Introdução à Inteligência Artificial

*Ano letivo 2019/2020 - 2º Semestre*

*Relatório Meta 2*

*Trabalho Prático Nº1:*

*Reactive D31: The AI Awakens*

Alexandre Maria Martins Magalhães Teixeira Serra

2017248031

[amserra@student.dei.uc.pt](mailto:amserra@student.dei.uc.pt)

PL6

João Gabriel de Matos Fernandes

2017247486

joaof@student.dei.uc.pt

PL2

João Pedro de Sá Dinis

2017248548

joaodinis@student.dei.uc.pt

PL6

Introdução 3

Objetivos do trabalho 3

Metodologia 4

Como fizemos? 4

Implementação 4

Testes e divisão de tarefas 5

Resultados 6

Conclusões 7

Introdução

Objetivos do trabalho

Este trabalho teve como objetivo implementar e estudar agentes reativos.

Agentes reativos são agentes que, através de estímulos e percepções do ambiente em que se encontram, executam ações com base um sistema de regras simples.

Neste trabalho, o agente reactivo é representado por um robô denominado D31, nomenclatura que usaremos para o resto do relatório, por motivos de simplificação de escrita. O D31 encontra-se num ambiente povoado por obstáculos e recursos, onde se tem de desviar e apanhar, respetivamente.

A implementação foi feita através da ferramenta Unity, popular motor de desenvolvimento de jogos altamente portáveis, e da linguagem C#. A versão de software Unity usada foi a 2019.3.1f1, com plataforma objetivo Windows.

Neste relatório iremos abordar a metodologia que usamos para resolver este trabalho, os resultados obtidos e as conclusões a que chegamos. Por fim dedicamos uma secção aos conhecimentos obtidos durante a resolução do trabalho.

Todo o código utilizado encontra-se em anexo e devidamente comentado.

Metodologia

Como fizemos?

Nesta secção iremos expor a forma como implementámos o D31, quais os procedimentos seguidos para testar e a divisão de tarefas pelos elementos do grupo.

Implementação

As partes que nos competiram implementar foram o detetor de blocos, as funções de ativação gaussiana e logarítmica negativa, os limiares de ativação mínimo e máximo e os limites inferior e superior. O código encontra-se em anexo, sendo que é de realçar o ficheiro *LinearRobotUnitBehaviour.cs*, que contém tudo exceto a implementação das fórmulas matemáticas das funções de ativação, esta última presente, de igual forma, nos ficheiros *BlockDetectorScript.cs* e *ResourceDetectorScript.cs*.

A sequência lógica de passos seguidos no ficheiro *LinearRobotUnitBehaviour.cs* foi:

1. Obter a força, no seu estado cru, através dos sensores (de recursos e blocos)
2. Ir buscar o ângulo para onde essa força deverá ser aplicada \*

3. Aplicar limiares e obter resposta da função de ativação

4. Aplicar limites

5. Modificar o peso que a força final do sensor de recursos ou blocos exercerá

Foi ainda implementado um novo mapa que permite colocar o D31 face a um ambiente mais hostil, com paredes estreitas, caminhos sem saída ou que vão dar a lava e recursos perto de paredes.

* No caso do sensor de blocos, adicionamos-lhe uma variável - *angleOffset* - que possibilita com que o D31 vá em direção oposta aos blocos.

Testes e divisão de tarefas

De forma a testar o D31 com várias configurações, dividimos as possíveis combinações de funções de ativação de blocos e recursos (linear-linear, linear-gaussiana, linear-logarítmica, gaussiana-linear, gaussiana-gaussiana, gaussiana-logarítmica, logarítmica-linear, logarítmica-gaussiana e logarítmica-logarítmica) pelos diferentes membros do grupo.

Cada membro ficou, então, encarregue de testar 3 combinações de funções de ativação para os 5 mapas\*.

Para testar, cada elemento pode manipular, para cada um dos sensores, as seguintes variáveis:

* Angle of sensors (determina o número de sensores)
* Range of sensors (alcance dos sensores)
* Weight (peso do sensor na força final)
* X thresholds (limiares de ativação mínimo e máximo)
* Y thresholds (limites superior e inferior)
* Mean (média, para a função de ativação gaussiana apenas)
* Variance (desvio padrão, para a função de ativação gaussiana apenas)

Os testes seguiram-se de uma forma semelhante a um *try-catch*. Tentávamos com uns determinados valores para as variáveis. Se não fosse bem sucedido, percebíamos qual a variável que poderíamos ajustar e assim o fazíamos. Pode parecer um método um pouco aleatório à primeira vista, porém passados inúmeros testes (ao todo foram feitos mais de 100 testes) começámos a ganhar uma “intuição” natural para perceber o problema e ajustar a variável. Muitos dos testes foram registados numa folha Excel, que se encontra em anexo, registando o tempo que o D31 demorou, se caiu ou não, se apanhou todos os blocos e, em certos casos, algumas observações.

\* Na verdade o mapa 2a foi testado por apenas um membro do grupo, já que a função de ativação para os blocos não é necessária no mapa em questão.

Resultados

Os resultados mais relevantes estão presentes no ficheiro Excel em anexo. São 77 testes documentados, sendo que esse número passa a centena se considerarmos também os testes não documentados. Cada folha do ficheiro Excel corresponde a um mapa, contendo os valores testados e, o mais importante para tirarmos conclusões, os tempos médios para cada tipo de sensor. Deixamos aqui um resumo dessas estatísticas:

|  | Mapa1a | Mapa1b | Mapa2b | Mapa3 |
| --- | --- | --- | --- | --- |
| Número de testes | 19 | 25 | 19 | 12 |
| Tempo mínimo\* | 6.0 | 9.0 | 25.0 | 41.0 |
| Tempo médio | 8.7 | 13.4 | 33.7 | 51.0 |

* sem tocar em nada

| Tempos médios para o detetor de blocos, com diferentes funções de ativação | | | | |
| --- | --- | --- | --- | --- |
| Tempo médio (s) | Mapa1a | Mapa1b | Mapa2b | Mapa3 |
| Linear | 10.3 | 11.0 | 35.0 | 56.0 |
| Gaussiana | 7.6 | 16.6 | 30.3 | 47.0 |
| Logaritmica Negativa | 8.0 | 11.0 | 35.7 | N/A |

| Tempos médios para o detetor de recursos, com diferentes funções de ativação ativação-1 | | | | |
| --- | --- | --- | --- | --- |
| Tempo médio (s) | Mapa1a | Mapa1b | Mapa2b | Mapa3 |
| Linear | 10.0 | 13.5 | 34.0 | 62.0 |
| Gaussiana | 8.0 | 14.0 | 36.0 | 46.0 |
| Logaritmica Negativa | 6.7 | 13.0 | 31.0 | 46.0 |

Conclusões