

Distrify

Relatório Intercalar



Mestrado Integrado em Engenharia Informática
e Computação

Programação em Lógica

Grupo Distrify_2:

André Rodrigues Barros – up201303567

Edgar Duarte Ramos – up201305973

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

11 de Setembro de 2015

Índice

1. O jogo Distrify	3
2. Representação do estado do jogo	4 a 6
3. Visualização do Tabuleiro	7 a 9
4. Jogadas	10

1 - O jogo Distrify

Distrify é um jogo de conexão jogado por dois jogadores, constituído por peças pretas e brancas, num tabuleiro de 9x9 até 19x19.

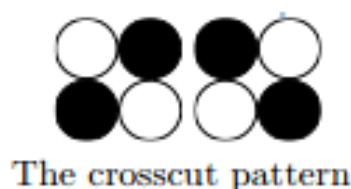
O jogador que controla as peças pretas joga primeiro, colocando uma peça preta em qualquer local do tabuleiro. As jogadas seguintes são alternadas entre os jogadores e cada jogada consiste na colocação de uma ou duas peças em qualquer espaço livre. No caso em que sejam jogadas duas peças:

a) só podem ser jogadas na vertical ou horizontal

b) não podem resultar na formação de um *triplet* ou *crosscut*.

Um *triplet* refere-se a três peças da mesma cor seguidas em qualquer direção.

Um *crosscut* refere-se a um bloco de 2x2 em que cantos opostos são controlados por jogadores diferentes



O jogador que controla as peças pretas ganha se, em qualquer momento do jogo, existir um caminho de peças pretas que se prolongue desde a linha superior até à inferior. As peças brancas ganham se houver um caminho que ligue a coluna mais à esquerda até à mais à direita.

2 - Representação do Estado de jogo

Representação do estado inicial do jogo:

[[0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0],]

	a	b	c	d	e	f	g	h	i
1									
2									
3									
4									
5									
6									
7									
8									
9									

Representação de um estado intermédio do jogo:

[[1,0,0,0,0,0,0,0,0],

[0,2,2,0,0,0,0,0,0],

[2,0,0,0,0,0,0,0,0],

[0,2,0,0,1,0,0,0,0],

[0,2,0,0,1,0,0,0,0],

[2,0,1,1,0,0,0,0,0],

[2,1,0,0,0,0,0,0,0],

[2,1,0,0,0,0,0,0,0],

[1,0,0,0,0,0,0,0,0]]

	a	b	c	d	e	f	g	h	i
1	X								
2		0	0						
3	0								
4		0			X				
5		0			X				
6	0		X	X					
7	0	X							
8	0	X							
9	X								

Representação de um estado final do jogo:

[[1,0,0,2,2,1,0,0,0],

[0,2,2,0,0,1,0,0,0],

[2,0,0,0,0,1,0,0,0],

[0,2,0,0,1,0,0,0,0],

[0,2,0,0,1,0,0,0,0],

[2,0,1,1,0,0,0,0,0],

[2,1,0,0,0,0,0,0,0],

[2,1,0,0,0,0,0,0,0],

[1,0,0,0,0,0,0,0,0]]

	a	b	c	d	e	f	g	h	i
1	X			0	0	X			
2		0	0			X			
3	0					X			
4		0			X				
5		0			X				
6	0		X	X					
7	0	X							
8	0	X							
9	X								

3 - Visualização do Tabuleiro

Para mostrar o tabuleiro de jogo, será utilizado, em princípio, o seguinte código:

```
starting_state([  
[0,0,0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0,0,0],  
[0,0,0,0,0,0,0,0,0] ]).
```

Num estado inicial define o estado inicial do tabuleiro.

```
printColumnId:- write(' a  b  c  d  e  f  g  h  i\n').  
  
printInitialSeperator:- write( ' ----- \n').
```

Posteriormente, preenchemos a linha de cabeçalho do tabuleiro com as letras que servirão de guia para a colocação das peças no mesmo, aquando do começo do jogo. Após essa linha imprime-se uma linha continua para começar as divisórias do tabuleiro.

Três predicados onde se define o que apresentar no tabuleiro quando colocamos os diferentes números apresentados (0, 1 ou 2)

```
write_line(T).
```

```
show_lines(N2, T).
```

8


```
show_board(Board):-
```

```
    nl, printColumnId, printInitialSeperator,
```

```
    show_lines(1, Board), nl.
```

```
start:- starting_state(Board),  
show_board(Board).
```

Secção do código
encarregue de
imprimir o
tabuleiro
completo. De
notar que o
predicado
show_board usa
o show_lines.

4 - Jogadas

Cabeçalho do predicado de colocação de uma peça:

One_piece(Row, Column, Board)

Cabeçalho do predicado de colocação de duas peças:

Two_pieces(Row, Column, Row, Column, Board)