

Xadrersi

Relatório Intercalar



Mestrado Integrado em Engenharia Informática e
Computação

Programação em Lógica

Grupo Xadrersi_1:

António Cunha Seco Fernandes de Almeida - 201505836

João Paulo Madureira Damas - 201504088

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

15 de Outubro de 2017

1 O Jogo Xadrersi

Origens

Xadrersi é um jogo de tabuleiro para duas pessoas criado por Andi Lewicki, em 2003. Como o nome sugere, surge de uma combinação entre os conhecidos jogos de tabuleiro Xadrez e Reversi.

Regras

O jogo é disputado num tradicional tabuleiro 8x8, inicialmente vazio. A seu dispor, cada jogador possui as peças de que disporia se estivesse numa partida de Xadrez, exceto peões. Por outras palavras, cada jogador possui um rei, uma rainha, dois cavalos, dois bispos e duas torres.

Os jogadores vão colocando as suas peças no tabuleiro, uma a uma, alternadamente, até todas as peças estarem em jogo. Após colocada, uma peça não pode ser movida nem retirada (capturada). No entanto, ataca outras posições do tabuleiro segundo as regras de Xadrez (o rei ataca as posições adjacentes, o bispo diagonalmente, o cavalo em 'L', etc). Não se aplicam regras de cheque nem afastamento obrigatório entre reis, logo um rei pode ser colocado numa posição onde ficaria em cheque e os reis podem ficar em posições adjacentes. Os bispos devem ficar em posições de cores opostas, como no Xadrez.

De uma maneira geral, cada jogada é válida se e só se a peça colocada estiver a ser atacada por pelo menos uma peça adversária. A primeira exceção a esta regra é a primeira jogada, em que não há peças no tabuleiro. O jogador com as peças brancas começa e a primeira peça jogada é obrigatoriamente o rei, podendo ser colocado em qualquer posição. As figuras 1, 2 e 3 ilustram uma sequência de jogadas válidas (com as hipóteses em cada turno destacadas) a partir da posição inicial.

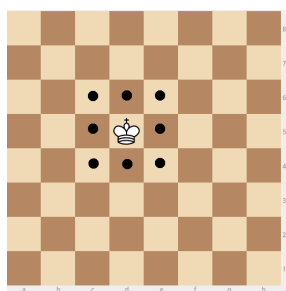


Figura 1: Após a primeira jogada, uma peça preta tem de ser colocada adjacente ao rei branco de modo a poder ser atacada por este, constituindo uma jogada válida

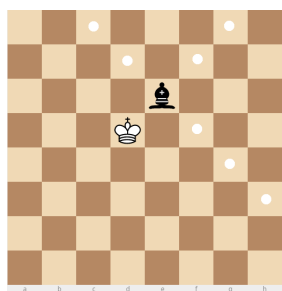


Figura 2: Agora, tendo o jogador de peças pretas jogado um bispo, o jogador de peças brancas deve colocar uma peça que interseje o mesmo segundo uma das suas diagonais livres

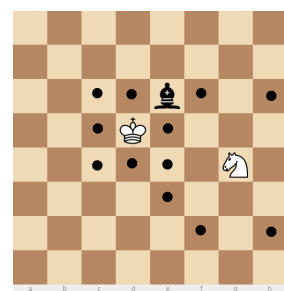


Figura 3: Neste caso, tendo já mais do que uma peça branca em jogo, o jogador de peças pretas apenas precisa de se certificar que a peça que colocar é atacada por pelo menos uma peça branca

Existe ainda outro caso excecional. Caso seja impossível colocar uma peça de tal modo que seja atacada por uma peça adversária, o jogador pode colocar a peça em qualquer posição livre no tabuleiro. Na figura 4 demonstra-se esta situação através de um caso simples.

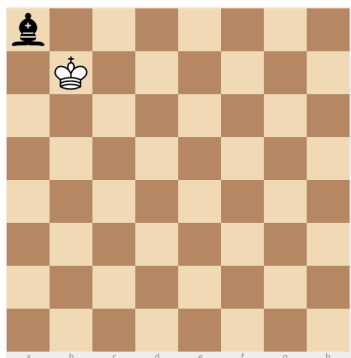


Figura 4: Após a jogada inicial do rei, o bispo foi colocado em A8, adjacente ao rei e por isso válido. No entanto, a sua única diagonal está bloqueada e por isso nenhuma posição livre é atacada pelo mesmo. Nesta situação, aplica-se a exceção e a próxima peça branca pode ser colocada em qualquer posição livre.

Para além da obrigatoriedade do rei branco ser a primeira peça jogada, existem ainda duas regras adicionais a assinalar para tentar garantir o equilíbrio do jogo:

- O rei preto é sempre a última peça a ser colocada no tabuleiro
- Quando um jogador coloca em jogo a sua rainha, o outro é obrigado, no turno seguinte, a colocar também a sua rainha em jogo

Objetivo

O objetivo é conseguir o maior número de pontos quando o jogo terminar, ou seja, quando todas as peças forem colocadas no tabuleiro. O número de pontos de um jogador corresponde ao número total de casas que as suas peças atacam. Caso várias peças ataquem a mesma casa, esta é contabilizada uma vez por cada peça que a ataca. Existe, por isso, a possibilidade de empate.

2 Representação do Estado do Jogo

De forma a sistematizar uma relação entre a representação interna das peças de jogo com a visualização das mesmas na consola, foi definida uma convenção. Os caracteres apresentados no ecrã baseam-se na nomenclatura em Inglês das peças de Xadrez, distinguindo-se as cores das mesmas através da utilização de maiúsculas ou minúsculas.

Peças Brancas

Rei	1	'K'
Rainha	2	'Q'
Bispo	3	'B'
Cavalo	4	'H'
Torre	5	'R'

Peças Pretas

Rei	6	'k'
Rainha	7	'q'
Bispo	8	'b'
Cavalo	9	'h'
Torre	10	'r'

Por último, uma casa vazia é internamente representada como 0 e visualizada na consola como um espaço vazio.

Estado Inicial de 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]]
```

Possível Estado Intermédio

```
[[0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 10, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 4, 0],
 [0, 0, 0, 1, 0, 0, 0, 0],
 [0, 0, 0, 0, 8, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0]]
```

Possível Estado Final

```
[[10, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 9, 0, 0, 6, 5],
 [0, 4, 0, 3, 10, 2, 0, 0],
 [0, 0, 0, 0, 0, 8, 4, 0],
 [0, 0, 7, 1, 3, 0, 0, 0],
 [0, 0, 0, 0, 8, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 9, 0],
 [0, 0, 5, 0, 0, 0, 0, 0]]
```

3 Visualização do Tabuleiro

Em modo de texto, serão utilizados os predicados seguintes para produzir uma representação dos estados de jogo. Bocados de código

estão omitidos com reticências - '(...)' - por se tratarem de repetições de caracteres ou padrões de caracteres.

```
getPieceDisplay(0, ' ').
```

```
%white pieces
```

```
getPieceDisplay(1, 'K').
getPieceDisplay(2, 'Q').
getPieceDisplay(3, 'B').
getPieceDisplay(4, 'H').
getPieceDisplay(5, 'R').
```

```
%black pieces
```

```
getPieceDisplay(6, 'k').
getPieceDisplay(7, 'q').
getPieceDisplay(8, 'b').
getPieceDisplay(9, 'h').
getPieceDisplay(10, 'r').
```

```
displayBoard( Board ) :-
    displayBoardHeader ,
    N is 8, displayBoardTail(Board, N),
    displayBottom.
```

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

```
displayBoardTail( [ Line | T ], N ):-
    write(' | _____ | _____ (... ) _____ | '), nl,
    write(' _____ | _____ (... ) _____ '), nl,
    write(' | _____ | _____ (... ) _____ | '), nl,
    displayLine(Line, N), nl,
    N1 is N-1,
    displayBoardTail(T, N1).
```

```
displayLine( [], N):-
    write(' | _____ '), displayNumber(N).
```

```
displayLine( [ CurrentPiece | T ], N ):-
    write(' | _____ '),
    getPieceDisplay(CurrentPiece, PieceDisplay),
    write(PieceDisplay),
    write(' _____ '),
    displayLine(T, N).
```

```
displayBoardHeader:-
    nl,
    write( '         a         b         (... )         g         h         ' ), nl.
```

```
displayBottom:-
    write( ' |         |         (... )         |         | ' ), nl,
    write( ' _____ ( ... ) _____ ' ), nl.
```

```
displayNumber(N) :- write(N), !.
```

Os exemplos de estados de jogo apresentados na secção 2 são representados da seguinte forma:

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

Figura 5: *Representação textual do estado de jogo inicial.*

a	b	c	d	e	f	g	h	
								8
								7
				r				6
						H		5
			K					4
				b				3
								2
								1

Figura 6: *Representação textual de um estado intermédio.*

a	b	c	d	e	f	g	h	
r								8
			h			k	R	7
	H		B	r	Q			6
					b	H		5
		q	K	B				4
				b				3
						h		2
		R						1

Figura 7: *Representação textual de um estado final.*

4 Movimentos

Como o jogo não permite alterar a posição de uma peça após a sua colocação no tabuleiro, nem é permitido captura das mesmas, apenas existe um tipo de movimentação.

Colocar uma peça no tabuleiro

`setPiece(Piece , Row, Column, Board)`.