

Breakthru

Relatório Intercalar



Mestrado Integrado em Engenharia Informática e
Computação

Programação em Lógica

Grupo Breakthru 2:

João Rafael de Figueiredo Cabral - 201304395

João Bernardo Martins de Sousa e Silva Mota - 201303462

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

11 de Outubro de 2015

1 O Jogo Breakthru

O *Breakthru* é um jogo de estratégia abstrata para dois jogadores em que cada lado começa com um número diferente de peças, colocadas em posições assimétricas e com objetivos diferentes para assegurar a vitória.

1.1 História

O jogo foi criado em 1965 por Alex Randolph, com um conjunto de regras baseadas nas do *Hnefatafl*¹, um jogo de estratégia abstrata cuja popularidade remonta à Escandinávia medieval, existindo vestígios da sua presença entre várias civilizações antigas².

1.2 Regras

1.2.1 Peças

A cor de cada jogador é sorteada através do lançamento de uma moeda, sendo que o jogador vencedor desse sorteio é o jogador dourado e o perdedor é o jogador prateado. O tabuleiro conta com 13 peças douradas, das quais 12 dispostas em quadrado à volta da 13ª peça, no centro do tabuleiro, e de 20 peças prateadas dispostas em volta das peças douradas.

O jogador dourado dispõe as suas 12 peças normais como desejar na zona central do tabuleiro, nele delimitada. O jogador prateado procede de forma semelhante, dispondo contudo as suas 20 peças fora da zona central, reservada ao jogador dourado.

Uma das peças douradas é diferente das restantes, sendo designada por navio almirante. A sua posição inicial é fixa, sendo obrigatório que seja colocada no centro do tabuleiro.

1.2.2 Movimentos e Capturas

Cada jogador deve, na sua vez, efetuar dois movimentos ou uma captura. O jogador dourado escolhe quem deve jogar primeiro.

Todas as peças podem ser movimentadas um número arbitrário de espaços livres na horizontal ou vertical, à semelhança do movimento de uma torre no xadrez. Durante o movimento nenhuma peça pode saltar por cima de outra ou ocupar uma casa já ocupada, como ilustrado na figura 3.

As capturas são feitas movimentando uma peça para uma casa que lhe seja diagonalmente adjacente e que esteja ocupada por uma peça do adversário, como ilustrado na figura 2, sendo esta removida do jogo.

Quando o navio almirante dourado efetua um movimento o jogador dourado não pode efetuar outro movimento nesse turno.

1.2.3 Objetivo do Jogo

O jogador prateado ganha o jogo ao capturar o navio almirante dourado. O jogador dourado ganha o jogo se o navio almirante chegar a uma das casas exteriores do tabuleiro.

¹<https://boardgamegeek.com/boardgame/335/breakthru>

²<https://boardgamegeek.com/boardgame/2932/hnefatafl>

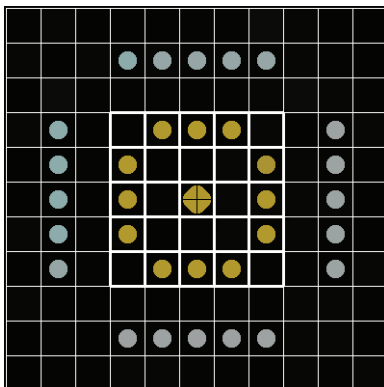


Figura 1: Uma posição inicial possível das peças num tabuleiro de Breakthru

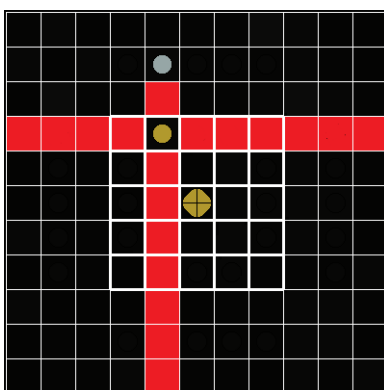


Figura 2: A vermelho: movimentos possíveis da peça dourada

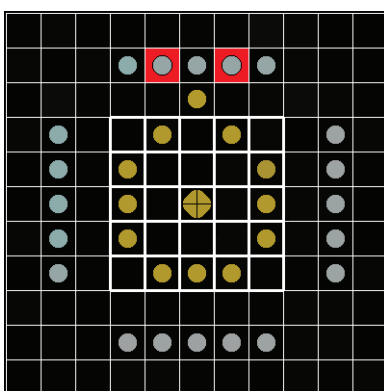


Figura 3: A vermelho: capturas possíveis da peça dourada

2 Representação do Estado do Jogo

O tabuleiro é representado como uma lista de listas. De seguida apresentam-se exemplos de tabuleiros iniciais, intermédios e finais criados em Prolog.

Possível estado inicial:

```
[[' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
 [' ',' ',' ',' ','p','p','p','p','p',' ',' ',' ',' ',' '],
 [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
 [' ','p',' ',' ',' ','d','d','d',' ',' ',' ','p',' ',' '],
 [' ','p',' ',' ','d',' ',' ',' ','d',' ',' ','p',' ',' '],
 [' ','p',' ',' ','d',' ',' ','F',' ','d',' ',' ','p',' ',' '],
 [' ','p',' ',' ','d',' ',' ',' ','d',' ',' ','p',' ',' '],
 [' ','p',' ',' ',' ','d','d','d',' ',' ',' ','p',' ',' '],
 [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
 [' ',' ','p','p','p','p','p','p',' ',' ',' ',' ',' '],
 [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ']]
```

Possível estado intermédio:

```
[[' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
 [' ',' ',' ','p','p',' ',' ','d','p',' ',' ',' ',' ',' '],
 [' ',' ',' ','d',' ',' ',' ',' ',' ',' ',' ',' ',' '],
 [' ',' ','p','d',' ',' ','d','p',' ',' ',' ',' ',' '],
 [' ','p',' ',' ',' ','p',' ',' ','d',' ',' ','p',' ',' '],
 [' ','p',' ',' ','d',' ',' ','F',' ','d',' ',' ','p',' ',' '],
 [' ','p',' ',' ','d',' ',' ',' ','d',' ',' ','p',' ',' '],
 [' ','p',' ',' ','d','d','d',' ',' ',' ','p',' ',' '],
 [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
 [' ',' ','p','p','p','p','p','p',' ',' ',' ',' ',' '],
 [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ']]
```

Possível estado final, onde as peças douradas venceram porque o navio almirante atingiu uma das casas exteriores:

```
[[' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
 [' ',' ',' ','p','p',' ',' ','d','p',' ',' ',' ',' ',' '],
 [' ',' ',' ','d',' ',' ',' ',' ',' ',' ',' ',' ',' '],
 [' ',' ','p','d',' ',' ','d','p',' ',' ',' ',' ',' '],
 [' ','p',' ',' ',' ','p',' ',' ','d',' ',' ','p',' ',' '],
 [' ','p',' ',' ','d',' ',' ',' ',' ','d','p',' ',' '],
 [' ','p',' ',' ','d',' ',' ',' ',' ','d',' ',' ','p',' ',' '],
 [' ','p',' ',' ','p','d',' ',' ','d',' ',' ','p',' ',' '],
 [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
 [' ',' ','p',' ','d',' ',' ','p','p',' ',' ',' ',' '],
 [' ',' ',' ',' ',' ',' ','F',' ',' ',' ',' ',' ']]
```

3 Visualização do Tabuleiro

Foram desenvolvidos alguns predicados em Prolog para permitir a visualização do tabuleiro usando caracteres *ASCII*, como exemplificado pela figura 4.

				p	p	p	p	p				
		p			d	d	d				p	
		p		d				d			p	
		p		d		F		d			p	
		p		d				d			p	
		p			d	d	d				p	
			p	p	p	p	p	p				

Figura 4: Tabuleiro inicial de um jogo de Breakthru

3.1 Predicados de Visualização

Foram implementados os predicados de visualização que se seguem:

```
printBoard([]):-write('-----').
printBoard([H|T]):-
    write('-----'),nl,
    printLine(H, '|'), nl,
    printBoard(T).

printLine([], Sep):-
    write(Sep).
printLine([H|T], Sep):-
    write(Sep),write(' '), write(H), write(' '),
    printLine(T, Sep).
```

4 Movimentos

São possíveis movimentos simples ou capturas, para os quais se desenvolverão os predicados cujos cabeçalhos se elencam na secção 4.1, que devem suceder somente se o movimento for legal ou for possível efetuar a captura, respetivamente.

4.1 Cabeçalhos dos Predicados

```
movePiece(RowOrigin, ColumnOrigin, RowDestiny, ColumnDestiny,
    Player, Board, NewBoard):-

capturePiece(RowOrigin, ColumnOrigin, RowDestiny, ColumnDestiny,
    Player, Board, NewBoard):-
```