



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

Enunciado	2
Descrição do cenário	2
Objetivo do trabalho	2
Resultados esperados e forma de avaliação	2
Plataforma / Ferramenta	3
Utilidades e características	3
JADE	3
Repast Simphony	3
SAJaS	3
Funcionalidades relevantes para o trabalho	3
Identificação e caracterização dos agentes	4
Protocolos de interacção	4
Faseamento do projecto	5
Recursos	6
Bibliografia	6
Software	6

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 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 “superagentes”, cujo raio de ação de comunicação é infinito, possibilitando a comunicação com os outros superagentes, independentemente da sua posição.

Objetivo do trabalho

Temos como objetivo desenvolver um programa 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

O nosso programa 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 superagentes, a distância máxima de comunicação dos agentes, tamanho do mapa, número de saídas, 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 do nosso programa tendo em conta o número de agentes utilizados, o tamanho do espaço, distância de comunicação dos agentes e número de obstáculos presentes no espaço.

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.

Este, 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

Quanto ao 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.

Identificação e caracterização dos agentes

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

Deste modo, vamos considerar que estes terão tarefas também um pouco distintas. Assim, os superagentes, 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 *goal-based*, *model-based* e também reativos. Assim sendo, os nossos agentes são *goal-based* no momento em que o seu objetivo é encontrar a saída e aplicam algoritmos para os ajudar nesse sentido. *Model-based* quando já sabem onde se encontra a saída, mas, no entanto, querem “ajudar” o maior número de agentes a atingir, também, a saída. São reativos quando se aproximam de outro agente, estabelecem comunicação, e transmitem conhecimento um ao outro sobre o mapa.

Protocolos de interacção

A cada *tick* de execução do programa, 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 superagentes enviam aos outros superagentes, a cada *tick*, os novos dados que possam ter surgido.

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

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 alguns agente enquanto executam os algoritmos propostos
4. Gerar aleatoriamente uma saída para o nosso espaço
5. Implementar o algoritmo A*
6. Implementar as restantes técnicas de cooperação entre agentes

Recursos

Bibliografia

- Página Web da disciplina -
https://paginas.fe.up.pt/~eol/AIAD/aiad1718_i.html
- Tutorials Point -
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

Software

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