Corrida de Reis

Relatório Intercalar

Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

Grupo Corrida_de_Reis_2:

Beatriz Henriques – up201502858

Beatriz Velho – up201700491

Faculdade de Engenharia da Universidade do Porto Rua Roberto Frias, sn,

4200-465 Porto, Portugal

15 Outubro 2017

Índice

Índice de Figuras

1 O jogo Corrida de Reis

Em 1961, Vernon Rylands Parton criou o jogo Corrida de Reis, uma variante do Xadrez. Sendo uma variante do Xadrez, a Corrida de Reis tem regras diferentes, bem como um objetivo diferente. É um jogo de tabuleiro 8x8 com um total de 64 casas, realizado por dois jogadores com peças de cor diferentes: preto e branco (Rachunek, 2017). Cada jogador possui 8 peças:

- Um rei;
- Uma rainha;
- Duas torres;
- Dois bispos;
- Dois cavalos.

No início do jogo, as peças são dispostas nas duas primeiras linhas do tabuleiro como apresentado na Figura 1, pelo que os dois jogadores têm a mesma vista do jogo (Rachunek, 2017).

O objetivo da Corrida de Reis é ser o primeiro a levar o próprio rei até à última linha do tabuleiro, isto é, antes do adversário. Para mover e capturar as peças são utilizadas as regras tradicionais do Xadrez, no entanto com a seguinte alteração:

- Não é permitido atacar o rei do jogador adversário, ou seja, não é possível fazer nenhuma jogada que coloque o rei adversário em xeque – posição onde pode ser capturado;

É importante relembrar que um rei não se pode pôr a si mesmo em xeque, como não pode mover-se para uma casa em que esteja uma peça adversária. É também importante relembrar que o cavalo é única peça que pode saltar sobre outras peças, adversárias ou não.

O jogo termina quando um jogador move o seu rei para a última linha do tabuleiro. Caso seja o rei branco a chegar primeiro à última linha e o rei preto consiga, na próxima jogada, mover-se também para a última linha, o jogo termina com um empate, isto porque o jogador com as peças brancas tem a vantagem de iniciar o jogo (Rachunek, 2017).

1.1 Imagens do Jogo Corrida do Reis

É apresentado nas figuras abaixo o estado do tabuleiro quando se inicia o jogo Corrida de Reis, seguido das figuras que representam os possíveis movimentos de cada peça, tendo em conta que não se encontram outras peças no tabuleiro, bem como figuras que representam possíveis fins de um jogo sem desistência.

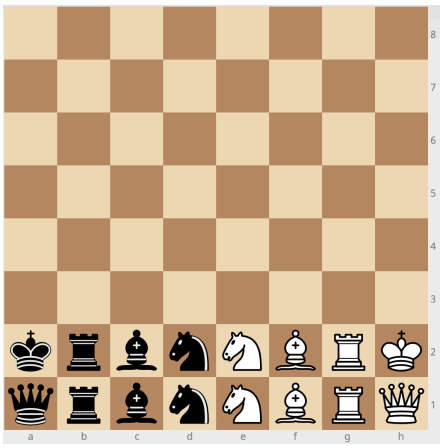


Figura 1: Estado inicial do tabuleiro

Fonte: (Lichess, 2017)

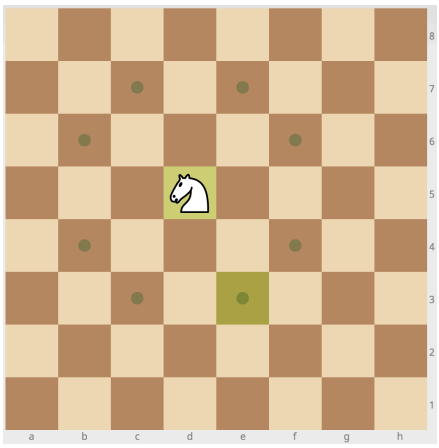


Figura 2: Possíveis jogadas do cavalo

Fonte: (Lichess, 2017)

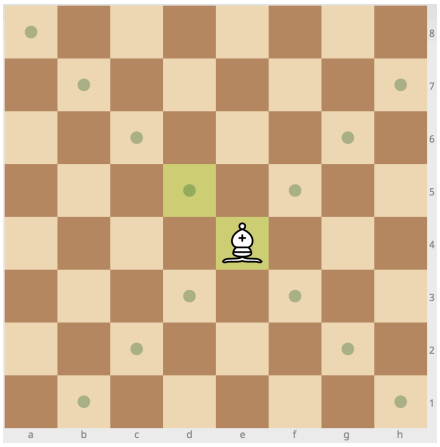


Figura 3: Possíveis jogadas do bispo

Fonte: (Lichess, 2017)

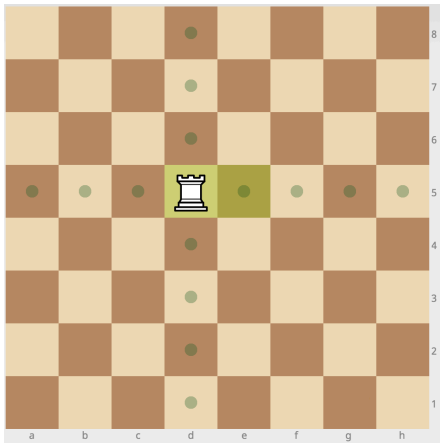


Figura 4: Possíveis jogadas da torre

Fonte: (Lichess, 2017)

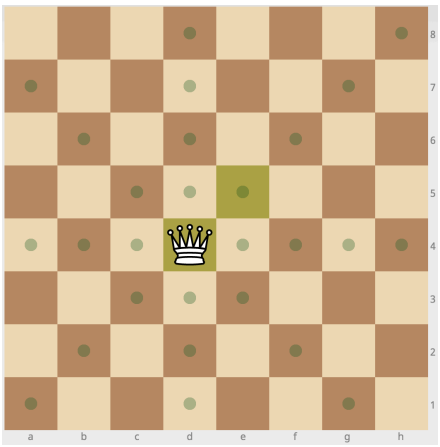


Figura 5: Possíveis jogadas da rainha

Fonte: (Lichess, 2017)

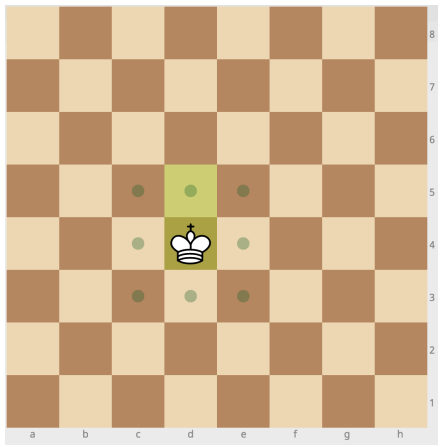


Figura 6: Possíveis jogadas do rei

Fonte: (Lichess, 2017)

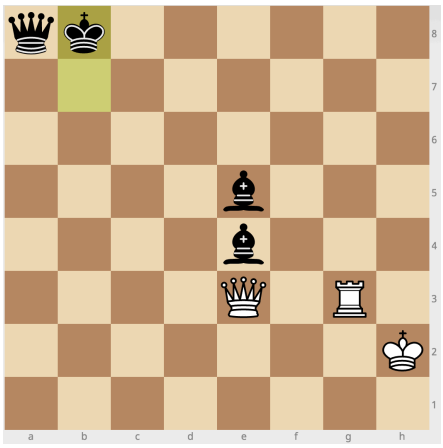


Figura 7: Jogo ganho pelas peças pretas

Fonte: (Lichess, 2017)

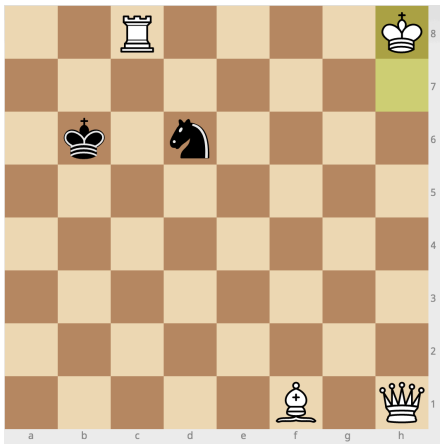


Figura 8: Jogo ganho pelas peças brancas

Fonte: (Lichess, 2017)

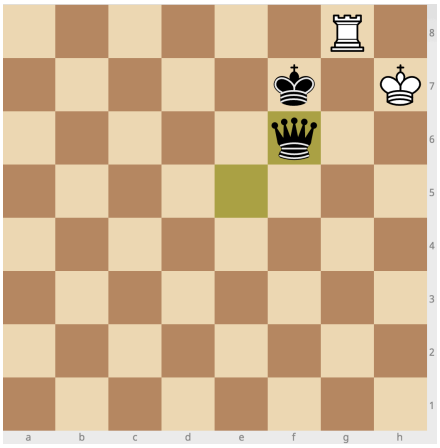


Figura 9: Jogo que irá acabar empatado

Fonte: (Lichess, 2017)

2 Abordagem inicial à modelação do jogo em Prolog

É descrito neste capítulo o método escolhido para a representação do estado do jogo, incluindo legenda do tabuleiro e exemplos de representações durante um jogo com imagens do tabuleiro impresso na consola. É também descrito o código utilizado para a impressão do tabuleiro na consola e por fim é apresentado os cabeçalhos dos predicados que serão implementados com a responsabilidade de validar e efetuar jogadas possíveis.

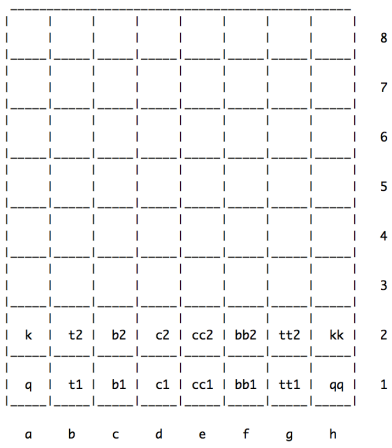
2.1 Representação do estado do jogo

Para representar o estado do jogo foi escolhida uma lista de listas. Cada lista representa uma linha do tabuleiro e tem oitos elementos que, por sua vez, representam as casas/colunas do tabuleiro. Uma casa pode ter um dos seguintes elementos:

- Casa vazia - (espaço vazio)
- k – Rei Preto
- q – Rainha Preta
- t1 – Torre Preta
- t2 – Torre Preta
- b1 – Bispo Preto
- b2 – Bispo Preto
- c1 – Cavalo Preto
- c2 - Cavalo Preto
- kk – Rei Branco
- qq – Rainha Branca
- tt1 – Torre Branca
- tt2 – Torre Branca
- bb1 – Bispo Branco
- bb2 – Bispo Branco
- cc1 – Cavalo Branco
- cc2 - Cavalo Branco

No tabuleiro só pode haver uma única peça das referidas acima, exceto a casa vazia.

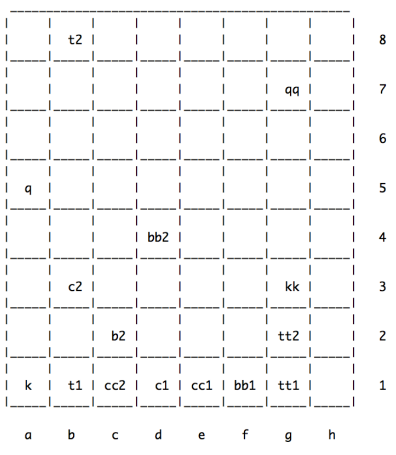
2.1.1 Lista representativa do estado inicial do jogo




```
[ [ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
  [ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
  [ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
  [ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
  [ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
  [ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
                [ k , t2 , b2 , c2 , cc2 , bb2 , tt2 , kk ],
                [ q , t1 , b1 , c1 , cc1 , bb1 , tt1 , qq ] ]
```

Figura 10: Estado inicial do tabuleiro apresentado na consola

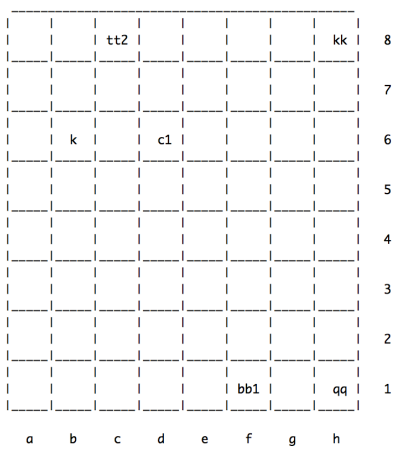
2.1.2 Lista representativa de um possível estado intermédio do jogo




```
[ [ vazio , t2 , vazio , vazio , vazio , vazio , vazio , vazio ],
  [ vazio , vazio , vazio , vazio , vazio , vazio , qq , vazio ],
[ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
  [ q , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
[ vazio , vazio , vazio , bb2 , vazio , vazio , vazio , vazio ],
  [ vazio , c2 , vazio , vazio , vazio , vazio , kk , vazio ],
[vazio , vazio , b2 , vazio , vazio , vazio , tt2 , vazio ],
  [ k , t1 , cc2 , c1 , cc1 , bb1 , tt1 , vazio ] ]
```

Figura 11: Possível estado intermédio do tabuleiro apresentado na consola

2.1.3 Lista representativa de um possível estado final do jogo




```
[ [ vazio , vazio , tt2 , vazio , vazio , vazio , vazio , kk ],
[ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
  [ vazio , k , vazio , c1 , vazio , vazio , vazio , vazio ],
[ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
[ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
[ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
[ vazio , vazio , vazio , vazio , vazio , vazio , vazio , vazio ],
  [ vazio , vazio , vazio , vazio , vazio , bb1, vazio , qq ] ]
```

Figura 12: Possível estado final do tabuleiro apresentado na consola

2.2 Visualização do tabuleiro em modo de texto

É apresentado no Excerto de Código 1 como se procede à inicialização do tabuleiro do jogo Corrida de Reis e no Excerto de Código 2 é apresentada a versão inicial do código que irá ser utilizado na impressão do tabuleiro com a representação do estado do jogo.

```
inicializarTabuleiro(
[[vazio,vazio,vazio,vazio,vazio,vazio,vazio,vazio],
[vazio,vazio,vazio,vazio,vazio,vazio,vazio,vazio],
[vazio,vazio,vazio,vazio,vazio,vazio,vazio,vazio],
[vazio,vazio,vazio,vazio,vazio,vazio,vazio,vazio],
[vazio,vazio,vazio,vazio,vazio,vazio,vazio,vazio],
[vazio,vazio,vazio,vazio,vazio,vazio,vazio,vazio],
[k , t2 , b2 , c2 , cc2, bb2 , tt2 , kk],
[q , t1 , b1 , c1 , cc1, bb1 , tt1 , qq]]).
```

Excerto de Código 1: Código para inicializar o tabuleiro do jogo Corrida de Reis

```
% Imprime as letras que permitem identificar uma coluna do tabuleiro
impimirIdentificadoresColunas:-
        write('   a     b     c     d     e     f     g     h').

% Lista com os numeros que permitem identificar uma linha do tabuleiro
numeroLinhas(['8','7','6','5','4','3','2','1']).

% Imprime o limite superior do tabuleiro
imprimirSeparadorInicial:-
        write(' _______________________________________________').

% Imprime o separador de linhas do tabuleiro
imprimirSeparadorLinhas:-
        write('|_____|_____|_____|_____|_____|_____|_____|_____|  ').

% Imprime o separador de colunas do tabuleiro
imprimirSeparadorColunas:-
        write('|     |     |     |     |     |     |     |     |').

% Imprime uma casa do tabuleiro com a peca "Peca"
imprimeCasa(Peca, L):-
        L = 1, write('  '), write(Peca), write('  ').
imprimeCasa(Peca, L):-
        L = 2, write('  '), write(Peca), write(' ').
imprimeCasa(Peca, L):-
        L = 3, write(' '), write(Peca), write(' ').
imprimeCasa(_, _):-
        write('     ').

% Imprime as pecas que estao numa determinada linha do tabuleiro
imprimirPecasLinha([]).
imprimirPecasLinha([H | T]):-
        atom_length(H, L), write('|'),
        imprimeCasa(H, L),
        imprimirPecasLinha(T).

% Imprime a linha numero "NLinha" do tabuleiro
imprimirLinha([] , []).
imprimirLinha(Linha , NLinha):-
        imprimirSeparadorColunas, nl,
        imprimirPecasLinha(Linha), write('|   '), write(NLinha), nl,
        imprimirSeparadorLinhas, nl.

% Imprime todas as linhas do tabuleiro
imprimirLinhas([],[]).
imprimirLinhas([Linha | T], [NLinha | ListaLinhas]):-
        imprimirLinha(Linha, NLinha),
        imprimirLinhas(T, ListaLinhas).

% Imprime as pecas capturadas por cada um dos jogadores
imprimirListaPecasCapturadas([]).
imprimirListaPecasCapturadas([H | T]):-
        write(H), write(' '),
        imprimirListaPecasCapturadas(T).

% Imprime uma lista de pecas capturadas pelo jogador "Jogador"
imprimirListaPecasCapturadas(Jogador, ListaPecas):-
        write('- Jogador '), write(Jogador), nl,
        imprimirListaPecasCapturadas(ListaPecas).
```

```
% Imprime as pecas capturadas pelos dois jogadores
imprimirPecasCapturadas:-
        pecasCapturadas(branco, Lista1),
        pecasCapturadas(preto, Lista2), nl,
        write(' ____________________'), nl,
write('|_Pecas_Capturadas_|'), nl,
        imprimirListaPecasCapturadas(branco, Lista1), nl, nl,
        imprimirListaPecasCapturadas(preto, Lista2).

% Imprime o tabuleiro com o estado atual do jogo
imprimirTabuleiro([H | T]):-
        imprimirSeparadorInicial, nl,
        numeroLinhas(ListaLinhas),
        imprimirLinhas([H | T], ListaLinhas), nl,
        impimirIdentificadoresColunas, nl,
        imprimirPecasCapturadas, nl, nl.
```

Excerto de Código 2: Código para a impressão do tabuleiro com o estado do jogo

A Figura 10, Figura 11 e Figura 12 são três resultados diferentes do *output* do predicado `imprimirTabuleiro` representado no Excerto de Código 2, tendo como argumento a lista junto a cada figura.

O predicado `imprimirPecasCapturadas` tem como função imprimir as peças que ambos os jogadores capturaram até ao momento em que o tabuleiro é impresso. Na figura seguinte é ilustrado um exemplo:

```
 _________________________________________
|     |     |     |     |     |     |     |     |
|     |     | tt2 |     |     |     |  kk |   8
|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |
|     |     |     |     |     |     |     |   7
|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |
|     |  k  |     |  c1 |     |     |     |   6
|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |
|     |     |     |     |     |     |     |   5
|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |
|     |     |     |     |     |     |     |   4
|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |
|     |     |     |     |     |     |     |   3
|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |
|     |     |     |     |     |     |     |   2
|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |
|     |     |     |     |     | bb1 |     |  qq |   1
|_____|_____|_____|_____|_____|_____|_____|_____|

   a     b     c     d     e     f     g     h

 __________________
|_Pecas_Capturadas_|
- Jogador branco
q t1 t2 b1 b2 c2

- Jogador preto
tt1 bb2 cc1 cc2
```

Figura 13: Possível estado de jogo com representação das peças capturadas

2.3 Cabeçalhos dos predicados a serem implementados

Numa fase posterior irão ser implementados os seguintes predicados:

- moverPeca(Jogador, Peca, Linha, Coluna, LinhaDest, ColunaDest, Tabuleiro)

Predicado que irá mover uma peça de um jogador da posição atual para uma posição destino. Este predicado é executado depois de o movimento ser validado.

- capturarPeca(Jogador, Peca, Linha, Coluna, Tabuleiro)

Predicado que, depois de um movimento de uma peça para uma casa que estava ocupada, coloca a peça anterior na lista de peças capturadas pelo jogador.

- validarJogada(Jogador, Peca, Linha, Coluna, LinhaDest, ColunaDest, Tabuleiro)

Predicado que irá validar o movimento da peça desde a casa onde se encontra até à casa destino. Depois de ser validado, o movimento pode ser efetuado ou pode ser negado, sendo neste caso pedida um novo movimento do jogador.

3 Bibliografia

Lichess. (2017). Racing Kings. Retrieved from https://lichess.org/study/7qOrZwG6

Rachunek, F. (2017). BrainKing - Regras do jogo (Corrida de Reis). Retrieved from https://brainking.com/pt/GameRules?tp=125