

# Manual Técnico

---

Inteligência Artificial - Escola Superior de Tecnologia de Setúbal

2024/2025

Prof. Joaquim Filipe

Eng. Filipe Mariano

202000634 Bruno Ascensão

202000584 Francisco Pereira

Projeto Nº 2: Época Normal

24/01/2025

---

## Índice

---

1. Algoritmo
2. Tipos abstratos dados
3. Limitações e Opções tomadas
  - 3.1. Memoização
  - 3.2. Tempo limite
  - 3.3. Profundidade
  - 3.4. Tabuleiro do utilizador
- 3.5. Análise crítica e estatística

---

## 1. Algoritmo

---

O algoritmo implementado foi o MiniMax com cortes Alpha Beta e memoização.

```
(DEFUN core (node depth &optional (alpha most-negative-fixnum) (beta most-  
positive-fixnum) (is-max-player T))
```

```

"
  Arguments
    - node (Node)
    - depth (int)
    - &optional alpha (integer or most-negative-fixnum)
    - &optional beta (integer or most-positive-fixnum)
    - &optional is-max-player (boolean or T)

  Returns
    - (node heuristic-value 1 0 0)
    - (node heuristic-value analysed-nodes alpha-cuts beta-cuts): From min/max
  helpers

  Executes the minimax with alpha-beta pruning. It returns with the node
  evaluation, the node and some statistics related to the execution.
"

  (IF (gethash node algorithm-hashtable) ;Verifies if the node is cached and
  return it
    (LIST node (gethash node algorithm-hashtable) 1 0 0)
    (COND ;Execute normally to find the node value
      ((OR (ZEROP depth) (FUNCALL terminal node) (NULL (FUNCALL spawner node)))
        ;If one of the end conditions
        (LET (
          (heuristic-value (FUNCALL heuristic node)) ;Get the heuristic value of
  the node
          (depth-calc (IF (FUNCALL terminal node) depth 0)) ;If a node is
  terminal give him a weight using the depth.
          ) ; The if validating the is-max-player is to inver the values, meaning
  if the node evaluation of a min node is negative to THAT PLAYER, means it positive
  to the max player, needing to flip signal
          (PROGN
            (IF memoization-usage (sethash node algorithm-hashtable heuristic-
  value)) ;Link heuristic value to node on the hash-table (if is to use memoization)
            (LIST node (* (IF is-max-player 1 -1) (1+ depth-calc) heuristic-value)
  1 0 0)
          )
        )
      )
    (is-max-player (max-node (children-spawner-sorter node is-max-player) depth
  alpha beta)) ;minimax maximizer helper
    (T (min-node (children-spawner-sorter node is-max-player) depth alpha beta))
  ;minimax minimizer helper
    )
  )
)

(DEFUN max-node (children children-depth alpha beta &optional (value most-
  negative-fixnum) (node NIL) (nodes-analysed 0) (alpha-cuts 0) (beta-cuts 0))
  "
  Max node helper
  "
  (IF (NULL children)
    (LIST node value nodes-analysed alpha-cuts beta-cuts)
    (LET* (

```

```

    (core-evaluation (core (CAR children) (1- children-depth) alpha beta NIL))
;Execute the minimax evaluation
    (nodes-analysed (+ nodes-analysed (NTH 2 core-evaluation))) ;Get the
amount of analysed nodes
    (alpha-cuts (+ alpha-cuts (NTH 3 core-evaluation))) ;Get the amount of
alpha cuts
    (beta-cuts (+ beta-cuts (NTH 4 core-evaluation))) ;Get the amount of beta
cuts

    (old-value value) ;saves the old value
    (value (max value (NTH 1 core-evaluation)))
    (node (IF (NOT (= old-value value)) (CAR children) node))
  )
  (IF (> value beta) ;Fail hard, beta cut
    (LIST node value nodes-analysed alpha-cuts (1+ beta-cuts))
    (max-node (CDR children) children-depth (max alpha value) beta value node
nodes-analysed alpha-cuts beta-cuts)
  )
)
)
)

(DEFUN min-node (children children-depth alpha beta &optional (value most-
positive-fixnum) (node NIL) (nodes-analysed 0) (alpha-cuts 0) (beta-cuts 0))
  "
    Min node helper
  "
  (IF (NULL children)
    (LIST node value nodes-analysed alpha-cuts beta-cuts)
    (LET* (
      (core-evaluation (core (CAR children) (1- children-depth) alpha beta T))
;Execute the minimax evaluation
      (nodes-analysed (+ nodes-analysed (NTH 2 core-evaluation))) ;Get the
amount of analysed nodes
      (alpha-cuts (+ alpha-cuts (NTH 3 core-evaluation))) ;Get the amount of
alpha cuts
      (beta-cuts (+ beta-cuts (NTH 4 core-evaluation))) ;Get the amount of beta
cuts

      (old-value value) ;saves the old value
      (value (min value (NTH 1 core-evaluation)))
      (node (IF (NOT (= old-value value)) (CAR children) node))
    )
    (IF (< value alpha) ;Fail hard, alpha cut
      (LIST node value nodes-analysed (1+ alpha-cuts) beta-cuts)
      (min-node (CDR children) children-depth alpha (min beta value) value node
nodes-analysed alpha-cuts beta-cuts)
    )
  )
)
)
)

```

Os métodos a serem chamados utilizando o "**funcall**" no core, são métodos que já foram colocados nas variáveis da closure onde o algoritmo está implementado.

- Terminal: Função que avalia se o nó é um nó-folha;
- Spawner: Função que gera descendentes de um nó;
- Heuristic: Função que avalia heurísticamente um nó.

---

## 2. Tipos abstratos dados

---

Os nós estão estruturados através de uma lista com quatro elementos:

- Jogada;
- Estado;
- Pontos;
- Linha a jogar.

```
((0 1) ((3 0 2 2 2 2) (0 2 2 2 2 2)) (3 0) 1)
```

---

## 3. Limitações e Opções tomadas

---

### 3.1. Memoização

Por estar a ser utilizada a versão free do "Lisp Works", existe uma limitação de memória que faz com que a memoização não possa ser aplicada numa árvore muito grande. Para contrariar isto, é perguntado ao utilizador se pretende utilizar a memoização. Esta pode ser utilizada apenas em árvores mais pequenas, por exemplo, um tabuleiro que tenha em todas as posições 2 peças.

---

### 3.2. Tempo limite

Não foi implementado nenhuma função com o objetivo de limitar o tempo de procura do algoritmo. Isto porque, ao fazer uma paragem enquanto o algoritmo ainda está a ser executado, iria fazer com que a árvore a meio da execução ficasse com um corte vertical. Para evitar isso, não foi implementado nenhum algoritmo de limitação de tempo de procura.

Foi ainda tentado criar uma fórmula matemática que conjugasse a quantidade de células vazias com o tempo limite achando assim a profundidade ideal para o tempo limite indicado. Isto não foi possível por a complexidade da árvore ser explosão combinatória.

Para mitigar o problema descrito anteriormente, foi decidido fazer um corte a meio da árvore quando o tempo limite é atingido. Esta solução não é a ideal contudo segue o requisito do tempo limite.

### 3.3. Profundidade

Devido à limitação descrita acima no tempo limite, ficou decidido que a profundidade seria sempre 10. Isto faz com que num tabuleiro o tempo máximo de execução seja de 21 segundos, tal como acontece no tabuleiro que tem 8 peças em todas as posições.

### 3.4. Tabuleiro do utilizador

Por uma questão de facilidade para o utilizador, ficou definido que em jogos onde existe pelo menos um utilizador o tabuleiro inicial teria, ao invés de oito peças em todas as posições teria apenas duas.

## 4. Análise crítica e estatística

Após analisar o ficheiro "log.dat" que guarda as estatísticas de todas as jogadas, quer de utilizadores quer de IAs, é possível verificar que a soma dos cortes alpha e beta não ultrapassam o número de cortes. Também é possível confirmar que, à partida, a IA nas suas jogadas toma a melhor decisão consoante o cenário. Relativamente ao tempo de execução, este é mais elevado em nós com muitas peças e sem células vazias. Este tempo, de acordo com as estatísticas não ultrapassa os 21 segundos no maior tabuleiro possível, o que tem oito peças em todas as posições.

```

===== Player vs AI =====
====  PC  ====
=====
-> 2|2|2|2|2|2  points: 0
    -----
    2|2|2|2|2|2  points: 0
=====

No original: (NIL ((2 2 2 2 2 2) (2 2 2 2 2 2)) (0 0) 0)
No solucao: ((0 1) ((3 0 2 2 2 2) (0 2 2 2 2 2)) (3 0) 1)
Valor heuristico: 4
Nos analisados: 50751
Cortes | Alpha: 13404 | Beta: 4361
Tempo | Maximo: 10 | Execucao: 0.332
-----

====  HUMAN  ====
=====
    3|0|2|2|2|2  points: 3
    -----
-> 0|2|2|2|2|2  points: 0
=====

====  PC  ====
=====
-> 3|0|2|2|2|0  points: 3
    -----

```

```

0|2|2|2|0|3  points: 3
=====

No original: ((1 4) ((3 0 2 2 2 0) (0 2 2 2 0 3)) (3 3) 0)
No solucao: ((0 0) ((0 0 2 2 2 0) (1 3 0 2 0 3)) (6 3) 1)
Valor heuristico: 3
Nos analisados: 12564
Cortes | Alpha: 2675 | Beta: 1641
Tempo | Maximo: 10 | Execucao: 0.082
-----

==== HUMAN  ====
=====
0|0|2|2|2|0  points: 6
-----
-> 1|3|0|2|0|3  points: 3
=====

==== PC  ====
=====
-> 0|0|2|0|3|1  points: 6
-----
1|3|0|2|0|0  points: 6
=====

No original: ((1 5) ((0 0 2 0 3 1) (1 3 0 2 0 0)) (6 6) 0)
No solucao: ((0 4) ((0 1 3 1 0 1) (1 3 0 2 0 0)) (6 6) 1)
Valor heuristico: 2
Nos analisados: 8524
Cortes | Alpha: 2020 | Beta: 1029
Tempo | Maximo: 10 | Execucao: 0.04
-----

==== HUMAN  ====
=====
0|1|3|1|0|1  points: 6
-----
-> 1|3|0|2|0|0  points: 6
=====

==== PC  ====
=====
-> 0|1|3|1|0|1  points: 6
-----
1|0|1|3|1|0  points: 6
=====

No original: ((1 1) ((0 1 3 1 0 1) (1 0 1 3 1 0)) (6 6) 0)
No solucao: ((0 2) ((1 2 0 1 0 1) (2 0 1 3 1 0)) (6 6) 1)
Valor heuristico: 4
Nos analisados: 14617
Cortes | Alpha: 3814 | Beta: 1791
Tempo | Maximo: 10 | Execucao: 0.074
-----

```

```

===== HUMAN =====
=====
1|2|0|1|0|1 points: 6
-----
-> 2|0|1|3|1|0 points: 6
=====

===== PC =====
=====
-> 1|2|0|1|0|1 points: 6
-----
0|1|2|3|1|0 points: 6
=====

No original: ((1 0) ((1 2 0 1 0 1) (0 1 2 3 1 0)) (6 6) 0)
No solucao: ((0 0) ((0 2 0 1 0 1) (0 1 2 3 1 0)) (7 6) 1)
Valor heuristico: 5
Nos analisados: 6515
Cortes | Alpha: 1847 | Beta: 680
Tempo | Maximo: 10 | Execucao: 0.033
-----

===== HUMAN =====
=====
0|2|0|1|0|1 points: 7
-----
-> 0|1|2|3|1|0 points: 6
=====

===== PC =====
=====
-> 0|2|0|1|0|1 points: 7
-----
0|0|3|3|1|0 points: 6
=====

No original: ((1 1) ((0 2 0 1 0 1) (0 0 3 3 1 0)) (7 6) 0)
No solucao: ((0 1) ((1 0 0 1 0 1) (0 0 3 3 1 0)) (8 6) 1)
Valor heuristico: 5
Nos analisados: 2235
Cortes | Alpha: 1270 | Beta: 312
Tempo | Maximo: 10 | Execucao: 0.017
-----

===== HUMAN =====
=====
1|0|0|1|0|1 points: 8
-----
-> 0|0|3|3|1|0 points: 6
=====

===== PC =====
=====

```

```

-> 1|0|0|1|0|1  points: 8
-----
0|0|3|3|0|1  points: 6
=====

No original: ((1 4) ((1 0 0 1 0 1) (0 0 3 3 0 1)) (8 6) 0)
No solucao: ((0 0) ((0 0 0 1 0 1) (0 0 3 3 0 1)) (9 6) 1)
Valor heuristico: 11
Nos analisados: 1289
Cortes | Alpha: 921 | Beta: 185
Tempo | Maximo: 10 | Execucao: 0.008
-----

====  HUMAN  ====
=====
0|0|0|1|0|1  points: 9
-----
-> 0|0|3|3|0|1  points: 6
=====

====  PC  ====
=====
-> 0|0|0|1|0|2  points: 9
-----
0|0|3|3|0|0  points: 6
=====

No original: ((1 5) ((0 0 0 1 0 2) (0 0 3 3 0 0)) (9 6) 0)
No solucao: ((0 5) ((0 0 0 2 1 0) (0 0 3 3 0 0)) (9 6) 1)
Valor heuristico: 16
Nos analisados: 732
Cortes | Alpha: 450 | Beta: 99
Tempo | Maximo: 10 | Execucao: 0.009
-----

====  HUMAN  ====
=====
0|0|0|2|1|0  points: 9
-----
-> 0|0|3|3|0|0  points: 6
=====

====  PC  ====
=====
-> 0|0|0|2|1|0  points: 9
-----
0|0|3|0|1|1  points: 7
=====

No original: ((1 3) ((0 0 0 2 1 0) (0 0 3 0 1 1)) (9 7) 0)
No solucao: ((0 4) ((0 0 0 3 0 0) (0 0 3 0 1 1)) (9 7) 1)
Valor heuristico: 8
Nos analisados: 706
Cortes | Alpha: 419 | Beta: 137

```



Tempo | Maximo: 10 | Execucao: 0.009

==== HUMAN ====

=====

0|0|0|3|0|0 points: 9

-----

-> 0|0|3|0|1|1 points: 7

=====

==== PC ====

=====

-> 0|0|0|3|0|0 points: 9

-----

0|0|0|1|2|2 points: 7

=====

No original: ((1 2) ((0 0 0 3 0 0) (0 0 0 1 2 2)) (9 7) 0)

No solucao: ((0 3) ((1 1 1 0 0 0) (0 0 0 1 2 2)) (9 7) 1)

Valor heuristico: 23

Nos analisados: 295

Cortes | Alpha: 169 | Beta: 52

Tempo | Maximo: 10 | Execucao: 0.0

==== HUMAN ====

=====

1|1|1|0|0|0 points: 9

-----

-> 0|0|0|1|2|2 points: 7

=====

==== PC ====

=====

-> 1|1|1|0|0|0 points: 9

-----

0|0|0|1|0|3 points: 8

=====

No original: ((1 4) ((1 1 1 0 0 0) (0 0 0 1 0 3)) (9 8) 0)

No solucao: ((0 2) ((1 2 0 0 0 0) (0 0 0 1 0 3)) (9 8) 1)

Valor heuristico: 23

Nos analisados: 346

Cortes | Alpha: 208 | Beta: 29

Tempo | Maximo: 10 | Execucao: 0.0

==== HUMAN ====

=====

1|2|0|0|0|0 points: 9

-----

-> 0|0|0|1|0|3 points: 8

=====

```

==== PC ====
=====
-> 1|2|0|0|0|0 points: 9
-----
0|0|0|0|1|3 points: 8
=====

No original: ((1 3) ((1 2 0 0 0 0) (0 0 0 0 1 3)) (9 8) 0)
No solucao: ((0 1) ((2 0 0 0 0 0) (0 0 0 0 1 3)) (10 8) 1)
Valor heuristico: 23
Nos analisados: 101
Cortes | Alpha: 69 | Beta: 7
Tempo | Maximo: 10 | Execucao: 0.0
-----

==== HUMAN ====
=====
2|0|0|0|0|0 points: 10
-----
-> 0|0|0|0|1|3 points: 8
=====

==== PC ====
=====
-> 2|0|0|0|1|1 points: 10
-----
0|0|0|0|1|0 points: 9
=====

No original: ((1 5) ((2 0 0 0 1 1) (0 0 0 0 1 0)) (10 9) 0)
No solucao: ((0 0) ((0 0 0 0 1 1) (1 0 0 0 1 0)) (11 9) 1)
Valor heuristico: 23
Nos analisados: 175
Cortes | Alpha: 91 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
-----

==== HUMAN ====
=====
0|0|0|0|1|1 points: 11
-----
-> 1|0|0|0|1|0 points: 9
=====

==== PC ====
=====
-> 0|0|0|0|1|1 points: 11
-----
1|0|0|0|0|1 points: 9
=====

No original: ((1 4) ((0 0 0 0 1 1) (1 0 0 0 0 1)) (11 9) 0)
No solucao: ((0 4) ((0 0 0 1 0 1) (1 0 0 0 0 1)) (11 9) 1)
Valor heuristico: 35

```

```
Nos analisados: 89
Cortes | Alpha: 101 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
```

```
===== HUMAN =====
```

```
=====
```

```
0|0|0|1|0|1 points: 11
```

```
-----
```

```
-> 1|0|0|0|0|1 points: 9
```

```
=====
```

```
===== PC =====
```

```
=====
```

```
-> 0|0|0|1|0|1 points: 11
```

```
-----
```

```
0|1|0|0|0|1 points: 9
```

```
=====
```

```
No original: ((1 0) ((0 0 0 1 0 1) (0 1 0 0 0 1)) (11 9) 0)
```

```
No solucao: ((0 3) ((0 0 1 0 0 1) (0 1 0 0 0 1)) (11 9) 1)
```

```
Valor heuristico: 32
```

```
Nos analisados: 36
```

```
Cortes | Alpha: 50 | Beta: 10
```

```
Tempo | Maximo: 10 | Execucao: 0.0
```

```
===== HUMAN =====
```

```
=====
```

```
0|0|1|0|0|1 points: 11
```

```
-----
```

```
-> 0|1|0|0|0|1 points: 9
```

```
=====
```

```
===== PC =====
```

```
=====
```

```
-> 0|0|1|0|0|1 points: 11
```

```
-----
```

```
0|0|1|0|0|1 points: 9
```

```
=====
```

```
No original: ((1 1) ((0 0 1 0 0 1) (0 0 1 0 0 1)) (11 9) 0)
```

```
No solucao: ((0 2) ((0 1 0 0 0 1) (0 0 1 0 0 1)) (11 9) 1)
```

```
Valor heuristico: 23
```

```
Nos analisados: 50
```

```
Cortes | Alpha: 36 | Beta: 9
```

```
Tempo | Maximo: 10 | Execucao: 0.0
```

```
===== HUMAN =====
```

```
=====
```

```
0|1|0|0|0|1 points: 11
```

```
-----
```

```
-> 0|0|1|0|0|1 points: 9
```

```

=====

====  PC  ====
=====
-> 0|1|0|0|0|1  points: 11
-----
    0|0|0|1|0|1  points: 9
=====

No original: ((1 2) ((0 1 0 0 0 1) (0 0 0 1 0 1)) (11 9) 0)
No solucao: ((0 1) ((1 0 0 0 0 1) (0 0 0 1 0 1)) (11 9) 1)
Valor heuristico: 12
Nos analisados: 53
Cortes | Alpha: 23 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
-----

====  HUMAN  ====
=====
    1|0|0|0|0|1  points: 11
-----
-> 0|0|0|1|0|1  points: 9
=====

====  PC  ====
=====
-> 1|0|0|0|0|1  points: 11
-----
    0|0|0|0|1|1  points: 9
=====

No original: ((1 3) ((1 0 0 0 0 1) (0 0 0 0 1 1)) (11 9) 0)
No solucao: ((0 0) ((0 0 0 0 0 1) (0 0 0 0 1 1)) (12 9) 1)
Valor heuristico: 12
Nos analisados: 14
Cortes | Alpha: 7 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
-----

====  HUMAN  ====
=====
    0|0|0|0|0|1  points: 12
-----
-> 0|0|0|0|1|1  points: 9
=====

====  PC  ====
=====
-> 0|0|0|0|0|1  points: 12
-----
    0|0|0|0|0|2  points: 9
=====

No original: ((1 4) ((0 0 0 0 0 1) (0 0 0 0 0 2)) (12 9) 0)

```

```
No solucao: ((0 5) ((0 0 0 0 1 0) (0 0 0 0 0 2)) (12 9) 1)
Valor heuristico: 23
Nos analisados: 2
Cortes | Alpha: 0 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
```

---

```
==== HUMAN  ====
```

```
=====
0|0|0|0|1|0  points: 12
-----
-> 0|0|0|0|0|2  points: 9
=====
```

```
==== PC  ====
```

```
=====
-> 0|0|0|0|2|1  points: 12
-----
0|0|0|0|0|0  points: 9
=====
```

```
No original: ((1 5) ((0 0 0 0 2 1) (0 0 0 0 0 0)) (12 9) 0)
No solucao: ((0 5) ((0 0 0 0 3 0) (0 0 0 0 0 0)) (12 9) 1)
Valor heuristico: 23
Nos analisados: 2
Cortes | Alpha: 0 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
```

---

```
==== HUMAN !SKIPPING!  ====
```

```
=====
0|0|0|0|3|0  points: 12
-----
-> 0|0|0|0|0|0  points: 9
=====
```

```
==== PC  ====
```

```
=====
-> 0|0|0|0|3|0  points: 12
-----
0|0|0|0|0|0  points: 9
=====
```

```
No original: ((0 5) ((0 0 0 0 3 0) (0 0 0 0 0 0)) (12 9) 0)
No solucao: ((0 4) ((0 1 1 1 0 0) (0 0 0 0 0 0)) (12 9) 1)
Valor heuristico: 21
Nos analisados: 1
Cortes | Alpha: 0 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
```

---

```
==== HUMAN !SKIPPING!  ====
```

```
=====
0|1|1|1|0|0  points: 12
```

```

-----
-> 0|0|0|0|0|0  points: 9
=====

====  PC  ====
=====
-> 0|1|1|1|0|0  points: 12
-----
    0|0|0|0|0|0  points: 9
=====

No original: ((0 4) ((0 1 1 1 0 0) (0 0 0 0 0 0)) (12 9) 0)
No solucao: ((0 2) ((0 2 0 1 0 0) (0 0 0 0 0 0)) (12 9) 1)
Valor heuristico: 22
Nos analisados: 3
Cortes | Alpha: 0 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
-----

====  HUMAN !SKIPPING!  ====
=====
    0|2|0|1|0|0  points: 12
-----
-> 0|0|0|0|0|0  points: 9
=====

====  PC  ====
=====
-> 0|2|0|1|0|0  points: 12
-----
    0|0|0|0|0|0  points: 9
=====

No original: ((0 2) ((0 2 0 1 0 0) (0 0 0 0 0 0)) (12 9) 0)
No solucao: ((0 1) ((1 0 0 1 0 0) (0 0 0 0 0 0)) (13 9) 1)
Valor heuristico: 32
Nos analisados: 2
Cortes | Alpha: 0 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
-----

====  HUMAN !SKIPPING!  ====
=====
    1|0|0|1|0|0  points: 13
-----
-> 0|0|0|0|0|0  points: 9
=====

====  PC  ====
=====
-> 1|0|0|1|0|0  points: 13
-----
    0|0|0|0|0|0  points: 9
=====

```

```
No original: ((0 1) ((1 0 0 1 0 0) (0 0 0 0 0 0)) (13 9) 0)
No solucao: ((0 0) ((0 0 0 1 0 0) (0 0 0 0 0 0)) (14 9) 1)
Valor heuristico: 43
Nos analisados: 2
Cortes | Alpha: 0 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
```

---

```
==== HUMAN !SKIPPING! ====
=====
0|0|0|1|0|0 points: 14
-----
-> 0|0|0|0|0|0 points: 9
=====
```

```
==== PC ====
=====
-> 0|0|0|1|0|0 points: 14
-----
0|0|0|0|0|0 points: 9
=====
```

```
No original: ((0 0) ((0 0 0 1 0 0) (0 0 0 0 0 0)) (14 9) 0)
No solucao: ((0 3) ((0 0 1 0 0 0) (0 0 0 0 0 0)) (14 9) 1)
Valor heuristico: 43
Nos analisados: 1
Cortes | Alpha: 0 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
```

---

```
==== HUMAN !SKIPPING! ====
=====
0|0|1|0|0|0 points: 14
-----
-> 0|0|0|0|0|0 points: 9
=====
```

```
==== PC ====
=====
-> 0|0|1|0|0|0 points: 14
-----
0|0|0|0|0|0 points: 9
=====
```

```
No original: ((0 3) ((0 0 1 0 0 0) (0 0 0 0 0 0)) (14 9) 0)
No solucao: ((0 2) ((0 1 0 0 0 0) (0 0 0 0 0 0)) (14 9) 1)
Valor heuristico: 43
Nos analisados: 1
Cortes | Alpha: 0 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
```

---

```
==== HUMAN !SKIPPING! ====
```

```

=====
    0|1|0|0|0|0  points: 14
    -----
-> 0|0|0|0|0|0  points: 9
=====

====  PC  ====
=====
-> 0|1|0|0|0|0  points: 14
    -----
    0|0|0|0|0|0  points: 9
=====

No original: ((0 2) ((0 1 0 0 0 0) (0 0 0 0 0 0)) (14 9) 0)
No solucao: ((0 1) ((1 0 0 0 0 0) (0 0 0 0 0 0)) (14 9) 1)
Valor heuristico: 43
Nos analisados: 1
Cortes | Alpha: 0 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
-----

====  HUMAN !SKIPPING!  ====
=====
    1|0|0|0|0|0  points: 14
    -----
-> 0|0|0|0|0|0  points: 9
=====

====  PC  ====
=====
-> 1|0|0|0|0|0  points: 14
    -----
    0|0|0|0|0|0  points: 9
=====

No original: ((0 1) ((1 0 0 0 0 0) (0 0 0 0 0 0)) (14 9) 0)
No solucao: ((0 0) ((0 0 0 0 0 0) (0 0 0 0 0 0)) (15 9) 1)
Valor heuristico: 99990
Nos analisados: 1
Cortes | Alpha: 0 | Beta: 0
Tempo | Maximo: 10 | Execucao: 0.0
-----

=====
    0|0|0|0|0|0  points: 15
    -----
-> 0|0|0|0|0|0  points: 9
=====

=====
WINNER: Player from the Top row
=====

```