

Cage

Relatório intercalar



Universidade do Porto

Faculdade de Engenharia

FEUP

Programação em Lógica

3º ano

Mestrado Integrado em Engenharia Informática e Computação

Turma 4 - Grupo Cage_2

José Peixoto	200603103	ei12134@fe.up.pt
Luís Cruz	201303248	up201303248@fe.up.pt

11 de Outubro de 2016

1 Descrição do jogo

O Cage é um jogo de estratégia em tabuleiro parecido com as damas que foi inventado por Mark Steere em maio de 2010. O autor descreve-o como um jogo para dois jogadores sem qualquer informação oculta. É abstracto e sem factor sorte, sem empates e sem temas. É jogado num tabuleiro de damas 10x10 ou 8x8 e ao contrário do jogo original das damas, no início todo tabuleiro está preenchido com peças promovidas “damas”. “Jogo de aniquilação de alta energia” é a frase escolhida pelo autor para caricaturizar o jogo, uma vez que o movimento para o centro do tabuleiro assegura a aniquilação de pelo menos uma das cores.

1.1 Regras

O Cage é jogado por dois jogadores num tabuleiro de damas com 50 damas vermelhas e 50 damas azuis na versão de tabuleiro 10x10 ou com 32 damas vermelhas e 32 damas azuis na versão de 8x8 tabuleiro. O tabuleiro é inicializado preenchendo todas as casas com damas de cor alternada.

1.1.1 Objetivo

Para vencer é necessário capturar todas as damas inimigas. No final, pode ganhar-se mesmo que se perca a última peça que se está a movimentar (saltar) para capturar todas as damas últimas inimigas.

1.1.2 Movimentos

Existem quatro tipos de movimentos:

1. Restrito
2. Centralizador
3. Adjacente
4. Salto

Durante um turno, um jogador apenas pode utilizar um tipo de movimento.

Restrição 1 Nunca se pode colocar uma dama ortogonalmente (horizontal ou verticalmente) adjacente a uma dama de cor idêntica. Nem de forma transitória durante um turno de vários movimentos.

Restrição 2 Nunca se pode movimentar uma dama que tenha adjacências ortogonais com damas inimigas para uma casa onde tal não aconteça.

Centralizador Este movimento de uma casa, permite à dama deslocar-se na horizontal, vertical ou diagonal para uma casa vazia e que permite que a dama se aproxime do centro do tabuleiro.

Adjacente Uma dama que não tenha adjacências ortogonais com damas inimigas pode-se mover uma casa em qualquer direção que contenha adjacências ortogonais com uma ou mais damas inimigas.

Salto O movimento de salto permite capturar uma dama inimiga, movimentando a dama do jogador de uma casa ortogonalmente adjacente de um lado da dama inimiga para a casa vazia adjacente do lado oposto. É possível capturar uma dama inimiga nas casas periféricas do tabuleiro de uma casa adjacente e do lado oposto da dama inimiga na borda do tabuleiro. O resultado é que quer a dama capturada quer a dama que captura são removidas do tabuleiro neste tipo de captura para fora dos limites do tabuleiro.

2 Representação do estado do jogo

Na representação de jogo usar-se-ão listas de listas que apenas incluem átomos para os diferentes tipos de peças (*red* e *blue*) e a casa vazia (*empty*).

2.1 Representação do estado inicial do tabuleiro:

```
[ [blue, red, blue, red, blue, red, blue, red] ,  
  [red, blue, red, blue, red, blue, red, blue] ,  
  [blue, red, blue, red, blue, red, blue, red] ,  
  [red, blue, red, blue, red, blue, red, blue] ,  
  [blue, red, blue, red, blue, red, blue, red] ,  
  [red, blue, red, blue, red, blue, red, blue] ,  
  [blue, red, blue, red, blue, red, blue, red] ,  
  [red, blue, red, blue, red, blue, red, blue]  
] ).
```

b	r	b	r	b	r	b	r
r	b	r	b	r	b	r	b
b	r	b	r	b	r	b	r
r	b	r	b	r	b	r	b
b	r	b	r	b	r	b	r
r	b	r	b	r	b	r	b
b	r	b	r	b	r	b	r
r	b	r	b	r	b	r	b

Figura 1: Estado inicial do jogo (linha de comandos)

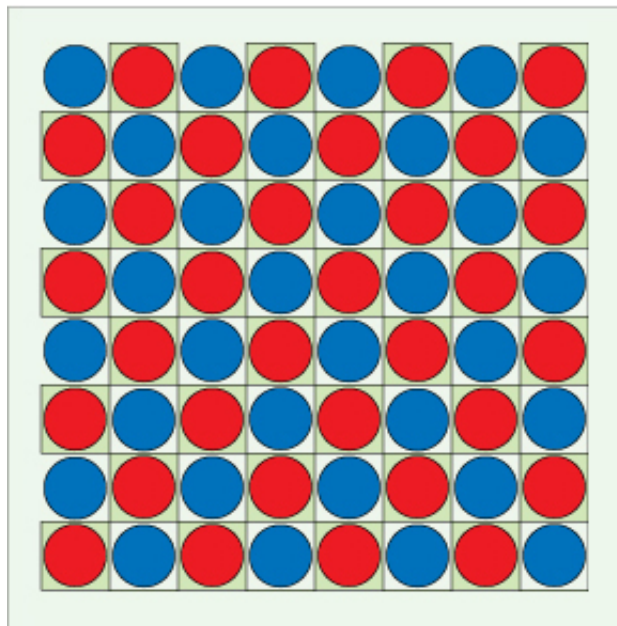


Figura 2: Estado inicial do jogo

2.2 Representação de um estado intermédio do tabuleiro:

```
[ [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, red, blue, empty, empty, empty, empty, empty],
  [red, empty, empty, empty, empty, empty, empty, empty],
  [empty, blue, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty]
].
```

	r	b					
r							
	b						

Figura 3: Estado intermédio de um jogo (linha de comandos)

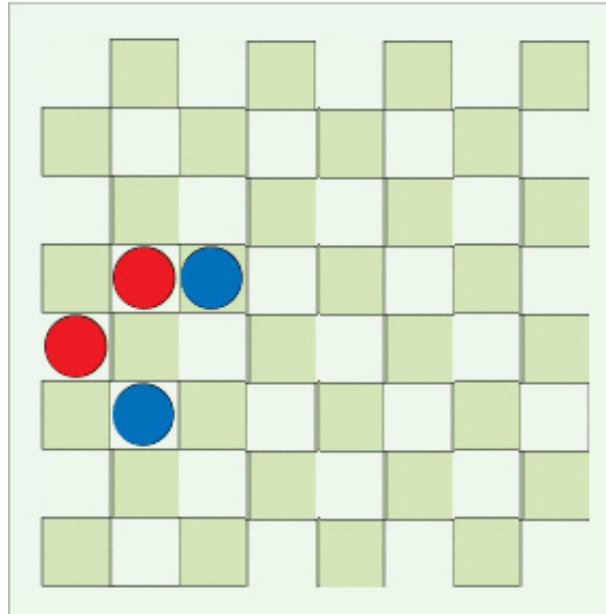


Figura 4: Estado intermédio de um jogo

2.3 Representação de um possível estado final do tabuleiro:

```
[ [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty],
  [empty, empty, empty, empty, empty, empty, empty, empty]
].
```

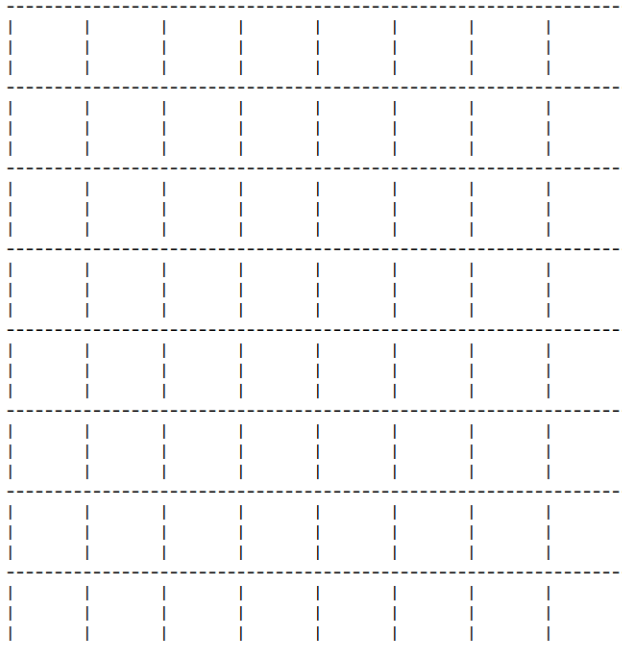


Figura 5: Estado final de um jogo CLI (linha de comandos)

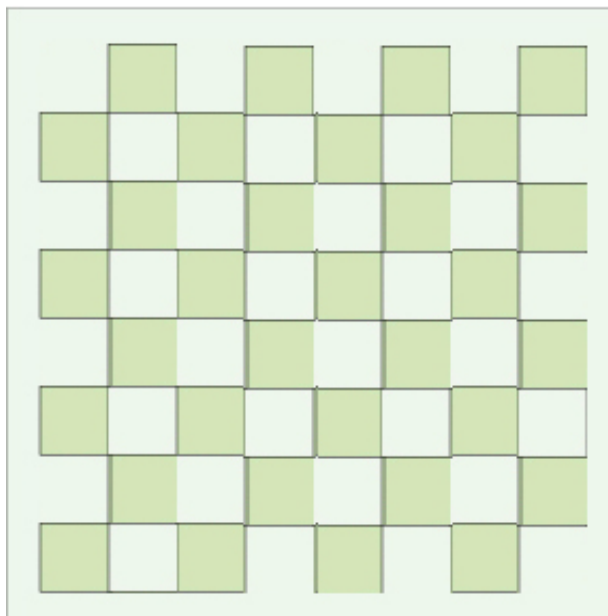


Figura 6: Estado final de um jogo

3 Visualização do tabuleiro em modo de texto

```
% container
board([[blue,red,blue,red,blue,red,blue,red],
       [red,blue,red,blue,red,blue,red,blue],
       [blue,red,blue,red,blue,red,blue,red],
       [red,blue,red,blue,red,blue,red,blue],
       [blue,red,blue,red,blue,red,blue,red],
       [red,blue,red,blue,red,blue,red,blue],
       [blue,red,blue,red,blue,red,blue,red],
       [red,blue,red,blue,red,blue,red,blue]
      ]).

% display board
display_board([H|T]) :-
    % how to display 1st line border?
    write(' ----- '), nl,
    display_empty_line([]),
    display_line(H), nl,
    display_empty_line([]),
    display_board(T).

display_board([]):-
    write(' ----- '), nl.

% display line
display_line([H|T]) :-
    symbol(H,S),
    write(' | '), write(S),
    display_line(T).
display_line([]) :-
    write(' | ').

%display empty line
display_empty_line([]):-
    write(' '), write('|'),write(' '),
    write('|'), write(' '), write('|'),
    write(' '), write('|'), write(' '),
    write('|'), write(' '), write('|'), write(' '),
    write('|'), write(' '), write('|'), write(' '),
    write('|'), nl.

% symbols
symbol(red,'r').
```



```
symbol(blue,'b').  
symbol(empty,'').
```

4 Movimentos

De uma forma geral, nos predicados dos movimentos a seguir apresentados é requerida a receção das linhas e colunas iniciais e finais da peça a mover e do tabuleiro.

4.1 Restrições

Restrições aos predicados dos movimentos declarados a seguir. Presume-se que a implementação dos outros predicados relativos aos restantes movimentos requerem a avaliação satisfatória das seguintes restrições.

```
restriction_1(row,col,adj_row,adj_col,board).  
restriction_2(row,col,adj_row,adj_col,board).
```

4.2 Centralizador

Cabeçalho do predicado do movimento que centraliza a dama ao centro do tabuleiro.

```
centering_move(row,col,dest_row,dest_col,board).
```

4.3 Adjacente

Cabeçalho do predicado do movimento para uma casa adjacente.

```
adjoining_move(row,col,dest_row,dest_col,board).
```

4.4 Salto

Cabeçalho do predicado do movimento em salto que permite capturar damas inimigas.

```
jump(row,col,dest_row,dest_col,board).
```

Referências

- [1] Sterling, Leon *The Art of Prolog*, The MIT Press 2nd edition, 2000.
- [2] Abstract games, http://www.marksteeregames.com/MSG_abstract_games.html, 14 10 2016.
- [3] Cage rules, http://www.marksteeregames.com/Cage_rules.html, 14 10 2016.