Manual de Utilizador

Inteligência Artificial - Escola Superior de Tecnologia de Setúbal

2024/2025

Prof. Joaquim Filipe

Eng. Filipe Mariano

202000634 Bruno Ascenção

202000584 Francisco Pereira

Projeto Nº 1: Época Normal

20/12/2024

Índice

1. Introdução

Este documento tem como objetivo ser o Manual de Utilizador para a utilização do programa desenvolvido.

O programa é destinado à resolução, através de diferentes algoritmos, de diversos cenários possíveis do jogo “Adji-boto*”. Todos os algoritmos aplicados foram lecionados nas aulas de Inteligência Artificial.

Neste documento vai ser abordado como instalar e utilizar o programa, para que este fique pronto a ser usado pelo utilizador.

2. Instalação e utilização

Para utilizar o programa é necessário que o utilizador instale o IDE LispWorks. O IDE LispWorks é um software de desenvolvimento de Common Lisp.

Após a instalação do IDE, o utilizador deve abrir o ficheiro "project.lisp" que é o cérebro de todo o programa. Após aberto o ficheiro, o utilizador deverá carregar o mesmo. No listener, após o ficheiro carregado, o utilizador poderá iniciar o programa escrevendo o comando "(start)".

3. Input/Output

3.1. Input

3.1.1. Ficheiros

4. Problemas.dat - Contém todos os problemas do programa.

3.1.2. Comandos

5. Menu problema - Utilizador deve escolher o problema a resolver;
6. Menu algoritmo - Utilizador deve escolher o algoritmo que será utilizado para resolver o problema
7. Heurística - O utilizador ao escolher o algoritmo "A*", aparece um menu para selecionar a heurística pretendida;
8. Profundidade máxima - O utilizador ao escolher o algoritmo "DF", aparece um menu para escolher a profunidade máxima que o algoritmo irá.

3.2. Output

9. Ficheiro estatísticas - Quando um problema termina de ser resolvido, é gerado um ficheiro na pasta "statistics". O nome do ficheiro é uma string da hora e data a que este foi gerado (hora, minuto, segundo, dia, mês, ano - xxxxxx_xxxxxx).

4. Exemplo de aplicação

Ao fazer "(start)" ao programa, será apresentado um menu inicial que pede ao utilizador para indicar qual problema quer resolver. Este deve escolher o número correspondente ao problema.

```
Available Problems
Problema 1: (0 0 0 0 0 2) (0 0 0 0 4 0)
Problema 2: (2 2 2 2 2 2) (2 2 2 2 2 2)
Problema 3: (0 3 0 3 0 3) (3 0 3 0 3 0)
Problema 4: (1 2 3 4 5 6) (6 5 4 3 2 1)
Problema 5: (2 4 6 8 10 12) (12 10 8 6 4 2)
Problema 6: (48 0 0 0 0 0) (0 0 0 0 0 48)
Problema 7: (8 8 8 8 8 8) (8 8 8 8 8 8)
Choose one (number):
```

Tendo já escolhido o problema, o utilizador passa ao menu seguinte que tem como objetivo escolher o algoritmo a utilizar na resolução do problema. Novamente, este deve escolher o respetivo número.

```
Available Algorithms
1- BF
2- A*
3- DF
Choose one (number):
```

De seguida, o algoritmo começa a trabalhar na resolução do problema. Se terminar, exibe no terminal e cria um ficheiro com estatísticas relativas à resolução. Esses dados são o número de nós expandidos e gerados, o tempo que demorou a resolver e todos os passos até à solução.

```
Problem solved!
Problem: ((0 3 0 3 0 3) (3 0 3 0 3 0))
Algorithm: BF

Expanded noded: 627
Generated noded: 2268

Elasped time: 0.198s
Solution:
((0 0 0 0 0 0) (0 0 0 0 0 0)) | g - 6
((0 0 0 0 0 0) (0 0 0 1 4 0)) | g - 5
((0 0 0 0 0 0) (0 0 3 0 3 0)) | g - 4
((0 0 0 4 1 0) (0 0 3 0 3 0)) | g - 3
((0 0 0 3 0 3) (0 0 3 0 3 0)) | g - 2
((1 0 0 3 0 3) (4 0 3 0 3 0)) | g - 1
((0 3 0 3 0 3) (3 0 3 0 3 0)) | g - 0
```

Após terminar a resolução do problema, o utilizador pode repetir todo o processo utilizando diferentes problemas e algoritmos.