



## **Relatório do 3º Trabalho Prático**

### **Inteligência Artificial**

#### **Autores:**

Miguel Portugal - n.º38128

Ricardo Oliveira - n.º42647

Vasco Barnabé - n.º42819

14 de Julho de 2021

# 1

## 1.1 (a)

**Exemplo de estado inicial:** estado\_inicial(e(2,0,0,0)).

## 1.2 (b)

**Estado terminal:** terminal(e(0,0,0,0)).

## 1.3 (c)

**Função de utilidade:**

valor(E,-10,P):- terminal(E), R is P mod 2, R=1.

valor(E,10,P):- terminal(E), R is P mod 2, R=0.

## 1.4 (d)

Instruções para compilar e executar o programa, utilizando a pesquisa minmax:

```
[minmax].  
g(nim).
```

*Ver ficheiros em anexo*

**Resultado (com o estado inicial: estado\_inicial(e(2,0,0,0)).)**

movimento: ret(1,1) - retira uma peça da posição 1.

estado resultante - e(1,0,0,0).

## 1.5 (e)

## 1.6 (f)

Utilizando **minmax** (ficheiro "minmaxCorte.pl"):

**Resultado (com o estado inicial: estado\_inicial(e(2,0,3,0)).)**

movimento: ret(3,1) - retira uma peça da posição 3.

estado resultante - e(2,0,2,0).

Utilizando **alfa-beta**:

Este algoritmo de pesquisa não foi implementado.

## 1.7 (g)

Instruções para compilar e executar o programa, utilizando o agente inteligente:

```
[minmax].  
[nim].  
joga.
```

*Ver ficheiros em anexo*

joga :- estado\_inicial(E), agente(E).

agente(E) :- terminal(E), valor(E,-10,\_), write(perdeu), nl.

agente(Ei) :- minimax\_decidir(Ei,Op), write(jogo(Op)), op1(Ei,Op,Es), write(Es), nl.

## 1.8 (h)

## 2 2

O jogo escolhido foi o 3 em linha (jogo conhecido como 4 em linha, sendo este mais simples em que basta uma sequência de 3 para ganhar).

Código fonte disponível no ficheiro "tlinha\_minmax.pl"

## 2.1 (a)

**Exemplo de estado inicial:** estado\_inicial([[1,1,1,1,1],[1,1,x,1,1],[o,x,o,x,o],[x,o,x,o,x]]).

## 2.2 (b)

**Estado terminal:**

terminal(G) :- linha(G,\_).

terminal(G) :- coluna(G,\_).

terminal(G) :- diagonal(G,\_).

terminal(G) :- completo(G\_).

## 2.3 (c)

**Função de utilidade:**

valor(G, 1) :- linhas(G,x).

valor(G, 1) :- colunas(G,x).

valor(G, 1) :- diagonal(G,x).

valor(G, -1) :- linhas(G,o).

valor(G, -1) :- colunas(G,o).

valor(G, -1) :- diagonal(G,o).

valor(\_, 0).

## 2.4 (d)

## 2.5 (e)