



**FEUP** **FACULDADE DE ENGENHARIA**  
**UNIVERSIDADE DO PORTO**

## **Exploração do Espaço Desconhecido**

Relatório Intercalar

Agentes e Inteligência Artificial Distribuída

4º Ano do Mestrado Integrado em Engenharia Informática e Computação

Elementos do grupo:

- Filipe Coelho - up201500072 - up201500072@fe.up.pt
- Hugo Cunha - up201404587 - up201404587@fe.up.pt
- José Vieira - up201404446 - up201404446@fe.up.pt

Porto, 5 de Novembro de 2017

# Índice

<b>Enunciado</b>	<b>2</b>
Descrição do cenário	2
Objetivo do trabalho	2
Resultados esperados e forma de avaliação	2
<b>Plataforma / Ferramenta</b>	<b>4</b>
Utilidades e características	4
JADE	4
Repast Simphony	4
SAJaS	4
Funcionalidades relevantes para o trabalho	4
<b>Especificação</b>	<b>5</b>
Identificação e caracterização dos agentes	5
Protocolos de interacção	5
Faseamento do projecto	6
<b>Recursos</b>	<b>7</b>
Bibliografia	7
Software	7

# Enunciado

## Descrição do cenário

No âmbito da unidade curricular de Agentes e Inteligência Artificial Distribuída, o grupo irá desenvolver um projeto no qual um certo número de agentes explora um espaço desconhecido, com obstáculos (que podem ser amovíveis) ou não, cujo objetivo final é que todos atinjam um ponto no espaço que é a saída.

Além de técnicas/algoritmos de procura de espaço, os quais serão utilizados pelos agentes, cada agente terá um raio de ação, no qual poderá comunicar com outros com o objetivo de facilitar a chegada ao destino final. Irá haver, ainda, um grupo de agentes denominados “super agentes”, cujo raio de ação de comunicação é infinito, possibilitando a comunicação com os outros super agentes, independentemente da sua posição.

Os agentes também possuem um raio de visão, o que condiciona as suas acções no espaço.

## Objetivo do trabalho

Temos como objetivo desenvolver uma aplicação com agentes de inteligência artificial que procuram, em conjunto e com partilha de informação entre os mesmos, uma saída de um espaço fechado. Assim, esperamos desenvolver as nossas apetências no que concerne sistemas multiagente e as suas aplicações a determinadas situações ou cenários.

## Resultados esperados e forma de avaliação

A nossa aplicação irá permitir a introdução de alguns parâmetros variáveis. Assim, ao iniciar a simulação, será possível indicar o número de agentes a utilizar, o número de super agentes, a distância máxima de comunicação dos agentes, tamanho do mapa, número variável de saídas disponíveis e ainda número de obstáculos que podem, ou não, necessitar da cooperação de diversos agentes para os ultrapassar.

Iremos avaliar a qualidade da nossa aplicação tendo em conta o número de agentes utilizados, o tamanho do espaço, distância de comunicação dos agentes, número de obstáculos presentes no espaço e o tempo que os agentes demoram a encontrar a saída.

# Plataforma / Ferramenta

## Utilidades e características

### JADE

Esta é uma ferramenta *open-source*, programada em Java, que permite o desenvolvimento de sistemas multi-agente *peer-to-peer*. O JADE está desenvolvido segundo as especificações FIPA.

Permite criar um sistema distribuído de agentes, podendo estes agir dentro deste domínio. Cada agente tem associado uma função *behaviour*, ou seja, tarefas a serem executadas por um agente.

### *Repast Symphony*

O Repast tem como objetivo realizar simulações usando agentes. O Repast não permite o desenvolvimento de sistemas multi-agente como o Jade, apenas a recolha de dados e apresentação dos mesmos ao utilizador.

### SAJaS

O SAJaS serve como “ponte” de ligação entre Repast e Jade, já que estes não comunicam entre si nativamente. O SAJaS permite o escalonamento dos agentes no Repast.

## Funcionalidades relevantes para o trabalho

O uso de Jade vai-nos permitir desenvolver uma plataforma multi-agente com foque para a utilização das abstrações agente e, dentro deste, a função *behaviour*, assim como o mecanismo de comunicação assíncrona (*peer-to-peer*) entre os mesmos.

No que concerne Repast+SAJaS, serão também importantes na medida em que SAJaS permite fazer de “ponte de ligação” entre Repast e Jade. Repast, como é dotado de mecanismos de visualização de agentes e as suas interações em ambiente gráfico, iremos utilizá-lo para mais facilmente avaliar resultados obtidos.

# Especificação

## Identificação e caracterização dos agentes

Tal como referido em cima, iremos utilizar dois tipos de agentes. O agente comum e um “super agente”. A diferença entre ambos está no raio de ação no que concerne a comunicação.

Vamos considerar que estes terão tarefas um pouco distintas. Assim, os super agentes, no início do programa, irão executar o algoritmo de Pledge, de forma a descobrir os limites do nosso espaço. Os agentes comuns irão executar o algoritmo de *Depth-first Search*, tentando, desta forma, otimizar a descoberta do espaço desconhecido. Quando os agentes descobrirem a saída (partilha de informação) irão executar o algoritmo  $A^*$  para descobrir o caminho mais curto para a mesma.

De acordo com a caracterização do tipo de agentes, classificamos os nossos em *Model-Based* e *Simple-Reflex*. Assim sendo, os nossos agentes são *Model-Based* quando vagueiam e executam os algoritmos de pesquisa, ou quando já sabem onde se encontra a saída, mas também querem “ajudar” o maior número de agentes a atingir a mesma, mantendo sempre um estado interno do mundo. São *Simple-Reflex* quando se aproximam de outro agente, estabelecem comunicação, e transmitem conhecimento um ao outro sobre o mapa.

## Protocolos de interação

A cada execução do *behaviour* de cada agente, o agente verifica os agentes que estão dentro do seu raio de ação e envia a sua matriz de descoberta. Assim que um agente recebe esta matriz, faz uma fusão entre a sua atual matriz e a que acabou de receber, de forma a aumentar o conhecimento sobre o espaço.

Da mesma forma os super agentes enviam aos outros super agentes, a cada execução do *behaviour*, os novos dados que possam ter surgido.

```

while (numAgentsAroundMe > 0)
{
    SendMyMatrix(agents[i]);
    Matrix m = ReceiveAgentMatrix(agents[i]);
    updateMyMatrix(m);
    i++;
}

```

Quando algum agente descobre a saída, vai à procura de outros agentes para fornecer a sua matriz com a solução. Os agentes que forem encontrados (denominados seguidores) seguem o agente que lhes indicou a saída (denominado mestre). Passado algum tempo, o agente mestre volta à saída, juntamente com todos os agentes seguidores. O agente mestre permanece na saída (passando a denominar-se agente orientador), enquanto que os seguidores voltam a vaguear para tentar encontrar mais agentes que ainda não saibam a localização da saída (passando a denominar-se agentes mestre).

O processo é repetido, e eventualmente voltam a dirigirem-se à saída. Caso algum dos agentes tenha “recrutado” novos agentes seguidores, informam o agente orientador. Em seguida perguntam se houveram novos agentes recrutados desde a última vez que o agente perguntou, e se existirem o agente volta a vaguear, caso contrário usa a saída e abandona o espaço.

## Faseamento do projecto

1. Implementar os dois primeiros algoritmos a utilizar: *Pledge* e *Depth-first Search*;
2. Criar um mapa com alguns obstáculos;
3. Testar o comportamento de apenas um agente no mundo, e caso surjam problemas, iterar até que o comportamento seja o esperado;
4. Testar o comportamento de múltiplos agentes no mundo, e caso surjam problemas, iterar até que o comportamento seja o esperado;
5. Gerar aleatoriamente uma saída para o nosso espaço;
6. Implementar o algoritmo A\*;
7. Implementar as restantes técnicas de cooperação entre agentes.

# Recursos

## Bibliografia

- Página Web da disciplina -  
[https://paginas.fe.up.pt/~eol/AIAD/aiad1718\\_i.html](https://paginas.fe.up.pt/~eol/AIAD/aiad1718_i.html)
- Tutorials Point -  
[https://www.tutorialspoint.com/artificial\\_intelligence/artificial\\_intelligence\\_agents\\_and\\_environments.htm](https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_agents_and_environments.htm)
- Introduction to Intelligent Agents -  
[http://www.mind.ilstu.edu/curriculum/ants\\_nasa/intelligent\\_agents.php](http://www.mind.ilstu.edu/curriculum/ants_nasa/intelligent_agents.php)
- Agents, Artificial Intelligence -  
<https://www.doc.ic.ac.uk/project/examples/2005/163/g0516334/index.html>

## Software

O projeto será desenvolvido utilizando o Eclipse e o IntelliJ como IDE, com recurso às *frameworks* Repast, SAJaS e JADE.