

Manual Técnico - Jogo do Cavalo (Projeto 2)

Inteligência Artificial



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

Índice

- [Introdução](#)
- [Arquitetura do Sistema](#)
- [Entidades e a sua implementação](#)

- [Algoritmo Alfabeta](#)
- [Análise da eficiência do algoritmo](#)
- [Limitações e requisitos não implementados](#)

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.

O jogo do cavalo joga-se com 2 jogadores em que cada jogador possui uma peça do xadrez tradicional, que é o cavalo. O Jogador 1 joga com o cavalo branco e o Jogador 2 joga com o cavalo preto.

O jogo começa com a colocação do cavalo branco na casa de maior pontuação da 1ª linha (A1-J1 do tabuleiro).

Após a primeira jogada (colocar o cavalo branco) segue-se a jogada do adversário, com colocação do cavalo preto na casa de maior pontuação da 10ª linha (A10-J10) do tabuleiro.

2. Arquitetura do Sistema

Este programa é composto por 4 ficheiros:

- **interact.lisp**
 - Carrega os outros ficheiros de código, escreve e lê ficheiros, e trata da interação com o utilizador.
- **algoritmo.lisp**
 - Contem a implementação do algoritmo a utilizado neste projeto.
- **jogo.lisp**
 - Contém o código relacionado com o problema
- **log.dat**
 - Contem informações estatísticas sobre uma jogada computador.

Como referido anteriormente o ficheiro **jogo.lisp** contém a implementação do código relacionado com o problema, como por exemplo os operadores, que vão ser descritos mais adiante. Esta informação é essencial para o ficheiro **algoritmo.lisp**, pois o algoritmo vai usar a implementação de funções como sucessores e da estrutura de dados de um nó que também consta no mesmo ficheiro.

O ficheiro **interact.lisp** apresenta uma maneira mais *user-friendly* do utilizador poder utilizar de forma simultânea o algoritmo implementado no **algoritmo.lisp**.

Resumindo: - **jogo.lisp** -> **algoritmo.lisp** -> **interact.lisp**

3. Entidades e a sua implementação

Para este problema o elemento principal é um tabuleiro.

O tabuleiro tem as dimensões 10x10 em que os valores de cada casa são entre 00 e 99, sem repetição. O cavalo é representado por 'T' no tabuleiro e as casas que o cavalo não pode visitar são marcadas como 'NIL'. A imagem seguinte representa uma ilustração do mesmo:

```
((94 25 54 89 21 8 36 14 41 96)
 (78 47 56 23 5 49 13 12 26 60)
 (0 27 17 83 34 93 74 52 45 80)
 (69 9 77 95 55 39 91 73 57 30)
 (24 15 22 86 1 11 68 79 76 72)
 (81 48 32 2 64 16 50 37 29 71)
 (99 51 6 18 53 28 7 63 10 88)
 (59 42 46 85 90 75 87 43 20 31)
 (3 61 58 44 65 82 19 4 35 62)
 (33 70 84 40 66 38 92 67 98 97))

((NIL 25 54 89 21 8 36 14 41 96)
 (78 47 56 23 5 49 13 12 26 60)
 (0 T 17 83 34 93 74 52 45 80)
 (69 9 77 95 55 39 91 73 57 30)
 (24 15 22 86 1 11 68 79 76 NIL)
 (81 48 32 2 64 16 50 37 29 71)
 (99 51 6 18 53 28 7 63 10 88)
 (59 42 46 85 90 75 87 43 20 31)
 (3 61 58 44 65 82 19 4 35 62)
 (33 70 84 40 66 38 92 67 98 97))
```

No segundo tabuleiro, o cavalo movimentou-se para a casa de valor 27. A casa inicial ficou marcada como NIL e a casa de valor 72 também ficou marcada como NIL, devido a uma das regras específicas deste jogo que vão ser abordadas de seguida.

3.1. Regras do Jogo

3.1.1. Regra do Número Simétrico

Se um número tiver dois dígitos diferentes, o seu simétrico é o seu inverso. Considere-se o exemplo anterior:

- O cavalo movimentou-se para a casa de valor 27
- O seu simétrico é o inverso de 27, neste caso 72.

Se o cavalo move-se para uma casa da qual o seu simétrico ainda não tiver sido removido, então do momento em que o cavalo se desloca para essa casa, o número simétrico é removido, ficando então a estar inacessível para o resto do jogo.

Para verificar qual o simétrico de um número e remove-lo do tabuleiro desenvolveram-se duas funções:

```
(defun numero-simetrico-p (numero)
  (and (numberp numero)
```

```
(+ (* (mod numero 10) 10) (floor numero 10))))

(defun remover-simetrico (simetrico tabuleiro)
  (mapcar (lambda (linha) (mapcar (lambda (celula) (if (eql celula simetrico) nil
celula)) linha)) tabuleiro))
```

Em que:

- `numero-simetrico-p` devolve o simétrico de um número
- `remover-simetrico` remove o simétrico caso exista no tabuleiro.

3.1.2. Regra do Número Duplo

Um número duplo no contexto deste jogo é um número que tem dois dígitos repetidos, como por exemplo 11, 22, 33... Se um cavalo movimentar-se para uma casa que tenha valor de um número duplo, por defeito o maior duplo da casa também é removido.

```
((NIL 25 54 89 21 8 36 14 41 96)
 (78 47 T 23 5 49 13 12 26 60)
 (0 27 17 83 34 93 74 52 45 80)
 (69 9 95 77 55 39 91 73 57 30)
 (24 15 22 86 1 11 68 79 76 72)
 (81 48 32 2 64 16 50 37 29 71)
 (99 51 6 18 53 28 7 63 10 88)
 (59 42 46 85 90 75 87 43 20 31)
 (3 61 58 44 65 82 19 4 35 62)
 (33 70 84 40 66 38 92 67 98 97))

((NIL 25 54 89 21 8 36 14 41 96)
 (78 47 NIL 23 5 49 13 12 26 60)
 (0 27 17 83 34 93 74 52 45 80)
 (69 9 95 T 55 39 91 73 57 30)
 (24 15 22 86 1 11 68 79 76 72)
 (81 48 32 2 64 16 50 37 29 71)
 (NIL 51 6 18 53 28 7 63 10 88)
 (59 42 46 85 90 75 87 43 20 31)
 (3 61 58 44 65 82 19 4 35 62)
 (33 70 84 40 66 38 92 67 98 97))
```

O cavalo movimentou-se para a casa de valor 77 e então o maior duplo presente no tabuleiro, neste caso sendo o 99, também foi removido do tabuleiro. Para esta validação foram implementadas as seguintes funções:

```
(defun numero-duplo-p (numero)
  (and (numberp numero)
    (not (= numero 0))
    (= (mod numero 10) (floor numero 10))))
```

```
(defun lista-duplos (tabuleiro)
  (cond
    ((null tabuleiro) nil)
    ((numero-duplo-p (car tabuleiro))
     (cons (car tabuleiro) (lista-duplos (cdr tabuleiro))))
    (t (lista-duplos (cdr tabuleiro)))))

(defun remover-duplo (maior-duplo tabuleiro)
  (mapcar (lambda (linha) (mapcar (lambda (celula) (if (eql celula maior-duplo) nil
                                                         celula)) linha)) tabuleiro))
```

Em que:

- **numero-duplo-p** Verifica se um dado número é um número duplo
- **lista-duplos** Devolve a lista de números duplos presentes no tabuleiro.
- **remover-duplos** Remove o duplo no tabuleiro (de notar que esta função precisa da função já existente no Common Lisp **max** para corretamente remover o maior duplo presente no tabuleiro).

De referir como regras adicionais, que o cavalo não pode referir para uma casa que esteja fora dos limites do tabuleiro e não pode visitar uma casa que esteja inacessível (marcada como **NIL**). Para tal implementou-se a função **movimento-valido** seguinte função que lida com todos os casos

```
(defun movimento-valido (linha coluna tabuleiro jogador)
  (cond
    ((or (null linha) (null coluna) (null tabuleiro)) nil)
    ((or (> linha 9) (< linha 0) (> coluna 9) (< coluna 0)) nil)
    ((null (celula linha coluna tabuleiro)) nil)
    ((and (numberp (celula linha coluna tabuleiro))
          (>= (celula linha coluna tabuleiro) 0)
          (<= (celula linha coluna tabuleiro) 99)) t)
    (t nil)))
```

3.2. Representação do Estado

Para representar um jogo de 2 jogadores, criaram-se duas variáveis globais para representar a peça de cada jogador (o jogador 1 representado pela peça de valor -1 e o jogador 2 representado pela peça de valor -2).

```
(defvar *jogador1* -1)
(defvar *jogador2* -2)
```

3.2.1. Nó

A estrutura de dados escolhida foi um nó em que:

- O **primeiro** elemento corresponde ao **estado** do nó, neste caso específico representa o tabuleiro;
- O **segundo** elemento representa o **nó pai**;

- O **terceiro** elemento os **pontos acumulados do jogador 1**;
- O **quarto** elemento os **pontos acumulados do jogador 2** do nó;
- O **quinto** elemento corresponde à **profundidade** do nó;

```
(defun criar-no (tabuleiro pai pontos-jogador1 pontos-jogador2 profundidade)
  "Cria um nó com o estado do tabuleiro, o nó pai, os pontos dos jogadores 1 e 2,
  e a profundidade."
  (list tabuleiro pai pontos-jogador1 pontos-jogador2 profundidade))

(defun estado-do-no (no)
  "Retorna o estado atual (tabuleiro) do nó"
  (first no))

(defun pai-do-no (no)
  "Retorna o nó-pai do nó"
  (second no))

(defun pontos-jogador1 (no)
  "Retorna os pontos acumulados do jogador 1"
  (third no))

(defun pontos-jogador2 (no)
  "Retorna os pontos acumulados do jogador 2"
  (fourth no))

(defun profundidade-do-no (no)
  "Retorna a profundidade do nó."
  (fifth no))
```

3.2.2. Operadores

Os operadores representam os movimentos possíveis num determinado estado. Para o Problema do Cavalo o máximo de movimentos possíveis serão 8, desde que essas casas não tenham sido ainda visitadas ou removidas pela regra dos simétricos ou duplos.

De forma a evitar código duplicado e atendendo que os operadores tinham o mesmo comportamento e apenas diferiam no movimento, criou-se uma função **movimentar-cavalo** que recebe o tabuleiro, a linha e a coluna para o qual o cavalo se move.

```
(defun movimentar-cavalo (tabuleiro deslocamento-linha deslocamento-coluna
  jogador)
  "Move o cavalo no tabuleiro com base no deslocamento fornecido, se válido."
  (let ((cavalo-posicao (posicao-cavalo tabuleiro jogador)))
    (if cavalo-posicao
      (let ((nova-linha (+ (first cavalo-posicao) deslocamento-linha))
            (nova-coluna (+ (second cavalo-posicao) deslocamento-coluna)))
        (if (movimento-valido nova-linha nova-coluna tabuleiro jogador)
```

```

        (let ((casa-destino (celula nova-linha nova-coluna tabuleiro)))
          (cond
            ((and casa-destino (numero-duplo-p casa-destino))
              (let* ((duplos (lista-duplos (concatena tabuleiro)))
                     (maior-duplo (when duplos (apply #'max duplos))))
                (values (remover-duplo maior-duplo (substituir nova-linha
novo-coluna (substituir (first cavalo-posicao) (second cavalo-posicao) tabuleiro
nil) jogador))
                        (if (numberp casa-destino) casa-destino 0)
                        nova-linha nova-coluna)))
            ((and casa-destino (numero-simetrico-p casa-destino))
              (let ((simetrico (numero-simetrico-p casa-destino)))
                (values (remover-simetrico simetrico (substituir nova-linha
novo-coluna (substituir (first cavalo-posicao) (second cavalo-posicao) tabuleiro
nil) jogador))
                        (if (numberp casa-destino) casa-destino 0)
                        nova-linha nova-coluna)))
            (t (values (substituir nova-linha nova-coluna (substituir (first
cavalo-posicao) (second cavalo-posicao) tabuleiro nil) jogador)
                      (if (numberp casa-destino) casa-destino 0)
                      nova-linha nova-coluna))))
          nil))))

```

De forma que:

```

(defun operador-1 (tabuleiro)
  "Move o cavalo 2 linhas para baixo e 1 coluna para a direita."
  (movimentar-cavalo tabuleiro 2 1))

(defun operador-2 (tabuleiro)
  "Move o cavalo 1 linha para baixo e 2 colunas para a direita."
  (movimentar-cavalo tabuleiro 1 2))

(defun operador-3 (tabuleiro)
  "Move o cavalo 1 linha para cima e 2 colunas para a direita"
  (movimentar-cavalo tabuleiro -1 2))

(defun operador-4 (tabuleiro)
  "Move o cavalo 2 linhas para cima e 1 coluna para a direita"
  (movimentar-cavalo tabuleiro -2 1))

(defun operador-5 (tabuleiro)
  "Move o cavalo 2 linhas para cima e 1 coluna para a esquerda"
  (movimentar-cavalo tabuleiro -2 -1))

(defun operador-6 (tabuleiro)
  "Move o cavalo 1 linha para cima e 2 colunas para a esquerda"
  (movimentar-cavalo tabuleiro -1 -2))

(defun operador-7 (tabuleiro)
  "Move o cavalo 1 linha para baixo e 2 colunas para a esquerda"
  (movimentar-cavalo tabuleiro 1 -2))

```

```
(defun operador-8 (tabuleiro)
  "Move o cavalo 2 linhas para baixo e 1 coluna para a esquerda"
  (movimentar-cavalo tabuleiro 2 -1))
```

```
CL-USER 2 > (pprint(operador-1(tabuleiro-teste)))
```

```
((NIL 25 54 89 21 8 36 14 41 96)
 (78 47 56 23 5 49 13 12 26 60)
 (0 T 17 83 34 93 74 52 45 80)
 (69 9 77 95 55 39 91 73 57 30)
 (24 15 22 86 1 11 68 79 76 NIL)
 (81 48 32 2 64 16 50 37 29 71)
 (99 51 6 18 53 28 7 63 10 88)
 (59 42 46 85 90 75 87 43 20 31)
 (3 61 58 44 65 82 19 4 35 62)
 (33 70 84 40 66 38 92 67 98 97))
```

Ao termos os operadores pode-se então implementar as funções de gerar um novo sucessor e os sucessores respetivamente:

```
(defun novo-sucessor (no operador jogador)
  (multiple-value-bind (novo-tabuleiro valor jogada-linha jogada-coluna)
    (funcall operador (estado-do-no no) jogador)
    (when novo-tabuleiro
      (let ((nova-profundidade (1+ (profundidade-do-no no))))
        (values
         (cond
          ((= jogador *jogador1*)
           (criar-no novo-tabuleiro (pai-do-no no) (+ (pontos-jogador1 no) valor)
            (pontos-jogador2 no) nova-profundidade))
          (t
           (criar-no novo-tabuleiro (pai-do-no no) (pontos-jogador1 no) (+
            (pontos-jogador2 no) valor) nova-profundidade)))
         jogada-linha jogada-coluna))))))

(defun sucessores (no lista-op jogador)
  (remove-if #'null
    (mapcar (lambda (operador)
      (multiple-value-bind (sucessor jogada-linha jogada-coluna)
        (novo-sucessor no operador jogador)
        (when sucessor
          (list sucessor jogada-linha jogada-coluna jogador))))
    lista-op)))
```

4. Algoritmo Alfabeta

Para desenvolver o algoritmo alfabeta foi necessário previamente criar uma função para avaliar um nó. Os critérios de escolha para avaliação de um nó foram:

- Diferença de pontos
- Número de movimentos possíveis do jogador
- Número de movimentos possíveis do jogador oposto.

O objetivo seria que a função 'avaliar-no' fosse capaz de calcular o valor de um nó que atendesse às condições mencionadas, sendo mais favorável quanto maior para o jogador MAX e menor para o jogador MIN.

```
(defun avaliar-no (no tipo-jogador)
  (let* ((tabuleiro (estado-do-no no))
        (pontos-jogador1 (third no))
        (pontos-jogador2 (fourth no))
        (diferenca-pontos (- pontos-jogador1 pontos-jogador2))
        (mobilidade-jogador1 (length (gerar-movimentos-possiveis tabuleiro
*jogador1* (operadores))))
        (mobilidade-jogador2 (length (gerar-movimentos-possiveis tabuleiro
*jogador2* (operadores)))))

    (cond ((eql tipo-jogador *jogador1*)
            (+ diferenca-pontos
              (* 0.5 mobilidade-jogador1)
              (* -0.5 mobilidade-jogador2)))
          (t
            (- diferenca-pontos
              (* 0.5 mobilidade-jogador2)
              (* -0.5 mobilidade-jogador1))))))
```

Tendo então criado uma função para avaliar um nó, desenvolveu-se então o algoritmo alfabeta que devolve o melhor valor e jogada para o jogador em questão:

```
(defun alfabeta (no profundidade alfa beta jogador &optional (chamada-final t))
  (let ((inicio-tempo (get-internal-real-time)))
    (when chamada-final
      (setf *cortes-alfa* 0)
      (setf *cortes-beta* 0)
      (setf *nos-analisados* 0))

    (cond ((zerop profundidade)
            (incf *nos-analisados*)
            (values (avaliar-no no jogador) nil))
          (t (multiple-value-bind (valor jogada)
                (alfabeta-aux (sucessores no (operadores) jogador) profundidade
                              (if (= jogador *jogador1*) most-negative-fixnum
                                  most-positive-fixnum)
                              nil alfa beta jogador)
                  (when chamada-final
                    (let ((tempo-final (* (/ (- (get-internal-real-time) inicio-
```

```

tempo) internal-time-units-per-second 1000)))
      (let ((estatistica (format nil "~% Jogador: ~A, Tempo Gasto: ~A
ms, Nós analisados: ~A, Profundidade: ~A, Valor: ~A, Cortes Alfa: ~A, Cortes Beta:
~A"
                                jogador tempo-final *nos-analisados*
profundidade valor *cortes-alfa* *cortes-beta*)))
        ;; Escreve no log
        (escrever-ficheiro estatistica)
        ;; Imprime na tela
        (format t "~A~%" estatistica))))
    (values valor jogada))))))

(defun alfabet-aux (sucessores profundidade melhor-valor melhor-jogada alfa beta
jogador)
  (cond
    ((null sucessores)
     (values melhor-valor melhor-jogada))
    (t
     (let* ((sucessor-info (first sucessores))
            (sucessor (first sucessor-info))
            (jogada-linha (second sucessor-info))
            (jogada-coluna (third sucessor-info))
            (multiple-value-bind (valor _)
              (alfabet-aux sucessor (- profundidade 1) alfa beta (trocar-jogador jogador)
nil)
              (incf *nos-analisados*)
              (let ((novo-melhor-valor (cond ((= jogador *jogador1*) (max valor melhor-
valor))
                                             (t (min valor melhor-valor))))
                (nova-melhor-jogada (cond ((or (and (= jogador *jogador1*) (> valor
melhor-valor))
                                             (and (not (= jogador *jogador1*)) (<
valor melhor-valor)))
                (list jogada-linha jogada-coluna))
                (t melhor-jogada)))
              (novo-alfa (cond ((= jogador *jogador1*) (max alfa valor))
                              (t alfa)))
              (novo-beta (cond ((not (= jogador *jogador1*)) (min beta valor))
                              (t beta))))
            (cond
              ((or (and (= jogador *jogador1*) (>= novo-alfa beta))
                  (and (not (= jogador *jogador1*)) (<= novo-beta alfa)))
               (if (= jogador *jogador1*)
                   (incf *cortes-beta*)
                   (incf *cortes-alfa*))
               (return-from alfabet-aux (values novo-melhor-valor nova-melhor-
jogada))))
              (t (alfabet-aux (rest sucessores) profundidade novo-melhor-valor nova-
melhor-jogada novo-alfa novo-beta jogador)))))))))

```

5. Análise da eficiência do algoritmo

Para analisar a eficiência do algoritmo jogou-se uma partida Humano vs Computador na qual:

- Profundidade máxima do alfabeto: 4
- Tempo limite de jogada: 3000 milissegundos
- O jogador MAX é o jogador 1 (Humano) e o jogador MIN é o computador (jogador 2)

```
CL-USER 4 > (iniciar-jogo)
```

```
Escolha o modo de jogo:
```

```
1 - Humano vs Computador
```

```
2 - Computador vs Computador
```

```
3 - Sair
```

```
Escolha uma opção: 1
```

```
Defina o tempo limite para jogadas do computador (1000 a 5000 milissegundos): 3000
```

```
Escolha a profundidade para o algoritmo alfabeto (1 a 5): 4
```

```
Quem deve começar a partida?
```

```
1 - Humano
```

```
2 - Computador
```

```
Escolha uma opção: 2
```

```
Vez do jogador: -2
```

```
Jogador: -2, Tempo Gasto: 265 ms, Nós analisados: 131, Profundidade: 4, Valor:  
35.0, Cortes Alfa: 4, Cortes Beta: 21
```

```
Jogador -2 move cavalo para (7, 5).
```

```
Pontos Ganhos: 69
```

```
((11 70 32 50 88 91 55 8 -1 16)  
(0 28 25 76 98 15 48 95 39 12)  
(36 6 3 29 45 83 58 10 20 82)  
(75 33 93 86 97 26 94 NIL 65 1)  
(37 73 92 59 21 67 60 22 9 71)  
(72 31 4 66 62 87 53 61 14 24)  
(44 19 85 90 17 64 46 NIL 47 38)  
(63 34 80 56 30 -2 79 35 7 43)  
(23 74 5 84 89 2 42 77 13 51)  
(40 49 54 57 NIL 41 52 27 78 68))
```

```
Pontos do Jogador 1: 99
```

```
Pontos do Jogador 2: 150
```

```
Tabuleiro atual:
```

```
((11 70 32 50 88 91 55 8 -1 16)  
(0 28 25 76 98 15 48 95 39 12)  
(36 6 3 29 45 83 58 10 20 82)  
(75 33 93 86 97 26 94 NIL 65 1)  
(37 73 92 59 21 67 60 22 9 71)  
(72 31 4 66 62 87 53 61 14 24)  
(44 19 85 90 17 64 46 NIL 47 38)
```

```
(63 34 80 56 30 -2 79 35 7 43)
(23 74 5 84 89 2 42 77 13 51)
(40 49 54 57 NIL 41 52 27 78 68))
```

Movimentos possíveis do Jogador -1: ((2 9) (1 6) (2 7))

Jogador -1, por favor insira a nova linha para o cavalo: 2

Agora insira a nova coluna para o cavalo: 9

Pontos Ganhos: 82

```
((11 70 32 50 88 91 55 8 NIL 16)
(0 NIL 25 76 98 15 48 95 39 12)
(36 6 3 29 45 83 58 10 20 -1)
(75 33 93 86 97 26 94 NIL 65 1)
(37 73 92 59 21 67 60 22 9 71)
(72 31 4 66 62 87 53 61 14 24)
(44 19 85 90 17 64 46 NIL 47 38)
(63 34 80 56 30 -2 79 35 7 43)
(23 74 5 84 89 2 42 77 13 51)
(40 49 54 57 NIL 41 52 27 78 68))
```

Pontos do Jogador 1: 181

Pontos do Jogador 2: 150

Vez do jogador: -2

Jogador: -2, Tempo Gasto: 550 ms, Nós analisados: 237, Profundidade: 4, Valor: 38.0, Cortes Alfa: 3, Cortes Beta: 25

Jogador -2 move cavalo para (6, 3).

Pontos Ganhos: 90

```
((11 70 32 50 88 91 55 8 NIL 16)
(0 NIL 25 76 98 15 48 95 39 12)
(36 6 3 29 45 83 58 10 20 -1)
(75 33 93 86 97 26 94 NIL 65 1)
(37 73 92 59 21 67 60 22 NIL 71)
(72 31 4 66 62 87 53 61 14 24)
(44 19 85 -2 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL 79 35 7 43)
(23 74 5 84 89 2 42 77 13 51)
(40 49 54 57 NIL 41 52 27 78 68))
```

Pontos do Jogador 1: 181

Pontos do Jogador 2: 240

Tabuleiro atual:

```
((11 70 32 50 88 91 55 8 NIL 16)
(0 NIL 25 76 98 15 48 95 39 12)
(36 6 3 29 45 83 58 10 20 -1)
```

(75 33 93 86 97 26 94 NIL 65 1)
(37 73 92 59 21 67 60 22 NIL 71)
(72 31 4 66 62 87 53 61 14 24)
(44 19 85 -2 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL 79 35 7 43)
(23 74 5 84 89 2 42 77 13 51)
(40 49 54 57 NIL 41 52 27 78 68))

Movimentos possíveis do Jogador -1: ((1 7))

Jogador -1, por favor insira a nova linha para o cavalo: 1

Agora insira a nova coluna para o cavalo: 7

Pontos Ganhos: 95

((11 70 32 50 88 91 55 8 NIL 16)
(0 NIL 25 76 98 15 48 -1 39 12)
(36 6 3 29 45 83 58 10 20 NIL)
(75 33 93 86 97 26 94 NIL 65 1)
(37 73 92 NIL 21 67 60 22 NIL 71)
(72 31 4 66 62 87 53 61 14 24)
(44 19 85 -2 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL 79 35 7 43)
(23 74 5 84 89 2 42 77 13 51)
(40 49 54 57 NIL 41 52 27 78 68))

Pontos do Jogador 1: 276

Pontos do Jogador 2: 240

Vez do jogador: -2

Jogador: -2, Tempo Gasto: 1253 ms, Nós analisados: 558, Profundidade: 4, Valor: 21.0, Cortes Alfa: 4, Cortes Beta: 60

Jogador -2 move cavalo para (5, 5).

Pontos Ganhos: 87

((11 70 32 50 88 91 55 8 NIL 16)
(0 NIL 25 76 98 15 48 -1 39 12)
(36 6 3 29 45 83 58 10 20 NIL)
(75 33 93 86 97 26 94 NIL 65 1)
(37 73 92 NIL 21 67 60 22 NIL 71)
(72 31 4 66 62 -2 53 61 14 24)
(44 19 85 NIL 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL 79 35 7 43)
(23 74 5 84 89 2 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))

Pontos do Jogador 1: 276

Pontos do Jogador 2: 327

Tabuleiro atual:

```
((11 70 32 50 88 91 55 8 NIL 16)
(0 NIL 25 76 98 15 48 -1 39 12)
(36 6 3 29 45 83 58 10 20 NIL)
(75 33 93 86 97 26 94 NIL 65 1)
(37 73 92 NIL 21 67 60 22 NIL 71)
(72 31 4 66 62 -2 53 61 14 24)
(44 19 85 NIL 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL 79 35 7 43)
(23 74 5 84 89 2 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))
Movimentos possíveis do Jogador -1: ((3 8) (0 9) (0 5) (2 5) (3 6))
```

Jogador -1, por favor insira a nova linha para o cavalo: 0

Agora insira a nova coluna para o cavalo: 9

Pontos Ganhos: 16

```
((11 70 32 50 88 91 55 8 NIL -1)
(0 NIL 25 76 98 15 48 NIL 39 12)
(36 6 3 29 45 83 58 10 20 NIL)
(75 33 93 86 97 26 94 NIL 65 1)
(37 73 92 NIL 21 67 60 22 NIL 71)
(72 31 4 66 62 -2 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL 79 35 7 43)
(23 74 5 84 89 2 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))
```

Pontos do Jogador 1: 292

Pontos do Jogador 2: 327

Vez do jogador: -2

Jogador: -2, Tempo Gasto: 314 ms, Nós analisados: 159, Profundidade: 4, Valor: -111.0, Cortes Alfa: 0, Cortes Beta: 27

Jogador -2 move cavalo para (3, 4).

Pontos Ganhos: 97

```
((11 70 32 50 88 91 55 8 NIL -1)
(0 NIL 25 76 98 15 48 NIL 39 12)
(36 6 3 29 45 83 58 10 20 NIL)
(75 33 93 86 -2 26 94 NIL 65 1)
(37 73 92 NIL 21 67 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 84 89 2 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))
```

Pontos do Jogador 1: 292

Pontos do Jogador 2: 424

Tabuleiro atual:

```
((11 70 32 50 88 91 55 8 NIL -1)
(0 NIL 25 76 98 15 48 NIL 39 12)
(36 6 3 29 45 83 58 10 20 NIL)
(75 33 93 86 -2 26 94 NIL 65 1)
(37 73 92 NIL 21 67 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 84 89 2 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))
```

Movimentos possíveis do Jogador -1: ((2 8))

Jogador -1, por favor insira a nova linha para o cavalo: 2

Agora insira a nova coluna para o cavalo: 8

Pontos Ganhos: 20

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL 25 76 98 15 48 NIL 39 12)
(36 6 3 29 45 83 58 10 -1 NIL)
(75 33 93 86 -2 26 94 NIL 65 1)
(37 73 92 NIL 21 67 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 84 89 NIL 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))
```

Pontos do Jogador 1: 312

Pontos do Jogador 2: 424

Vez do jogador: -2

Jogador: -2, Tempo Gasto: 917 ms, Nós analisados: 515, Profundidade: 4, Valor: -142.5, Cortes Alfa: 9, Cortes Beta: 71

Jogador -2 move cavalo para (1, 3).

Pontos Ganhos: 76

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL 25 -2 98 15 48 NIL 39 12)
(36 6 3 29 45 83 58 10 -1 NIL)
(75 33 93 86 NIL 26 94 NIL 65 1)
(37 73 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL NIL 35 7 43)
```

```
(23 74 5 84 89 NIL 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))
```

Pontos do Jogador 1: 312

Pontos do Jogador 2: 500

Tabuleiro atual:

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL 25 -2 98 15 48 NIL 39 12)
(36 6 3 29 45 83 58 10 -1 NIL)
(75 33 93 86 NIL 26 94 NIL 65 1)
(37 73 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 84 89 NIL 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))
```

Movimentos possíveis do Jogador -1: ((4 9) (0 7) (1 6) (3 6) (4 7))

Jogador -1, por favor insira a nova linha para o cavalo: 1

Agora insira a nova coluna para o cavalo: 6

Pontos Ganhos: 48

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL 25 -2 98 15 -1 NIL 39 12)
(36 6 3 29 45 83 58 10 NIL NIL)
(75 33 93 86 NIL 26 94 NIL 65 1)
(37 73 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 38)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))
```

Pontos do Jogador 1: 360

Pontos do Jogador 2: 500

Vez do jogador: -2

Jogador: -2, Tempo Gasto: 224 ms, Nós analisados: 143, Profundidade: 4, Valor: -171.5, Cortes Alfa: 1, Cortes Beta: 25

Jogador -2 move cavalo para (2, 5).

Pontos Ganhos: 83

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL 25 NIL 98 15 -1 NIL 39 12)
(36 6 3 29 45 -2 58 10 NIL NIL)
(75 33 93 86 NIL 26 94 NIL 65 1)
```



```
(37 73 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))
```

Pontos do Jogador 1: 360

Pontos do Jogador 2: 583

Tabuleiro atual:

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL 25 NIL 98 15 -1 NIL 39 12)
(36 6 3 29 45 -2 58 10 NIL NIL)
(75 33 93 86 NIL 26 94 NIL 65 1)
(37 73 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 54 57 NIL 41 52 27 NIL 68))
```

Movimentos possíveis do Jogador -1: ((0 4) (2 4) (3 5))

Jogador -1, por favor insira a nova linha para o cavalo: 2

Agora insira a nova coluna para o cavalo: 4

Pontos Ganhos: 45

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL 25 NIL 98 15 NIL NIL 39 12)
(36 6 3 29 -1 -2 58 10 NIL NIL)
(75 33 93 86 NIL 26 94 NIL 65 1)
(37 73 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 52 27 NIL 68))
```

Pontos do Jogador 1: 405

Pontos do Jogador 2: 583

Vez do jogador: -2

Jogador: -2, Tempo Gasto: 505 ms, Nós analisados: 305, Profundidade: 4, Valor: -203.0, Cortes Alfa: 6, Cortes Beta: 33

Jogador -2 move cavalo para (3, 3).

Pontos Ganhos: 86

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL 25 NIL 98 15 NIL NIL 39 12)
(36 6 3 29 -1 NIL 58 10 NIL NIL)
(75 33 93 -2 NIL 26 94 NIL 65 1)
(37 73 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 52 27 NIL NIL))
```

Pontos do Jogador 1: 405
Pontos do Jogador 2: 669

Tabuleiro atual:

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL 25 NIL 98 15 NIL NIL 39 12)
(36 6 3 29 -1 NIL 58 10 NIL NIL)
(75 33 93 -2 NIL 26 94 NIL 65 1)
(37 73 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 52 27 NIL NIL))
```

Movimentos possíveis do Jogador -1: ((3 6) (0 5) (0 3) (1 2) (3 2))

Jogador -1, por favor insira a nova linha para o cavalo: 1

Agora insira a nova coluna para o cavalo: 2

Pontos Ganhos: 25

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL -1 NIL 98 15 NIL NIL 39 12)
(36 6 3 29 NIL NIL 58 10 NIL NIL)
(75 33 93 -2 NIL 26 94 NIL 65 1)
(37 73 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 430
Pontos do Jogador 2: 669

VeZ do jogador: -2

Jogador: -2, Tempo Gasto: 371 ms, Nós analisados: 219, Profundidade: 4, Valor: -268.5, Cortes Alfa: 9, Cortes Beta: 31

Jogador -2 move cavalo para (4, 1).

Pontos Ganhos: 73

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL -1 NIL 98 15 NIL NIL 39 12)
(36 6 3 29 NIL NIL 58 10 NIL NIL)
(75 33 93 NIL NIL 26 94 NIL 65 1)
(NIL -2 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 430

Pontos do Jogador 2: 742

Tabuleiro atual:

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL -1 NIL 98 15 NIL NIL 39 12)
(36 6 3 29 NIL NIL 58 10 NIL NIL)
(75 33 93 NIL NIL 26 94 NIL 65 1)
(NIL -2 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(63 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 NIL 27 NIL NIL))
```

Movimentos possíveis do Jogador -1: ((0 4) (0 0) (2 0) (3 1))

Jogador -1, por favor insira a nova linha para o cavalo: 2

Agora insira a nova coluna para o cavalo: 0

Pontos Ganhos: 36

```
((11 70 32 50 88 91 55 8 NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL 39 12)
(-1 6 3 29 NIL NIL 58 10 NIL NIL)
(75 33 93 NIL NIL 26 94 NIL 65 1)
(NIL -2 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 66 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(NIL 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 466

Pontos do Jogador 2: 742

Veza do jogador: -2

Jogador: -2, Tempo Gasto: 193 ms, Nós analisados: 80, Profundidade: 4, Valor: -300.5, Cortes Alfa: 2, Cortes Beta: 12

Jogador -2 move cavalo para (5, 3).
Pontos Ganhos: 66

```
((11 70 32 50 NIL 91 55 8 NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL 39 12)
(-1 6 3 29 NIL NIL 58 10 NIL NIL)
(75 33 93 NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 -2 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(NIL 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 466

Pontos do Jogador 2: 808

Tabuleiro atual:

```
((11 70 32 50 NIL 91 55 8 NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL 39 12)
(-1 6 3 29 NIL NIL 58 10 NIL NIL)
(75 33 93 NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 -2 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(NIL 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 NIL 27 NIL NIL))
```

Movimentos possíveis do Jogador -1: ((3 2) (0 1))

Jogador -1, por favor insira a nova linha para o cavalo: 3

Agora insira a nova coluna para o cavalo: 2

Pontos Ganhos: 93

```
((11 70 32 50 NIL 91 55 8 NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL 12)
(NIL 6 3 29 NIL NIL 58 10 NIL NIL)
(75 33 -1 NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 -2 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(NIL 34 80 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 559
Pontos do Jogador 2: 808

Vez do jogador: -2

Jogador: -2, Tempo Gasto: 397 ms, Nós analisados: 178, Profundidade: 4, Valor: -303.0, Cortes Alfa: 3, Cortes Beta: 20

Jogador -2 move cavalo para (7, 2).
Pontos Ganhos: 80

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL 12)
(NIL 6 3 29 NIL NIL 58 10 NIL NIL)
(75 33 -1 NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(NIL 34 -2 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 559
Pontos do Jogador 2: 888

Tabuleiro atual:

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL 12)
(NIL 6 3 29 NIL NIL 58 10 NIL NIL)
(75 33 -1 NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL 21 NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(NIL 34 -2 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 NIL 27 NIL NIL))
```

Movimentos possíveis do Jogador -1: ((4 4) (5 1))

Jogador -1, por favor insira a nova linha para o cavalo: 4

Agora insira a nova coluna para o cavalo: 4

Pontos Ganhos: 21

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL 58 10 NIL NIL)
(75 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL -1 NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
```

```
(NIL 34 -2 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL 57 NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 580

Pontos do Jogador 2: 888

Veza do jogador: -2

Jogador: -2, Tempo Gasto: 264 ms, Nós analisados: 107, Profundidade: 4, Valor: -287.5, Cortes Alfa: 4, Cortes Beta: 12

Jogador -2 move cavalo para (9, 3).

Pontos Ganhos: 57

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL 58 10 NIL NIL)
(NIL 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL -1 NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(NIL 34 NIL 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL -2 NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 580

Pontos do Jogador 2: 945

Tabuleiro atual:

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL 58 10 NIL NIL)
(NIL 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL -1 NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 64 46 NIL 47 NIL)
(NIL 34 NIL 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL -2 NIL 41 NIL 27 NIL NIL))
```

Movimentos possíveis do Jogador -1: ((6 5) (5 6) (3 6) (2 3) (5 2))

Jogador -1, por favor insira a nova linha para o cavalo: 6

Agora insira a nova coluna para o cavalo: 5

Pontos Ganhos: 64

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL 58 10 NIL NIL))
```

```
(NIL 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL NIL NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 -1 NIL NIL 47 NIL)
(NIL 34 NIL 56 30 NIL NIL 35 7 43)
(23 74 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL -2 NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 644

Pontos do Jogador 2: 945

Vez do jogador: -2

Jogador: -2, Tempo Gasto: 335 ms, Nós analisados: 151, Profundidade: 4, Valor: -336.0, Cortes Alfa: 3, Cortes Beta: 7

Jogador -2 move cavalo para (8, 1).

Pontos Ganhos: 74

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL 58 10 NIL NIL)
(NIL 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL NIL NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 -1 NIL NIL NIL NIL)
(NIL 34 NIL 56 30 NIL NIL 35 7 43)
(23 -2 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL NIL NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 644

Pontos do Jogador 2: 1019

Tabuleiro atual:

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL 58 10 NIL NIL)
(NIL 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL NIL NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL 53 NIL 14 24)
(44 19 85 NIL 17 -1 NIL NIL NIL NIL)
(NIL 34 NIL 56 30 NIL NIL 35 7 43)
(23 -2 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL NIL NIL 41 NIL 27 NIL NIL))
```

Movimentos possíveis do Jogador -1: ((8 6) (7 7) (4 6) (7 3) (8 4))

Jogador -1, por favor insira a nova linha para o cavalo: 7

Agora insira a nova coluna para o cavalo: 7

Pontos Ganhos: 35

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL 58 10 NIL NIL)
(NIL 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL NIL NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL NIL NIL 14 24)
(44 19 85 NIL 17 NIL NIL NIL NIL NIL)
(NIL 34 NIL 56 30 NIL NIL -1 7 43)
(23 -2 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL NIL NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 679
Pontos do Jogador 2: 1019

Vez do jogador: -2

Jogador: -2, Tempo Gasto: 140 ms, Nós analisados: 55, Profundidade: 4, Valor: -423.0, Cortes Alfa: 2, Cortes Beta: 6

Jogador -2 move cavalo para (6, 2).
Pontos Ganhos: 85

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL NIL 10 NIL NIL)
(NIL 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL NIL NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL NIL NIL 14 24)
(44 19 -2 NIL 17 NIL NIL NIL NIL NIL)
(NIL 34 NIL 56 30 NIL NIL -1 7 43)
(23 NIL 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL NIL NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 679
Pontos do Jogador 2: 1104

Tabuleiro atual:

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 15 NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL NIL 10 NIL NIL)
(NIL 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL NIL NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL NIL NIL 14 24)
(44 19 -2 NIL 17 NIL NIL NIL NIL NIL)
(NIL 34 NIL 56 30 NIL NIL -1 7 43)
(23 NIL 5 NIL 89 NIL 42 77 13 51)
(40 49 NIL NIL NIL 41 NIL 27 NIL NIL))
```

Movimentos possíveis do Jogador -1: ((8 9) (5 8))

Jogador -1, por favor insira a nova linha para o cavalo: 8

Agora insira a nova coluna para o cavalo: 9

Pontos Ganhos: 51

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 NIL NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL NIL 10 NIL NIL)
(NIL 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL NIL NIL 60 22 NIL 71)
(72 31 4 NIL 62 NIL NIL NIL 14 24)
(44 19 -2 NIL 17 NIL NIL NIL NIL NIL)
(NIL 34 NIL 56 30 NIL NIL NIL 7 43)
(23 NIL 5 NIL 89 NIL 42 77 13 -1)
(40 49 NIL NIL NIL 41 NIL 27 NIL NIL))
```

Pontos do Jogador 1: 730

Pontos do Jogador 2: 1104

Vez do jogador: -2

Jogador: -2, Tempo Gasto: 91 ms, Nós analisados: 23, Profundidade: 4, Valor: -1152921504606846976, Cortes Alfa: 1, Cortes Beta: 2

Jogador -2 move cavalo para (5, 0).

Pontos Ganhos: 72

```
((11 70 32 50 NIL 91 55 NIL NIL NIL)
(0 NIL NIL NIL 98 NIL NIL NIL NIL NIL)
(NIL 6 3 29 NIL NIL NIL 10 NIL NIL)
(NIL 33 NIL NIL NIL 26 94 NIL 65 1)
(NIL NIL 92 NIL NIL NIL 60 22 NIL 71)
(-2 31 4 NIL 62 NIL NIL NIL 14 24)
(44 19 NIL NIL 17 NIL NIL NIL NIL NIL)
(NIL 34 NIL 56 30 NIL NIL NIL 7 43)
(23 NIL 5 NIL 89 NIL 42 77 13 -1)
(40 49 NIL NIL NIL 41 NIL NIL NIL NIL))
```

Pontos do Jogador 1: 730

Pontos do Jogador 2: 1176

Fim do jogo!

Jogador 2 venceu!

Pontos finais do Jogador 1: 730

Pontos finais do Jogador 2: 1176

Recorrendo então ao ficheiro log.dat:

Jogador: -2, Tempo Gasto: 265 ms, Nós analisados: 131, Profundidade: 4, Valor: 35.0, Cortes Alfa: 4, Cortes Beta: 21

Jogador: -2, Tempo Gasto: 550 ms, Nós analisados: 237, Profundidade: 4, Valor: 38.0, Cortes Alfa: 3, Cortes Beta: 25

Jogador: -2, Tempo Gasto: 1253 ms, Nós analisados: 558, Profundidade: 4, Valor: 21.0, Cortes Alfa: 4, Cortes Beta: 60

Jogador: -2, Tempo Gasto: 314 ms, Nós analisados: 159, Profundidade: 4, Valor: -111.0, Cortes Alfa: 0, Cortes Beta: 27

Jogador: -2, Tempo Gasto: 917 ms, Nós analisados: 515, Profundidade: 4, Valor: -142.5, Cortes Alfa: 9, Cortes Beta: 71

Jogador: -2, Tempo Gasto: 224 ms, Nós analisados: 143, Profundidade: 4, Valor: -171.5, Cortes Alfa: 1, Cortes Beta: 25

Jogador: -2, Tempo Gasto: 505 ms, Nós analisados: 305, Profundidade: 4, Valor: -203.0, Cortes Alfa: 6, Cortes Beta: 33

Jogador: -2, Tempo Gasto: 371 ms, Nós analisados: 219, Profundidade: 4, Valor: -268.5, Cortes Alfa: 9, Cortes Beta: 31

Jogador: -2, Tempo Gasto: 193 ms, Nós analisados: 80, Profundidade: 4, Valor: -300.5, Cortes Alfa: 2, Cortes Beta: 12

Jogador: -2, Tempo Gasto: 397 ms, Nós analisados: 178, Profundidade: 4, Valor: -303.0, Cortes Alfa: 3, Cortes Beta: 20

Jogador: -2, Tempo Gasto: 264 ms, Nós analisados: 107, Profundidade: 4, Valor: -287.5, Cortes Alfa: 4, Cortes Beta: 12

Jogador: -2, Tempo Gasto: 335 ms, Nós analisados: 151, Profundidade: 4, Valor: -336.0, Cortes Alfa: 3, Cortes Beta: 7

Jogador: -2, Tempo Gasto: 140 ms, Nós analisados: 55, Profundidade: 4, Valor: -423.0, Cortes Alfa: 2, Cortes Beta: 6

Jogador: -2, Tempo Gasto: 91 ms, Nós analisados: 23, Profundidade: 4, Valor: -1152921504606846976, Cortes Alfa: 1, Cortes Beta: 2

- Verifica-se quanto menor o numero de nós analisados menor será o tempo da jogada
- No última jogada do computador o valor devolvido -1152921504606846976, significa que o alfabeto não conseguiu devolver uma jogada válida.

O computador foi o vencedor da partida.

6. Limitações e Requisitos não implementados

- Tem que se sempre aumentar o tamanho da *stack*; se jogar-se vários jogos por vezes mais do que uma vez.

- O algoritmo alfabeta para grandes profundidades torna-se muito moroso, daí o limite máximo ser 5.
- Não foi implementada memoização nem procura quinsciente;
- Não foi possível implementar no movimento válido o facto de que um jogador não pode mover-se para uma casa ameaçada pelo adversário.