

# Projeto Nº2: Jogo do Cavalo

## Manual Técnico



No âmbito da Unidade Curricular de Inteligência Artificial

Docente: Prof. Joaquim Filipe

Elaborado por:

João Quintiliano nº 201900287

## Conteúdo

Introdução.....	3
Teoria de jogos.....	3
Composição do Projeto.....	4
Implementação .....	4
Operadores.....	5
Composição do nó .....	6
Funções seletoras do nó.....	6
Algoritmo.....	6
Minimax com cortes alfabeta .....	6
Requisitos não implementados .....	6

# Introdução

Este Manual Técnico tem como finalidade descrever os diferentes objetos que compõem o projeto, a sua implementação, os diferentes algoritmos, as limitações encontradas e os resultados obtidos.

Este projeto é uma continuação do projeto 1 do jogo do cavalo e tem como objetivo aplicar os conhecimentos adquiridos nas aulas no tema da teoria de jogos. Neste projeto já temos o jogo completo entre dois jogadores onde temos dois tipos de jogadores, humano e computador. Para as jogadas do computador foi utilizado o algoritmo alfabeta que permite prever a melhor jogada tendo em conta que este tem um outro jogador a querer prejudicar as suas jogadas.

Assim como no projeto o jogo do cavalo usa um tabuleiro de 10x10 como o de xadrez. O cavalo pode realizar movimentos em L como o cavalo no xadrez podendo fazer em todas as direções totalizando a possibilidade de 8 movimentos.

## Teoria de jogos

Sendo este um projeto no tema da teoria de jogos foram identificadas as seguintes características do jogo do cavalo:

- Jogo de soma-zero →
- Jogo combinatório →
- Jogo de 2 pessoas →
- Jogo simétrico → É um jogo simétrico uma vez que os resultados não dependem do jogador mas sim da estratégia

### Jogo de soma-zero

Trata-se de um jogo de soma-zero uma vez cada jogador beneficia da sua jogada podendo dar prejuízo ao adversário.

### Jogo combinatório

Trata-se de um jogo combinatório uma vez que é necessário explorar várias possibilidades de jogada para encontrar a melhor.

### Jogo de 2 pessoas

Trata-se de um jogo de 2 pessoas uma vez que temos dois jogadores adversários em que cada um tenta maximizar os seus pontos e minimizar os pontos do seu adversário.

### Jogo simétrico

É um jogo simétrico uma vez que os resultados não dependem do jogador, mas sim da estratégia aplicada.

### Jogo sequencial

Este é um jogo sequencial visto que cada jogador tem que esperar o outro jogador fazer a sua jogada tendo assim conhecimento da jogada antecessora.

# Composição do Projeto

O projeto foi implementado em Lisp Puro através do software LispWorks.

O Projeto foi dividido em 3 ficheiros distintos de forma a manter o projeto organizado:

- interact.lisp → Carrega os outros ficheiros de código, escreve e lê de ficheiros e trata da interação com o utilizador.
- jogo.lisp → Código relacionado com o problema.
- algoritmo.lisp → Deve conter a implementação do algoritmo de jogo independente do domínio.

## Implementação

### Tabuleiro

O tabuleiro é representado sob a forma de uma lista composta por 10 outras listas, cada uma delas com 10 átomos. Cada uma das listas representa uma linha do tabuleiro, enquanto cada um dos átomos possui o valor da respetiva casa. Cada linha do tabuleiro será representada por um número de 1 a 10 e cada coluna será representada por uma letra de A a J tal como num tabuleiro de xadrez. Cada átomo pode tomar o valor de 00 a 99. As casas que já foram visitadas deixaram de ter um valor atribuído impossibilitando que o cavalo possa cair nesta casa novamente.


	A	B	C	D	E	F	G	H	I	J	
1											1
2											2
3											3
4		88									4
5	44									79	5
6	32		34		27						6
7	55	87	72	58	81	40	17		38	95	7
8	00	29	70	42	13	10		78	14	24	8
9	56	01	71	39	84	77	03	25	63	66	9
10	65	90	89	45	46	02	57	31	86	53	10
	A	B	C	D	E	F	G	H	I	J	

Figura 1: Exemplo de tabuleiro

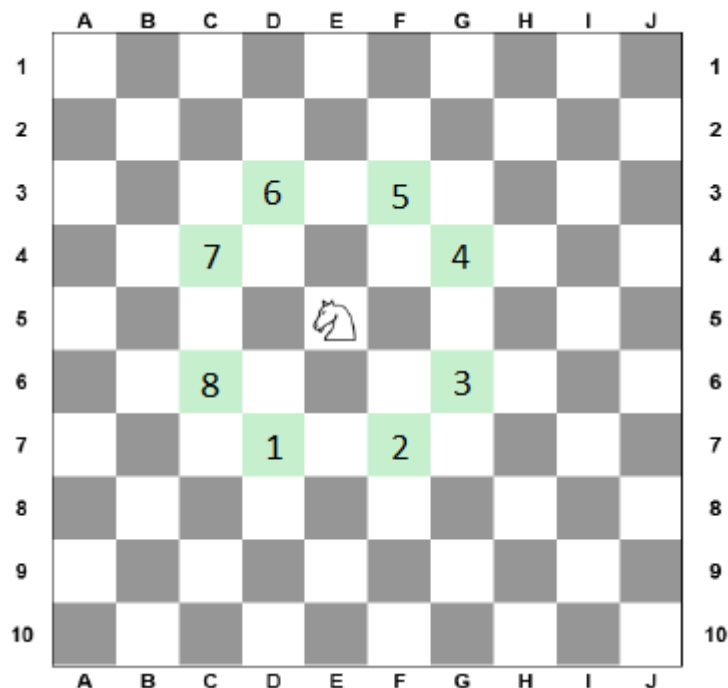
```
(  
  (NIL NIL NIL NIL NIL NIL NIL NIL NIL T)  
  (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)  
  (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL)  
  (NIL 88 NIL NIL NIL NIL NIL NIL NIL NIL)  
  (44 NIL NIL NIL NIL NIL NIL NIL NIL 79)  
  (32 NIL 34 NIL 27 NIL NIL NIL NIL NIL)  
  (55 87 72 58 81 40 17 NIL 38 95)  
  (0 29 70 42 13 10 NIL 78 14 24)  
  (56 1 71 39 84 77 3 25 63 66)  
  (65 90 89 45 46 2 57 31 86 53)  
)
```

Figura 2: Representação do tabuleiro

## Operadores

Os operadores representam os movimentos possíveis num determinado estado. Para o Problema do Cavalo o máximo de movimentos possíveis serão 8, o que consequentemente faz a necessidade da implementação de 8 operadores (linha, coluna):

- Operador 1  $\rightarrow$  (2, -1)
- Operador 2  $\rightarrow$  (2, 1)
- Operador 3  $\rightarrow$  (1, 2)
- Operador 4  $\rightarrow$  (-1, 2)
- Operador 5  $\rightarrow$  (-2, 1)
- Operador 6  $\rightarrow$  (-2, -1)
- Operador 7  $\rightarrow$  (-1, -2)
- Operador 8  $\rightarrow$  (1, -2)



## Regras de Movimentação

Regra da Colocação do Cavalo  $\rightarrow$  para iniciar o jogo, o cavalo só pode ser colocado numa casa da 1ª linha (A1 – J1)

Regra do Simétrico  $\rightarrow$  é atribuído NIL ao valor simétrico da casa visitada (exemplo: 67  $\rightarrow$  76)

Regra do Duplo  $\rightarrow$  é atribuído NIL ao valor duplo mais alto (ex: 99) se a casa visitada for um duplo (ex: 33, 55, 77)

## Composição do nó

O nó é composto da seguinte forma:

- tabuleiro
- profundidade do nó na árvore
- pontuação do jogador max
- pontuação do jogador min
- operadores do jogador max
- operadores do jogador min
- valor do nó
- posição do cavalo do jogador max no tabuleiro
- posição do cavalo do jogador min no tabuleiro

## Funções seletoras do nó

As funções seletoras do nó implementadas são as seguintes:

- no-tabuleiro
- no-profundidade
- no-pontuacao-max
- no-pontuacao-min
- no-operadores-max
- no-operadores-min
- no-valor
- no-posicao-max
- no-posicao-min

## Algoritmo

### Minimax com cortes alfabeta

O minimax é um algoritmo de decisão que permite obter a melhor jogada possível tendo um agente contrário a tentar minimizar o seu resultado. Existe então dois tipos de agentes neste algoritmo sendo o max que procura maximizar a sua pontuação e min que procura minimizar a pontuação do max. É gerada uma árvore das várias hipóteses, neste caso de jogadas,

## Requisitos não implementados

Não foram implementados os seguintes requisitos:

- As funções seletoras do nó implementadas são as seguintes:
- Implementação do limite de tempo para o computador
- Procura quiescente
- Memoização