferença dos resultados esperados e obtidos, nomeadamente na eficácia dos agentes robots. A informação que os robots partilham com os militares não é tão importante para o desempenho do sistema como o esperado, o que é visível nos testes da longevidade e do número de robots.

Por outro lado, em relação aos outros testes realizados, os resultados não são muito diferentes dos esperados, sendo que a partilha de informação toma um lugar muito importante no sistema. A experiência 5, em quem foi variado o número de capitães mostra que menos capitães é melhor para o desempenho geral do sistema. No entanto, isto deve-se ao facto de o capitão perder mais tempo a comunicar (por rádio e por telemóvel) além de que o capitão tenta, quando possível, afastar-se dos demais capitães para permitir uma melhor sobertura de toda a floresta. Num mapa em que existem poucos caminhos possíveis até a saída esta estratégia provou-se pouco útil visto que alguns capitães ficavam perdidos no labirinto, muito distantes da saída, aumentando consideravelmente o tempo que levariam a chegar até ela.

No cenário proposto, o uso de SMA permite visualizar a diferença de cada agente, actuando de acordo com os objectivos definidos para o agente, bem como a sua coordenação. Através de limitações em relação ao cenário proposto, é possível realizar previsões para cenários reais, verificando possivelmente quais os cenários mais benéficos.

Melhoramentos

- Melhorar método de exploração de novos caminhos pelos agentes. O método implementado utiliza apenas informação da vizinhança para tomar decisões sobre qual a direcção a seguir. Se o agente perder algum tempo a calcular o caminho mais curto até à próxima área inexplorada, o ganho pode ser maior que o tempo gasto a executar o algoritmo.
- Refinação do protocolo da comunicação, tornando-a mais eficaz (perder menos turnos executar mais rapidamente);
- Funcionamento com mapas com mais do que uma saída;
- Melhoramento da escalabilidade do sistema;

Recursos

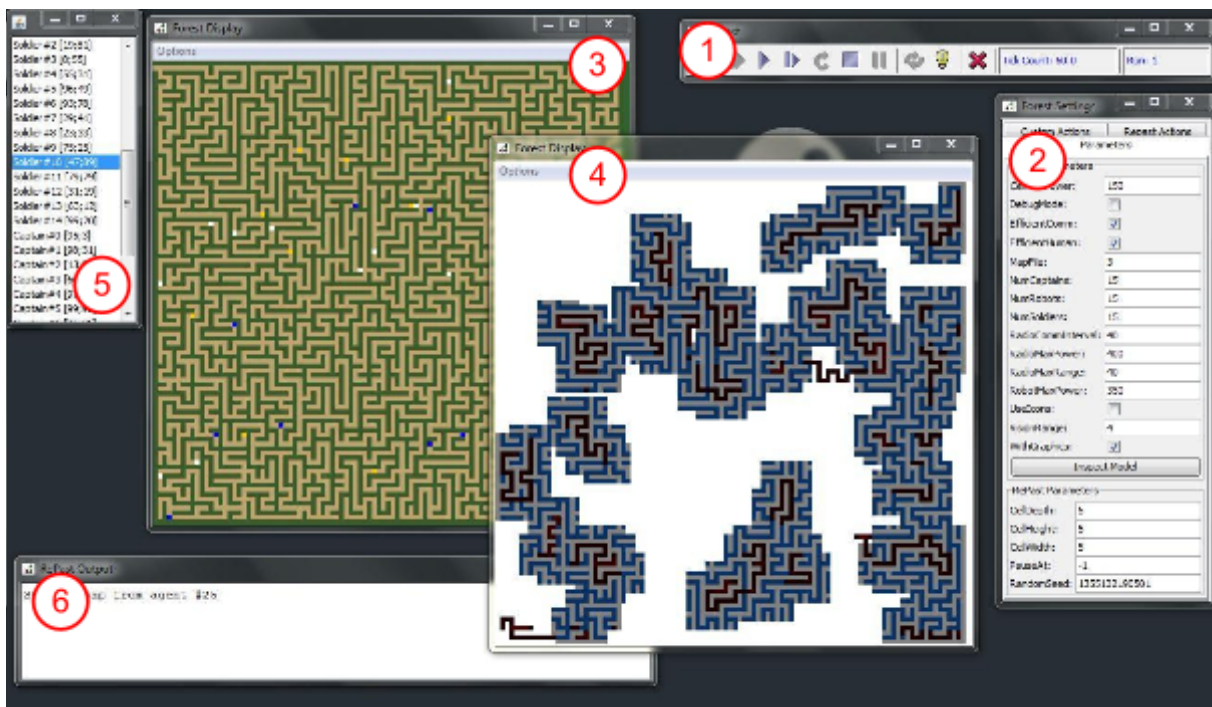
- Apontamentos da unidade curricular
- Software
 - Eclipse Juno
 - Repast J
- Elementos do grupo: Divisão uniforme do trabalho; 50% para cada elemento do grupo.

Apêndice

Definição de parâmetros

Parâmetro	Valor	Descrição
RadioCommInterval	int	Intervalo entre sucessivas comunicações via rádio
MapFile	int	Identificador do ficheiro de mapa. Por exemplo, MapFile=2 refere-se ao ficheiro map2.txt
NumCaptains		
NumRobots	int	Número de agentes de cada tipo
NumSoldiers		
RadioMaxPower	int	Número de utilizações possíveis do rádio até que descarregue a bateria.
VisionRange	int	Distancia a que os agentes humanos conseguem ver e obter informação.
EfficientHuman	bool	Se activado, os soldados e capitães utilizam o método mais eficiente de exploração, evitando percorrer os caminhos já conhecidos. Caso contrário, escolhe sempre um caminho aleatório nas intersecções.
EfficientComm	bool	Se activado, acelera a comunicação fazendo com que um soldados não perca um turno se verificar que não existe ninguém com quem iniciar comunicação por rádio.
CellMaxPower	bool	Número de utilizações possíveis do telemóvel até que descarregue a bateria.
RobotMaxPower	int	Número máximo de turnos que o robot aguenta com a carga da sua bateria.
UseIcons	int	Activa ou desactiva os icons da interface gráfica.
DebugMode	bool	Corre a simulação em modo verboso.
WithGraphics	bool	Activa ou desactiva a interface gráfica

Interface gráfica do RepastJ



1. Janela de controlo da simulação
2. Controlo dos parâmetros descritos na secção anterior
3. Representação gráfica da simulação. Os espaços verdes são árvores. Os ícones azuis são os capitães, os laranja são soldados e os ícones brancos são robots.
4. *Heat map* que representa o conhecimento de um agente quanto à floresta.
5. Janela de seleção do agente cujo conhecimento surge na janela 4.
6. Consola do Repast onde surgem algumas informações e mensagens de debug.

Após iniciar a aplicação, o utilizador deve correr a simulação utilizando os controlos do Repast. Surge a representação gráfica da simulação. A simulação é concluída quando todos os soldados e capitães tiverem chegado à saída. No final de cada simulação é guardada uma nota no ficheiro log.txt sobre o tempo médio que os agentes levaram a chegar à saída, assim como os parametros utilizados.