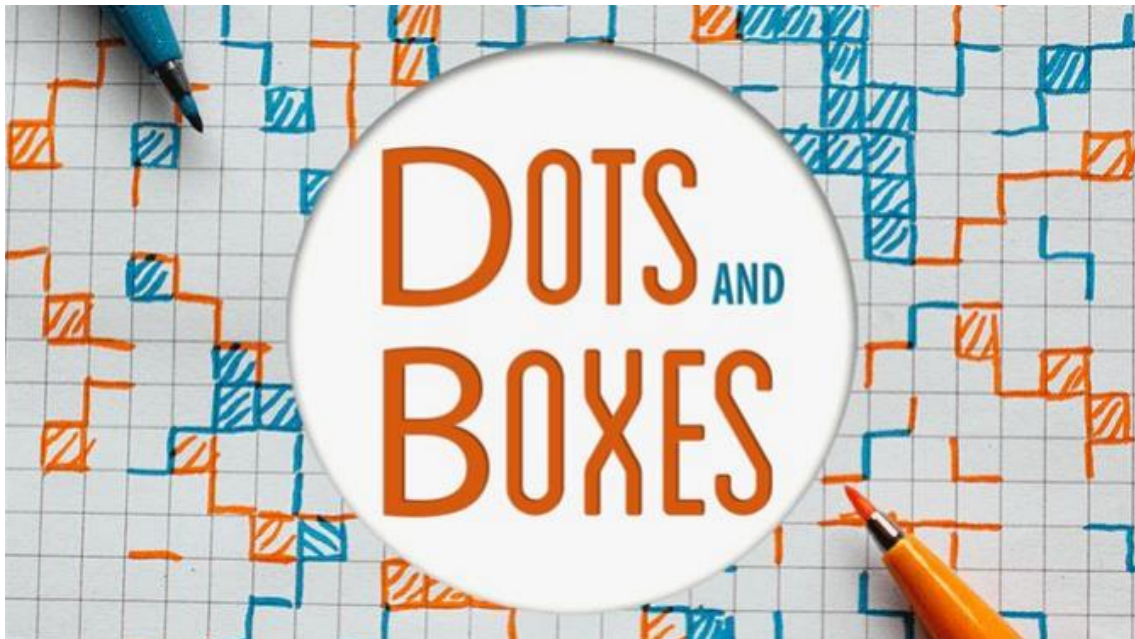


Projeto Nº 1, 1ª fase - Dots and Boxes



Manual do Utilizador

Inteligência Artificial – 2022/2023

Projeto realizado por:

- Nuno Reis Nº 202000753
- Vítor Nunes Nº 201901148

Acrónimos e convenções usadas

Para a implementação dos algoritmos foram usadas abreviações para o nome de cada algoritmo (Breadth-First foi denominado por **bfs** e Depth-First foi denominado por **dfs**) e a denominação das funções e das variáveis foi feita em português, recorrendo apenas a letras minúsculas e ao hífen (-) para separar as palavras, tendo em conta que a expressão *get* foi utilizada esporadicamente como sufixo em algumas funções que tinham como objetivo obter algo em específico:

```
(defun get-arcos-horizontais (estado)
  (first estado)
)
```

Introdução

O Dots and Boxes é um jogo de estratégia cujo objetivo é fechar um certo número de caixas num tabuleiro com diversas dimensões, de acordo com o problema pretendido. Para isso, em cada jogada pode ser colocado um arco que liga dois pontos, tendo em conta que esse arco pode ser vertical ou horizontal.

Instalação e Utilização

O primeiro passo a ter em conta é alterar as funções *get-problemas*, *get-problemas1* e *escrever-no* localizadas no ficheiro *projeto.lisp*, onde deve ser colocada a localização do ficheiro correspondente à função consoante o computador do utilizador, como por exemplo:

get-problemas - C:/Users/User/Documents/IA_Projeto/P1/problemas.dat

De seguida, os ficheiros *projeto.lisp*, *puzzle.lisp* e *procura.lisp* fornecidos devem ser compilados com o objetivo de poder utilizar o programa. Após a sua compilação, o utilizador deve introduzir o seguinte comando no *listener*: “(iniciar)”.

De entre as várias opções de problemas que são apresentadas, o utilizador escolhe o problema que ele pretende indicando a respetiva letra. Após escolher o problema, o utilizador agora deve escolher qual o algoritmo que ele pretende que resolva o problema, tendo em conta que se escolher *dfs*, o utilizador tem de inserir a profundidade máxima do algoritmo, e se escolher *a**, o utilizador tem de inserir o número de uma das duas heurísticas apresentadas.

Input/Output

O programa irá ler as funções presentes nos ficheiros *.lisp* mencionados no capítulo anterior. Ao resolver o problema, o programa irá mostrar ao utilizador o tabuleiro com o problema resolvido e as respetivas estatísticas no *listener*.

Exemplo de Aplicação

Para iniciar o programa basta inserir o seguinte comando no *listener*:

(iniciar)

Após inserir o comando, será lhe apresentado este output:

```
Níveis
Escolha uma opção
0- Terminar
Nível - A) objetivo - 3
Nível - B) objetivo - 7
Nível - C) objetivo - 10
Nível - D) objetivo - 10
Nível - E) objetivo - 20
Nível - F) objetivo - 35
|
```

Deverá inserir a letra do problema que pretende resolver. Após a resposta, será questionado sobre qual algoritmo gostaria de usar com o seguinte output:

```
Algoritmos
Escolha uma opção
0- Voltar atrás
1- Breadth-First Search
2- Depth-First Search
3- A*
|
```

Poderá agora responder com o número correspondente ao algoritmo que deseja utilizar. Caso escolha *Depth-First Search*, irá receber o seguinte output, onde deve escolher a profundidade limite do algoritmo desejada:

```
Qual a profundidade limite?
|
```

Caso escolha *A**, irá receber outro output, onde deve escolher a heurística desejada para o algoritmo tendo em conta os números apresentados no output:

```
Heurísticas
Escolha uma opção
0- Voltar atrás
1- Diferença entre o numero de caixas fechadas e o numero esperado de caixas fechadas
2- Diferença entre o numero de cantos do tabuleiro e o numero de cantos esperados(numero esperado de caixas fechadas * 4)
|
```

Após todas estas escolhas, o programa irá ser executado e em alguns segundos irá receber a solução do problema.

Por vezes, devido ao mau funcionamento do LispWorks, este pode crashar e será necessário reiniciar o IDE. Excluindo esta situação, não se espera a ocorrência de erros.