Título do Trabalho

Relatório Final

Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

Grupo 04:
João Nuno Fonseca Seixas - 201505648
Renato Alexandre Sousa Campos - 201504942

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

12 de Novembro de 2017

Resumo

Resumo sucinto do trabalho com 150 a 250 palavras (problema abordado, objetivo, como foi o problema resolvido/abordado, principais resultados e conclusões).

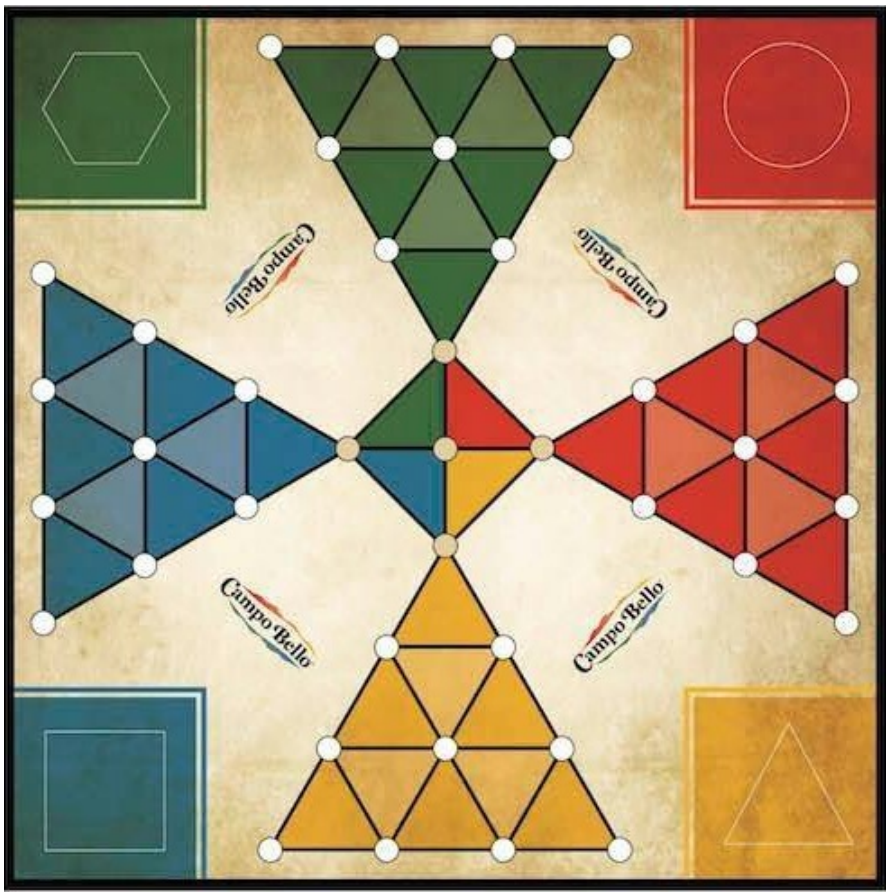
Conteúdo

1 Introdução

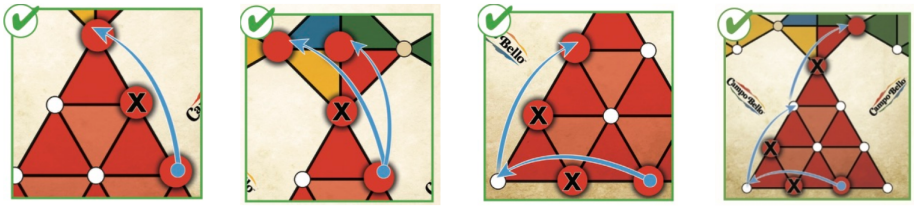
Descrever os objetivos e motivação do trabalho. Descrever num parágrafo breve a estrutura do relatório.

2 O Jogo Campo Bello

Campo Bello é um jogo que pode ser jogado de 2 a 4 jogadores. Tem inspiração no jogo clássico "Resta Um"ou "Peg Solitaire"em Inglês. É um jogo ainda recente, criado em 2017. Para ganhar, um jogador deve tentar remover todas as suas peças do tabuleiro antes dos adversários. O tabuleiro consiste em 4 triângulos que rodeiam um diamante central. Os triângulos correspodem às áreas iniciais de cada jogador. As peças são removidas ao saltar: saltar uma peça nossa causa a remoção da peça que foi saltada; saltar uma peça adversária permite-nos remover uma peça nossa do tabuleiro. Na variante de apenas 2 jogadores, que vai ser implementada, os jogadores ficam com triângulos opostos e jogam alternadamente. É também possível executar saltos duplos e triplos numa só jogada. No fim do jogo cada jogador pontua 1 ponto por cada uma das suas peças fora da área inicial e 3 pontos por cada uma das suas peças dentro da sua área inicial. O jogador com menos pontos ganha!



LEGAL MOVES



single jump | single jump | double jump | triple jump

ILLEGAL MOVES

end where you began | sharp corner jump | jump the same piece

http://www.campobellogame.com/

3 Lógica do Jogo

Descrever o projeto e implementação da lógica do jogo em Prolog, incluindo a forma de representação do estado do tabuleiro e sua visualização, execução de movimentos, verificação do cumprimento das regras do jogo, determinação do final do jogo e cálculo das jogadas a realizar pelo computador utilizando diversos níveis de jogo. Sugere-se a estruturação desta secção da seguinte forma:

3.1 Representação do Estado do Jogo

O estado do jogo é representado por 2 listas que identificam em que posições do tabuleiro estão as peças de cada jogador (1 lista para cada jogador). As peças de cada jogador sáo designadas em Inglês por "movers".

Tabuleiro no estado de jogo inicial:

```
YMovers=[y1,y2,y3,y4,y5,y6,y7,y8,y9,g1,g2,g3,g4,g5,g6,g7,g8,g9]
BMovers=[r1,r2,r3,r4,r5,r6,r7,r8,r9,b1,b2,b3,b4,b5,b6,b7,b8,b9]
```

Tabuleiro no estado de jogo intermédio:

```
YMovers=[g6,g2,y4,y5,y6,y7,y8,y9]
BMovers=[y0,b8,r1,r3,r4,r5,r6,r7,r8,r9]
```

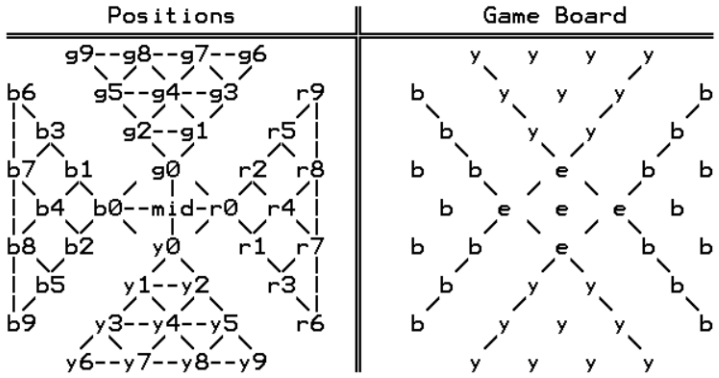
Tabuleiro no estado de jogo final:

```
YMovers=[g5,y5,y9,g6,g8,y6]
BMovers=[r5,r6,r8,y0,b8,b2,b3]
```

Com o decorrer do jogo as listas perdem elementos. O estado final do jogo é alcançado quando uma das listas estiver vazia ou quando não houver mais jogadas possíveis para o jogador que vai jogar.

3.2 Visualização do Tabuleiro

O predicado para visualização do tabuleiro mostra as posições do tabuleiro do lado esquerdo a usar para os movimentos e do lado direito o estado atual das peças. Usa *put_code* para melhorar o aspeto e facilitar a visualição.




```
displayBoard(Y,B) :-
        nl,
        write('       Positions        '), put_code(186), write('       Game Board'), nl,
        put_code(205),put_code(205),put_code(205),put_code(205),put_code(205),put_code(205),
        put_code(205),put_code(205),put_code(205),put_code(205),put_code(205),put_code(205),
        put_code(205),put_code(205),put_code(205),put_code(205),put_code(205),put_code(205),
        put_code(205),put_code(205),put_code(205),put_code(205),put_code(205),put_code(205),
        put_code(205),put_code(205),put_code(205),put_code(205),nl,
        displayLine(1,Y,B),
        displayLine(2,Y,B),
        displayLine(3,Y,B),
        displayLine(4,Y,B),
        displayLine(5,Y,B),
        displayLine(6,Y,B),
        displayLine(7,Y,B),
        displayLine(8,Y,B),
        displayLine(9,Y,B),
        nl,nl.
```

Estes predicados encarregam-se de imprimir qual a peça do tabuleiro está naquela posição.

```
displaySingleP(P, PiecesY, _) :- member(P,PiecesY),!, write(y), write('   ').
displaySingleP(P, _, PiecesB) :- member(P,PiecesB),!, write(b), write('   ').
displaySingleP(_,_,_) :- write(e), write('   ').
displayPos([H|T],PiecesY,PiecesB) :- displaySingleP(H,PiecesY,PiecesB),
                                     displayPos(T,PiecesY,PiecesB).
displayPos([],_,_).
```

O resto é específico de cada linha. Eis o exemplo de uma dessas linhas:

```
displayLine(1,PiecesY,PiecesB) :- write('    g9--g8--g7--g6      '),
       put_code(186) ,
       write('       '), displayPos([g9,g8,g7,g6],PiecesY,PiecesB),
       nl,
       write('      '), put_code(92), write(' / '),
       put_code(92), write(' / '), put_code(92), write(' /       '),
       put_code(186),
       write('        '), put_code(92), write('         /  '),
       nl.
```

3.3 Lista de Jogadas Válidas

Obtenção de uma lista de jogadas possíveis. Exemplo: *valid_moves(+Board, -ListOfMoves).*

3.4 Execução de Jogadas

Validação e execução de uma jogada num tabuleiro, obtendo o novo estado do jogo. Exemplo: *move(+Move, +Board, -NewBoard).*

3.5 Avaliação do Tabuleiro

Avaliação do estado do jogo, que permitirá comparar a aplicação das diversas jogadas disponíveis. Exemplo: *value(+Board, +Player, -Value).*

3.6 Final do Jogo

Verificação do fim do jogo, com identificação do vencedor. Exemplo: *game_over(+Board, -Winner).*

3.7 Jogada do Computador

Escolha da jogada a efetuar pelo computador, dependendo do nível de dificuldade. Por exemplo: *choose_move(+Level, +Board, -Move).*

4 Interface com o Utilizador

Descrever o módulo de interface com o utilizador em modo de texto.

5 Conclusões

Que conclui deste projecto? Como poderia melhorar o trabalho desenvolvido?

A Nome do Anexo

Código Prolog implementado devidamente comentado e outros elementos úteis que não sejam essenciais ao relatório.