

# 2º Trabalho Prático

## Quadrado Mágico

Licenciatura em Eng. Informática  
Inteligência Artificial



**Helder Godinho 42741**

**Mariana Silva 54389**

**Docente: Irene Pimenta**

1. Considere a seguinte variante do jogo do 4 em linha. Joga-se num tabuleiro de 6 X 7. Considere que tem peças brancas e peças pretas. O primeiro a jogar é o jogador que tem as peças brancas. As peças podem ser colocadas em qq casa vazia como no jogo do galo.

- a) Escolha uma estrutura de dados para representar os estado do jogo.

---

```
1 % Estado do jogo: uma lista de listas (6x7) inicializada
  com células vazias
2 estado_inicial([ [v,v,v,v,v,v,v],
3                  [v,v,v,v,v,v,v],
4                  [v,v,v,v,v,v,v],
5                  [v,v,v,v,v,v,v],
6                  [v,v,v,v,v,v,v],
7                  [v,v,v,v,v,v,v] ]).
```

---

- b) Defina o predicado terminal(estado) que sucede quando o estado é terminal.

---

```
1
2 Um estado é terminal se um dos jogadores tiver 4 peças em
  linha ou se o tabuleiro estiver cheio.
3
4 terminal(E):- dividir_em_linhas(E,SL),E = e(L,J),(linhas(e
  (SL),J);
5               colunas(e(SL),J);/* diagonal(e(SL),J);*/
               cheio(L)).
6
7 cheio(L):- \+member(v,L).
8
9 linhas(e([]),_):- fail.
10 linhas(e([H|T]),X) :- linhas(H,X); linhas(e(T),X).
11 linhas([X,X,X,X|_],X):- X \= v.
12 linhas([_|T],X):- linhas(T,X).
13
14 colunas(e([]),_):- fail.
15 colunas(e([_,_,_,[]]),_):- fail.
16 colunas(e([A,B,C,D|T]),X) :- colunas(A,B,C,D,X); colunas(e
  ([B,C,D|T]),X).
17 colunas([X|_],[X|_],[X|_],[X|_],X):- X \= v.
18 colunas([_|A],[_|B],[_|C],[_|D],X):- colunas(A,B,C,D,X).
19 colunas([],[],[],[],_):- fail.
20
21
22 diagonal(e([]),_):- fail.
23 diagonal(e([_,_,_,[]]),_):- fail.
24 diagonal(e([A,B,C,D|T]),X) :- diagonal(A,B,C,D,X);
  diagonal(e([B,C,D|T]),X).
25 diagonal([X|_],[_,X|_],[_,_,X|_],[_,_,_,X|_],X):- X \= v.
26 diagonal([_,_,_,X|_],[_,_,X|_],[_,X|_],[X|_],X):- X \= v.
27 diagonal([_|A],[_|B],[_|C],[_|D],X):- diagonal(A,B,C,D,X).
28 diagonal([],[],[],[],_):- fail.
29
```



```

30 dividir_em_linhas(e(L,_), S) :- divide(L,S).
31 divide([],[]).
32 divide([V1,V2,V3,V4,V5,V6,V7|T],[[V1,V2,V3,V4,V5,V6,V7|T1
    ]):- divide(T,T1).

```

- c) Defina uma função de utilidade que para um estado terminal deve retornar o valor do estado (ex: -1 perde, 0 empata, 1 ganha).

```

1 valor(e(L,_),0,P):- \+ member(v,L),!.
2 valor(E,V,P):-
3     terminal(E),
4     X is P mod 2,
5     ( X == 1, V = 1 ; X == 0, V = -1 ).

```

- d) Use a implementação da pesquisa minimax dada na aula pratica para escolher a melhor jogada num estado. Teste a sua descrição do jogo com vários estados.

```

1 estado_inicial(e([      b,w,b,b,w,b,w,
2                               w,b,w,w,w,b,w,
3                               v,v,v,v,b,w,b,
4                               w,b,v,w,w,b,w,
5                               w,w,b,w,b,w,b,
6                               w,b,w,b,w,b,w], w)
    ).

```

7 Resultado:24

```

1 estado_inicial(e([v,b,w,v,b,w,b,
2                   w,b,v,w,w,b,w,
3                   b,w,b,b,w,b,w,
4                   w,b,w,w,w,b,w,
5                   v,w,b,w,b,w,b,
6                   w,b,w,b,w,b,w], w)).

```

7 Resultado:29

```

1 estado_inicial(e([w,b,w,b,w,b,w,
2                   w,b,w,b,w,b,w,
3                   w,b,w,b,w,b,w,
4                   b,w,b,w,b,w,b,
5                   b,w,b,w,b,w,b,
6                   w,b,w,b,w,b,v], w)).

```

7 Resultado:42

```

1 estado_inicial(e([w,b,w,v,w,b,w,
2                   w,b,w,b,w,b,w,
3                   b,w,b,w,b,w,b,
4                   w,b,w,b,w,b,w,
5                   b,w,b,w,b,w,b,
6                   w,b,w,b,w,b,w], w)).

```

7 Resultado:4



---

```
1 estado_inicial(e([w,b,w,b,w,b,w,
2                   b,w,b,w,b,w,b,
3                   w,b,w,b,w,b,w,
4                   b,w,b,w,b,w,b,
5                   w,b,w,b,w,b,w,
6                   b,w,b,w,b,v,w], w)).
7 Resultado:41
```

---

e) Implemente a pesquisa Alfa-Beta e compare os resultados (tempo e espaço).

---

```
1 estado_inicial(e([      b,w,b,b,w,b,w,
2                               w,b,w,w,w,b,w,
3                               v,v,v,v,b,w,b,
4                               w,b,v,w,w,b,w,
5                               w,w,b,w,b,w,b,
6                               w,b,w,b,w,b,w], w)
3                               ).
```

```
7 Melhor Jogada Minimax: 24
8 Tempo de Execução Minimax: 20 ms
9 uncaught exception: error(instantiation_error,(is)/2)
```

---

```
1 estado_inicial(e([v,b,w,v,b,w,b,
2                   w,b,v,w,w,b,w,
3                   b,w,b,b,w,b,w,
4                   w,b,w,w,w,b,w,
5                   v,w,b,w,b,w,b,
6                   w,b,w,b,w,b,w], w)).
```

```
7 Melhor Jogada Minimax: 29
8 Tempo de Execução Minimax: 4 ms
9 Melhor Jogada Alfa-Beta: 29
10 Tempo de Execução Alfa-Beta: 2 ms
```

---

---

```
1 estado_inicial(e([w,b,w,b,w,b,w,
2                   w,b,w,b,w,b,w,
3                   w,b,w,b,w,b,w,
4                   b,w,b,w,b,w,b,
5                   b,w,b,w,b,w,b,
6                   w,b,w,b,w,b,v], w)).
```

```
7 Melhor Jogada Minimax: 42
8 Tempo de Execução Minimax: 0 ms
9 Melhor Jogada Alfa-Beta: 42
10 Tempo de Execução Alfa-Beta: 0 ms
```

---

---

```
1 estado_inicial(e([w,b,w,v,w,b,w,
2                   w,b,w,b,w,b,w,
3                   b,w,b,w,b,w,b,
4                   w,b,w,b,w,b,w,
5                   b,w,b,w,b,w,b,
6                   w,b,w,b,w,b,w], w)).
```

---



```

7 Melhor Jogada Minimax: 4
8 Tempo de Execução Minimax: 1 ms
9 Melhor Jogada Alfa-Beta: 4
10 Tempo de Execução Alfa-Beta: 0 ms

```

```

1 estado_inicial(e([w,b,w,b,w,b,w,
2                   b,w,b,w,b,w,b,
3                   w,b,w,b,w,b,w,
4                   b,w,b,w,b,w,b,
5                   w,b,w,b,w,b,w,
6                   b,w,b,w,b,v,w], w))).
7 Melhor Jogada Minimax: 41
8 Tempo de Execução Minimax: 0 ms
9 Melhor Jogada Alfa-Beta: 41
10 Tempo de Execução Alfa-Beta: 0 ms

```

- f) Defina uma função de avaliação que estime o valor de cada estado do jogo use os dois algoritmos anteriores com corte em profundidade e compare os resultados (tempo e espaço).

```

1 ã
2 No sei

```

- g) Implemente um agente inteligente que joga o 4 linha: 1 - Joga uma peça, atualiza e mostra o tabuleiro 2 - Lê a jogada do adversário e actualiza e mostra o tabuleiro, volta a 1 até o jogo terminar.

```

1 ã
2 No sei

```

- h) Apresente uma tabela com o número de nós expandidos para diferentes estados do jogo (10 no mínimo) com os vários algoritmos.

```

1 ã
2 No Sei

```

2. Escolha um jogo de dois jogadores diferente do jogo do galo e do jogo do 31 (por exemplo o jogo do NIM).

- a) Escolha uma estrutura de dados para representar os estado do jogo.

```

1 e(1, 2, 2, 5)

```

- b) Defina o predicado terminal(estado) que sucede quando um estado é terminal.

```

1 terminal(e(0, 0, 0, 0)).

```

- c) Defina uma função de utilidade que para um estado terminal que deve retornar o valor do estado.

```

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

```

```
4
5 valor(E, 1, P):- terminal(E),
6                               R is P mod 2,
7                               R=0.
```

---

- d) Implemente um agente inteligente que joga o jogo que escolheu usando várias estratégias, minimax, corte alfa-beta e corte em profundidade.
- 
- 

- d) Apresente uma tabela com o número de nós expandidos para diferentes estados do jogo (10 no mínimo) com os vários algoritmos.
- 
-