

Programação em Lógica

Relatório Intercalar: “Mad Bishops”

Turma 4

Beatriz Mendes: *up201604253@fe.up.pt*

João Malheiro: *up201605926@fe.up.pt*

Índice

Mad Bishops	2
Bispos	2
Capturar Peças	2
Vencedor	2
Representação do tabuleiro	3
Estado Inicial	3
Estado intermédio	3
Estado final	4
Representação do jogo	4
Referências	5

Mad Bishops

“*Mad Bishops*” foi inventado por Mark Steere em Março de 2010. O seu objetivo é capturar todas as peças do adversário.

O jogo “*Mad Bishops*” é jogado por dois jogadores num tabuleiro de 10x10 casas. Inicialmente, cada jogador possui 25 bispos vermelhos ou azuis. Ambos os jogadores jogam em turnos alternados, movendo os bispos da sua cor: um movimento por turno, começando pelo jogador que possui as peças de cor vermelha. Os jogadores são obrigados a jogar nos seus turnos, não podendo passar a vez.

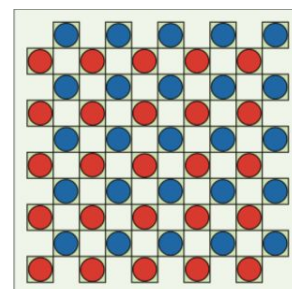


Figura 1: tabuleiro inicial

Bispos

Os bispos movem-se similarmente ao movimento dos bispos em Xadrez: um bispo pode ser movido nas suas diagonais, tanto para uma casa adjacente ou uma casa separada por uma ou mais quadrículas contíguas vazias.

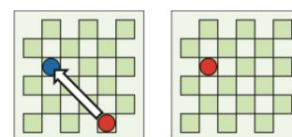


Figura 2: capturar adversário

Capturar Peças

Um bispo captura outra peça substituindo a peça de outro jogador e colocar a sua.

Quando uma peça se encontra numa posição em que pode capturar uma peça do adversário, o jogador é obrigado a fazê-lo se o pretende mover. Por exemplo, na figura 3, o jogador ao qual pertence a peça vermelha apenas lhe é permitido fazer o jogada mais à direita, de forma a capturar a peça do adversário.

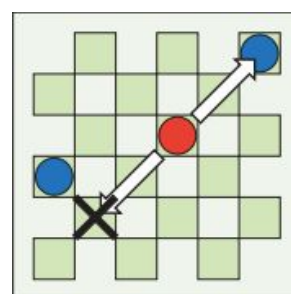


Figura 3: movimento obrigatório

No entanto, caso o jogador tenha mais peças que possa mover sem capturar, é permitido que o faça.

No exemplo da figura 4, o jogador tem a opção de fazer ambos os movimentos.

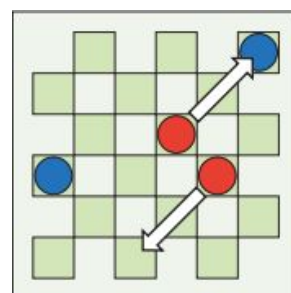


Figura 4: movimentos possíveis

Vencedor

Vence o jogo, o último jogador com peças suas no tabuleiro.

Representação do tabuleiro

Estado Inicial

```
initialBoard([
  [0, 2, 0, 2, 0, 2, 0, 2, 0, 2],
  [3, 0, 3, 0, 3, 0, 3, 0, 3, 0],
  [0, 2, 0, 2, 0, 2, 0, 2, 0, 2],
  [3, 0, 3, 0, 3, 0, 3, 0, 3, 0],
  [0, 2, 0, 2, 0, 2, 0, 2, 0, 2],
  [3, 0, 3, 0, 3, 0, 3, 0, 3, 0],
  [0, 2, 0, 2, 0, 2, 0, 2, 0, 2],
  [3, 0, 3, 0, 3, 0, 3, 0, 3, 0],
  [0, 2, 0, 2, 0, 2, 0, 2, 0, 2],
  [3, 0, 3, 0, 3, 0, 3, 0, 3, 0]
]).
```

	0	1	2	3	4	5	6	7	8	9
0	.	R	.	R	.	R	.	R	.	R
1	B	.	B	.	B	.	B	.	B	.
2	.	R	.	R	.	R	.	R	.	R
3	B	.	B	.	B	.	B	.	B	.
4	.	R	.	R	.	R	.	R	.	R
5	B	.	B	.	B	.	B	.	B	.
6	.	R	.	R	.	R	.	R	.	R
7	B	.	B	.	B	.	B	.	B	.
8	.	R	.	R	.	R	.	R	.	R
9	B	.	B	.	B	.	B	.	B	.

Figura 5: tabuleiro em estado inicial

Estado intermédio

```
mediumBoard([
  [0, 1, 0, 2, 0, 1, 0, 2, 0, 3],
  [3, 0, 1, 0, 2, 0, 1, 0, 1, 0],
  [0, 2, 0, 3, 0, 1, 0, 3, 0, 1],
  [1, 0, 1, 0, 1, 0, 1, 0, 2, 0],
  [0, 2, 0, 2, 0, 3, 0, 1, 0, 3],
  [2, 0, 1, 0, 1, 0, 1, 0, 1, 0],
  [0, 3, 0, 2, 0, 3, 0, 3, 0, 1],
  [1, 0, 1, 0, 3, 0, 1, 0, 2, 0],
  [0, 1, 0, 3, 0, 2, 0, 3, 0, 3],
  [2, 0, 1, 0, 1, 0, 3, 0, 1, 0]
]).
```

	0	1	2	3	4	5	6	7	8	9
0	.	.	.	R	.	.	.	R	.	B
1	B	.	.	.	R
2	.	R	.	B	.	.	.	B	.	.
3	R	.
4	.	R	.	R	.	B	.	.	.	B
5	R
6	.	B	.	R	.	B	.	B	.	.
7	B	.	.	.	R	.
8	.	.	.	B	.	R	.	B	.	B
9	R	B	.	.	.

Figura 6: tabuleiro em estado intermédio

Estado final

```
finalBoard([
  [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
  [1, 0, 1, 0, 2, 0, 1, 0, 1, 0],
  [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
  [1, 0, 1, 0, 1, 0, 1, 0, 1, 0],
  [0, 1, 0, 2, 0, 1, 0, 1, 0, 1],
  [1, 0, 1, 0, 1, 0, 1, 0, 1, 0],
  [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
  [1, 0, 1, 0, 1, 0, 1, 0, 1, 0],
  [0, 1, 0, 1, 0, 1, 0, 2, 0, 1],
  [1, 0, 1, 0, 1, 0, 1, 0, 1, 0]
]).
```

	0	1	2	3	4	5	6	7	8	9
0
1	R
2
3
4	.	.	.	R
5
6
7
8	R	.	.
9

Figura 7: tabuleiro em estado final**Representação do jogo**

O predicado principal na representação do jogo é *'displayGame'*, implementado com recursividade. Este é chamado através de *'start.'* Este predicado recebe o estado do jogo, o qual é constituído por uma lista de listas que contém, respetivamente, o **tabuleiro** e o **jogador** atual (*"red"* ou *"blue"*).

Como ilustrado na *figura 8*, os componentes anteriormente referidos são impressos na consola.

O tabuleiro, em si, é impresso através de recursividade. Em *'displayGame'* é chamado o predicado *'printTable'*. Este imprime cada linha, recursivamente, chamando *'printList'* com a "head" do tabuleiro e a si próprio com a "tail", que por sua vez, imprime cada "casa" através do predicado *'printCell'*.

M A D B I S H O P S										
PLAYER: red										
	0	1	2	3	4	5	6	7	8	9
0	.	R	.	R	.	R	.	R	.	R
1	B	.	B	.	B	.	B	.	B	.
2	.	R	.	R	.	R	.	R	.	R
3	B	.	B	.	B	.	B	.	B	.
4	.	R	.	R	.	R	.	R	.	R
5	B	.	B	.	B	.	B	.	B	.
6	.	R	.	R	.	R	.	R	.	R
7	B	.	B	.	B	.	B	.	B	.
8	.	R	.	R	.	R	.	R	.	R
9	B	.	B	.	B	.	B	.	B	.

Figura 8: resultado de displayGame**Legenda:**

[0] espaço ocupado [1] espaço livre [2] bispo vermelho [3] bispo azul

Referências

- [1] “IG Game Center” <http://www.iggamecenter.com/info/en/madbishops.html>
- [2] “Mark Steere Games” http://www.marksteeregames.com/Mad_Bishops_rules.pdf