

Manual de Utilizador - Jogo do Cavalo

Inteligência Artificial



- **Docente:** Filipe Mariano
- **Realizado por:** João Jardim Nº 20180002 SW-04

Índice

- [Introdução](#)
- [Objetivo](#)
- [Funcionamento do programa](#)

- [Limitações do programa](#)

1. Introdução

O Jogo do Cavalo é uma variante do problema matemático conhecido como o Passeio do Cavalo, cujo objetivo é, através dos movimentos do cavalo, visitar todas as casas de um tabuleiro similar ao de xadrez. Esta versão decorrerá num tabuleiro de 10 linhas e 10 colunas (10x10), em que cada casa possui uma pontuação.

Cada casa tem um valor associado, representado por um número e cada vez que o cavalo passa por uma casa acumula os pontos dessa mesma casa.

2. Objetivo

Atendendo ao que foi referido anteriormente o objetivo é que o cavalo consiga no menor número de jogadas alcançar um objetivo determinado de pontos. Para tal implementaram-se três algoritmos:

- Algoritmo de Busca em Largura (BFS)
- Algoritmo de Busca em Profundidade (DFS)
- Algoritmo A*

3. Funcionamento do programa

3.1 Iniciar o programa

Para abrir o programa o utilizador tem que abrir o LispWorks e carregar o ficheiro projeto.lisp.

```
; Loading text file
C:\Users\PC\Desktop\Universidade\IA2023\IA\Projeto\projeto.lisp
; Loading text file
C:\Users\PC\Desktop\Universidade\IA2023\IA\Projeto\puzzle.lisp
; Loading text file
C:\Users\PC\Desktop\Universidade\IA2023\IA\Projeto\procura.lisp

---- Press Space to continue ----
```

De seguida o utilizador deverá de iniciar o programa, da seguinte maneira:

```
CL-USER 11 : 1 > (menu-inicial)
Bem-vindo ao Jogo do Cavalo!
Escolha uma opção:
1. Jogar
2. Sair
Opção:
```

3.2. Menu Inicial

No menu inicial do jogo o utilizador tem duas opções, a opção de jogar ou opção de sair, terminando o programa. A imagem seguinte mostra o menu.

```
Bem-vindo ao Jogo do Cavalo!  
Escolha uma opção:  
1. Jogar  
2. Sair  
Opção:
```

3.3. Menu de Jogo

Ao seleccionar o menu de jogo, são apresentados os problemas a resolver, bem como o objetivo de pontos de cada um.

```
Escolha um problema:  
1. Problema A (Objetivo: 70 pontos)  
2. Problema B (Objetivo: 60 pontos)  
3. Problema C (Objetivo: 270 pontos)  
4. Problema D (Objetivo: 600 pontos)  
5. Problema E (Objetivo: 300 pontos)  
6. Problema F (Objetivo: 2000 pontos)  
0. Voltar  
Opção:
```

3.4. Menu de Escolha de Algoritmos

Depois de seleccionar o problema a resolver são apresentados para a escolha do utilizador os três algoritmos referidos no ponto 2.

```
Escolha um algoritmo para resolver o problema:  
1. BFS  
2. DFS  
3. A*  
0. Voltar  
Opção:
```

Ao seleccionar o algoritmo DFS, o utilizador terá de indicar qual será a profundidade máxima a ter em conta; se o utilizador não indicar a profundidade máxima será de 8, conforme indica na imagem seguinte:

```
Escolha um algoritmo para resolver o problema:  
1. BFS  
2. DFS  
3. A*
```

0. Voltar

Opção: 2

Escolha a profundidade máxima para o DFS (**pressione** Enter para valor por defeito 8):

Após seleccionar então o algoritmo depois é apresentado o resultado final obtido por esse algoritmo, as jogadas que foram realizadas bem como se o objetivo foi atingido. Neste exemplo foi escolhido o algoritmo BFS para resolver o problema B.

Algoritmo: BFS

Jogadas efetuadas:

```
((2 NIL 4 NIL 6 NIL 8 NIL 10 NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL 3 NIL 5 NIL 7 NIL 9 NIL 11)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL 22 NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
((NIL NIL 4 NIL 6 NIL 8 NIL 10 NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL T NIL 5 NIL 7 NIL 9 NIL 11)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL 22 NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
((NIL NIL T NIL 6 NIL 8 NIL 10 NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL 5 NIL 7 NIL 9 NIL 11)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL 22 NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
((NIL NIL NIL NIL 6 NIL 8 NIL 10 NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL T NIL 7 NIL 9 NIL 11)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL 22 NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
 (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
```

5 / 9

```
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL T)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
```

Estado final:

```
((NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL T)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
```

Nós Gerados: 9

Nós Expandidos: 9

Profundidade: 9

Pontos Acumulados: 65

Penetrância: 1.0

Fator de Ramificação Média: 0.9999896

Objetivo atingido.

Tendo em conta a particularidade do algoritmo para resolver um problema, são apresentadas as informações anteriores bem como a heurística do nó-solução (desde que objetivo atingido) ou último nó expandido no caso do objetivo não ter sido atingido.

Algoritmo: A*

Jogadas efetuadas:

```
((2 NIL 4 NIL 6 NIL 8 NIL 10 NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL 3 NIL 5 NIL 7 NIL 9 NIL 11)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL 22 NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
((NIL NIL 4 NIL 6 NIL 8 NIL 10 NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL T NIL 5 NIL 7 NIL 9 NIL 11))
```

7 / 9

```
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
((NIL NIL NIL NIL NIL NIL NIL NIL 10 NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL T NIL 11)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL 22 NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
((NIL NIL NIL NIL NIL NIL NIL NIL T NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL 11)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL 22 NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
((NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL T)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
```

Estado final:

```
((NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL T)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)
(NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
```

Nós Gerados: 9
Nós Expandidos: 9
Profundidade: 9
Pontos Acumulados: 65
Heurística: 0
Penetrância: 1.0

Fator de Ramificação Média: 0.9999896
Objetivo atingido.

O resultado obtido de cada algoritmo também é escrito num ficheiro "estatisticas.txt", presente na pasta deste projeto.

4. Limitações do programa

- Atendendo que não se conseguiu de uma maneira arbitrária modificar o caminho dos ficheiros:
 - Para inicializar o `projeto.lisp`: O utilizador terá de alterar o caminho conforme necessário, no início do ficheiro.

```
(load "C:/Users/PC/Desktop/Universidade/IA2023/IA/Projeto/puzzle.lisp")  
(load  
 "C:/Users/PC/Desktop/Universidade/IA2023/IA/Projeto/procura.lisp")
```

- No ficheiro `procura.lisp` na função `exibir-informacoes`:

```
(caminho-arquivo  
 "C:/Users/PC/Desktop/Universidade/IA2023/IA/Projeto/estatisticas.txt")
```

- No menu da escolha do problema a escolher se o utilizador escolher uma opção inválida o programa termina como tal o utilizador tem que reiniciar o programa.
- Para o jogador sair tem de voltar sempre ao menu inicial.