

Dameo

Relatório



Mestrado Integrado em Engenharia Informática e
Computação

Programação em Lógica

Grupo Dameo_4:

Luís Miguel Cardoso Lopes Correia - up201503342

Vicente Fernandes Ramada Caldeira Espinha - up201503764

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

12 de Novembro de 2017

1 Resumo

O jogo consiste num jogo de tabuleiro e foi criado um tabuleiro e guardado numa lista. Existem dois tipos de jogadores: humano, a quem é perguntado a posição inicial e final do movimento, e computador. Se a jogada não for válida, aparece uma mensagem a dizer-lo. Em relação ao jogador computador, ele escolhe aleatoriamente a peça inicial e a posição final, verificando, de seguida, se a jogada é válida ou não. Se não for, volta a gerar aleatoriamente uma "jogada". Para o jogo terminar, basta que um dos jogadores fique com zero peças no tabuleiro.

2 Índice

• Resumo	2
• Índice	3
• Introdução	4
• Dameo	5
• Lógica do Jogo	7
– Representação do estado do Jogo	7
– Visualização do tabuleiro	8
– Lista de Jogadas válidas	9
– Execução de Jogadas	9
– Avaliação do tabuleiro	9
– Final do Jogo	9
– Jogada de Computador	9
• Interface com o utilizador	10
• Conclusões	11
• Bibliografia	11
• Anexos	11

3 Introdução

O objectivo deste trabalho passou pelo desenvolvimento de um jogo de tabuleiro Dameo. Este relatório surge como um complemento ao trabalho prático, para que qualquer um perceba o trabalho desenvolvido. Assim, o relatório está dividido nas seguintes secções:

- Descrição do Jogo - breve descrição do jogo, história e regras;
- Lógica do Jogo - explicação por detrás da implementação do jogo, representação do estado de jogo, visualização do tabuleiro, validação das jogadas, avaliação do tabuleiro, verificação do fim do jogo e jogada por parte do computador;
- Conclusões - conclusões retiradas da elaboração do projeto.

4 Dameo

Dameo is a strategy board game for two players invented by Christian Freeling in 2000. It is a variant of the game draughts (or checkers) and is played on an 8x8 checkered game board. This game owes much of its existence to the game Croda invented in 1995 by Ljuban Dedić of Croatia, Associate Professor of Mathematics at the University of Split and the 1989 International Checkers champion of the former Yugoslavia. Croda, the result of a search for the checkers variant with the smallest percentage of draws, is itself a variation of Turkish draughts. Dameo differs from Croda in its initial setup and number of men, enhanced movement of kings, and the addition of linear movement of a line of men.

Rules:

The rules mention men and kings. A king is a promoted man. If the difference doesn't matter, they may also mention pieces, for instance 'the number of pieces on the board'.

Linear movement:

Linear movement is defined as the move of a straight unbroken line of men of the same color, one square along the line of squares they occupy, provided the square in front of is vacant. It includes the move of a single man, which may be considered as a line-of-one. Linear movement does not apply to kings.

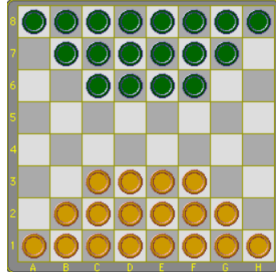


Figura 1: Demo Board

The diagram shows the initial position. The reason for this choice is natural tendency of draughts games that allow straight forward movement, to build up forces along the sides. This initial position creates a certain balance between the center and the sides. The object is to leave your opponent without a valid move, either by capturing all his pieces, or by blocking them completely. Draws may occur by mutual exhaustion of material.

Movement:

Men, whether single or linear, move forward only. A king moves queenwise, as indicated. In Dameo, the player can open with any man. On the left are 26 unique opening moves white has at its disposal. For each of these there's a symmetric one. Black, on his first move, doesn't face a symmetric situation, so he actually has a choice of 52 answers. If a man moves onto the opponent's back row, it is promoted as a king. If a linear move reaches the back row, only the head man is promoted.

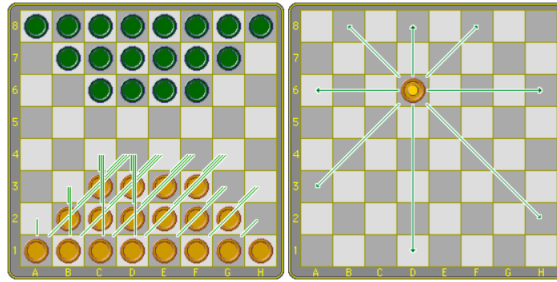


Figura 2: Demo Board

Capture:

Capture takes precedence over a non-capturing move. Only if the player to move has no capture to make, may he move a single man, or a line of men, or a king.

- Although pieces may move diagonally, all captures are straight only. Men may capture forwards, backwards and sideways by short leap. Kings may move queenwise, but they capture only rookwise, by the long leap.
- Capturing, whether by a men or kings, is compulsory. If a piece makes a capture and is now in a position to make another one, it must do so. Thus multiple captures may be made in the same turn. When a multiple capture is being made, the captured pieces are only removed at the end of the turn, and it is not allowed to jump over the same piece twice in that turn, although vacant squares may be passed over more than once.
- Majority capture takes precedence: if a player has a choice of captures, he must choose the one that results in the largest number of pieces being captured (kings and men counting equally). When a king has more than one option in terms of captures and destination squares, it must choose its route so that it maximizes the capturing sequence. If there is more than one way to do this, it is free to choose.
- if a man ends its move on the opponent's back row, it is promoted to king. A man passing the back row in a capture, but not ending on it, does not promote.

Since a capturing sequence must be completed before the pieces are taken off the board, and since a man may not be jumping twice, Dameo allows for the 'Coup Turc'.

5 Lógica do Jogo

5.1 Representação do estado do Jogo

O tabuleiro é uma lista de listas, cada lista é uma linha do tabuleiro (como uma matriz). Os diferentes estados do tabuleiro vão ser representados como no esquema abaixo. As células '1' são uma peça normal da primeira equipa e as '2' são as da segunda equipa. As células '3' e '4' representam os reis da primeira e segunda equipa, respetivamente. Os '0' são casas vazias.

Exemplo de um estado inicial de um tabuleiro:

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

Exemplo de um estado intermedio de um tabuleiro:

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

Exemplo de um estado final de um tabuleiro:

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

5.2 Visualização do tabuleiro

O display do tabuleiro é feito usando recursividade.

Primeiramente, uma linha (uma lista de listas) é analisada e é feito o seu display. O processo de análise é o seguinte : se a célula '1' é detetada, vai imprimir para a consola como 'o'; o mesmo acontece com as outras células('0' imprime ' ', 2' imprime 'x', '3' imprime 'O', '4' imprime 'X'). Quando a lista é vazia (fim da lista), o processo termina.

De seguida, o predicado `show_line` é chamado noutro predicado, que processa a lista até não haver mais (fim do tabuleiro).

Também adicionamos algumas limites e separações, de forma a ficar mais intuitivo para o jogador.

```
show_board ([]) :- write(' _____\n').
show_board ([A|B]) :-
    write(' _____\n'),
    show_line(A),
    nl,
    show_board(B).
```

```
show_line ([]) :- write(' | ').
show_line ([A|B]) :-
    write(' | '),
    analyse_piece(A),
    show_line(B).
```

```
analyse_piece(X) :-
    blank_space(X);
    first_team(X);
    second_team(X);
    first_team_king(X);
    second_team_king(X).
```

```
blank_space(X) :- X == 0, write(' ').
first_team(X) :- X == 1, write('o').
second_team(X) :- X == 2, write('x').
first_team_king(X) :- X == 3, write('O').
second_team_king(X) :- X == 4, write('X').
```

```
print :- board(X), show_board(X).
```


	o		o		o		o		o		o		o	
	o		o		o		o		o		o		o	
			o		o		o		o					
			x		x		x		x					
		x		x		x		x		x				
	x		x		x		x		x		x			

Figura 3: Dameo's Console Board

5.3 Lista de Jogadas Válidas

Da maneira como foi implementado o jogo, nunca obtemos uma lista de jogadas válidas, apenas verificamos se a jogada é válida após o jogador ou o computador tentarem fazer uma.

5.4 Execução de Jogadas

"movemente(Player,Board,Row,Col,NewRow,NewCol,NewBoard)"

O predicado movemente verifica se a nova peça chegou à ponta contrária do tabuleiro e se tranformará num rei.De seguida, atualiza o tabuleiro, colocando a posição anterior da peça como uma casa vazia e a nova posição com o elemento respetivo, retornando a NewBoard(novo estado do tabuleiro)

5.5 Avaliação do tabuleiro

A avaliação do tabuleiro é feita principalmente através dos predicados "validMovement"(se for uma peça normal,verifica se o movimento que o jogador fez é valido),"valideKingMovement"(se for um King,verifica se o movimento que o jogador fez é valido) e "checkEaten"(verifica se havia a oportunidade de comer alguma peça)

5.6 Final do jogo

"checkEnd(Play,NewPlay,Board,NewEnd)"

O final do jogo é verificado pelo predicado checkEnd que retorna NewEnd a 0, se ainda não houver vencedor,1 se as peças "o" ganharam ou 2 se as peças "x" venceram.

5.7 Jogada de Computador

O computador escolhe a sua jogada (posição inicial e final) de forma aleatória e, de seguida, verifica se esta é válida ou não.

6 Interface com o utilizador

A Interface do utilizador encontra-se maioritariamente no ficheiro "menu.pl".

Na imagem seguinte, podemos observar os menus do jogo, sendo que a opção sair, fecha o Sictus.

```

-----
Daneo
-----
1: Jogar
2: Creditos
3: Sair
-----
Introduza a sua escolha
|: 1
-----
Daneo
-----
1: Jogador vs Jogador
2: Jogador vs Computador
3: Computador vs Computador
4: Voltar ao menu principal
-----
Introduza a sua escolha
|: █

```

Na imagem seguinte, podemos observar a interface durante um jogo entre dois humanos.

```

-----
1 | x | x | x | x | x | x | x | x |
2 | | x | x | x | x | x | x | |
3 | | | x | x | x | x | | |
4 | | | | | | | | |
5 | | | | | | | | |
6 | | | o | o | o | o | | |
7 | | o | o | o | o | o | o | |
8 | o | o | o | o | o | o | o | o |
-----
  A  B  C  D  E  F  G  H
Jogada | "x" | "o" | "X" | "O"
1      | 18 | 18 | 0  | 0
-----
Jogam as "o"
Introduza a coluna da peça a mover:a
Introduza a linha da peça a mover:8
Introduza a coluna da posicao a mover:a
Introduza a nova linha da posicao a mover:7

```

Num jogo entre dois computadores, apenas imprime a posição inicial e final da peça, através do predicado `displayPlayPc(Type,Player,Row,Col,NewRow,NewCol)`, e no fim do jogo o tabuleiro é mostrado.

7 Conclusões

Com este projeto é possível concluir que a linguagem Prolog se destingue claramente de outras linguagens de programação e o planeamento de cada predicado é um fator muito importante. A adaptação a esta linguagem foi complicada, contudo depois de estar feito a lógica para "Jogador vs Jogador", a passagem para modo computador foi rápida. Todavia, não conseguimos implementar as duas dificuldades do jogo, apesar de o jogador não sentir diferença, pois embora para uma dificuldade superior precisaríamos de ir buscar a melhor jogada, a lógica do jogo já obriga a que tenha sempre de comer o maior número de peças. Assim, apesar de o computador escolher posições aleatórias, apenas as válidas servirão para a jogada.

8 Bibliografia

"<http://www.mindsports.nl/index.php/arena/dameo/>"
'<http://www.swi-prolog.org/>'

9 Anexos e Notas

Nota:

- Como tentamos até pouco tempo antes da entrega, calcular a melhor jogada possível, não tivemos tempo para traduzir a parte Dameo do relatório intercalar, que tinha sido feita em inglês.
- Para correr o Jogo, basta fazer consult no ficheiro "menu.pl" e executar "dameo."
- Os ficheiros encontram-se na pasta em que está este documento e são "dameo.pl", "menu.pl" e "utilities.pl"