

Faculdade de Engenharia da Universidade do Porto



Martian Chess

Programação Lógica 2016/2017

Mestrado Integrado em Engenharia Informática e Computação

Grupo 1:

Tiago José Grosso Pacheco - 201402722 - up201402722@fe.up.pt

Vasco Ferreira Ribeiro - 201402723 - up201402723@fe.up.pt

16 de outubro de 2016

1. O Jogo Martian Chess

O jogo de estratégia Martian Chess foi desenhado por Andrew Looney, um dos inventores das pirâmides de *Icehouse*. As primeiras versões das regras do jogo datam de 29 de janeiro de 1995, sendo, assim, considerado o segundo jogo de pirâmides mais antigo, logo a seguir ao famoso *Icehouse*.

Martian Chess é um jogo de tabuleiro parecido com o Xadrez, uma vez que se joga num tabuleiro de Xadrez, temos vários tipos de peças e o objetivo é capturar as peças do adversário. No entanto, o controlo das peças é determinado pela localização, que determina quais as peças se pode mover. Cada jogador tem um quadrante de jogo, podendo apenas mover as peças presentes ao quadrante que lhe pertence e atacar as presentes nas outras zonas. O objetivo do jogo é fazer mais pontos (cada peça capturada dá pontos – Rainha: 3 pontos; Drone: 2 pontos; Pião: 1 ponto).

O jogo pode ser jogado por 2 ou 4 jogadores. Para efeitos deste trabalho, iremos abordar somente a variante de 2 jogadores.

O tabuleiro tem uma posição inicial de jogo, com as peças dispostas em cantos opostos e com os Piões dispostos na frente, seguidos dos Drones e posteriormente das Rainhas.

Cada Peça tem um tipo de movimento; contudo, saltar por cima de peças não é permitido:

- Rainha – Pode mover-se em todas as direções o número de casas que pretender;
- Drone – Pode mover-se 1 ou 2 casas na horizontal ou na vertical;
- Pião – Pode mover-se 1 casa na diagonal.

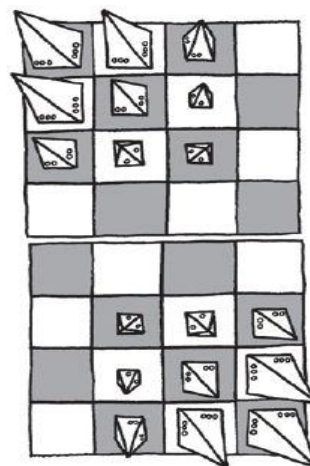


Figura 1 - Posição Inicial do Jogo

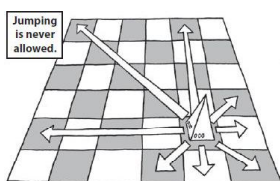


Figura 2 - Movimento Rainha

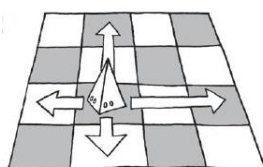


Figura 4 - Movimento Drone

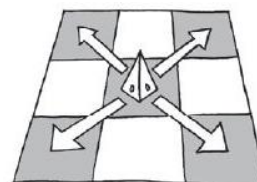


Figura 3 - Movimento Pião

O jogo tem, no entanto, a possibilidade de, durante o jogo, fazer algumas promoções. Isto é, quando um jogador não possuir rainhas, ele poderá criar uma, movendo um Drone para uma casa onde esteja um Pião, fundindo-os. O mesmo pode acontecer juntando 2 Piões para dar origem a um Drone (quando não possuir nenhum Drone no seu quadrante).

O jogo termina quando um dos quadrantes estiver totalmente vazio.

Fontes:

- <http://www.looneylabs.com/rules/martian-chess;>
- http://icehousegames.org/wiki/index.php?title=Martian_Chess;

2. Representação do Estado do Jogo

A representação do jogo será feita com uma lista de listas, em que cada elemento de uma lista (um caracter) representa uma das posições do tabuleiro. A lista “Board” terá, assim, várias listas, cada uma delas representativa de uma das linhas do tabuleiro:

```
board([[r,r,d,v],
      [r,d,p,v],
      [d,p,p,v],
      [v,v,v,v],
      [v,v,v,v],
      [v,p,p,d],
      [v,p,d,r],
      [v,d,r,r]
      ]).
```

| | | | |
|---|---|---|---|
| r | r | d | v |
| r | d | p | v |
| d | p | p | v |
| v | v | v | v |
| v | v | v | v |
| v | p | p | d |
| v | p | d | r |
| v | d | r | r |

Figura 5 - Display do Board

3. Visualização do Tabuleiro

Para permitir a visualização do Tabuleiro, foram implementadas algumas funções em Prolog.

Primeiro chamamos a função `display_board()`. Esta função escreve, de forma recursiva, cada linha, invocando a função `display_line()` para cada lista dentro da lista `board` (lista de listas). Entre cada linha, vai-se colocando hífen de forma a tornar mais legível o tabuleiro.

A função `display_line()` por sua vez, recebe uma lista (cada lista será uma das linha do board) e imprime todos os elementos, colocando hífen verticais entre cada um deles por uma questão de legibilidade e aparência.

```
display_board([L1|Ls]):-
    write(' |-----|'), nl,
    write(' | '), display_line(L1), nl,
    display_board(Ls).
|
display_board([]):-
    write(' |-----|'), nl.

display_line([E1|Es]):-
    write(E1), write(' | '),
    display_line(Es).
display_line([]).
```

Figura 6 - Funções `display_board` e `display_line`

4. Movimentos

Para permitir os movimentos, criamos uma função `move(Xi, Yi, Xf, Yf)`. Esta função irá receber como parâmetros `Xi` e `Yi`, coordenadas da posição inicial da peça que o jogador pretende mover e `Xf` e `Yf`, posições finais da peça. No entanto, o movimento poderá não ser possível e a função terá também de ver qual a peça que se encontra nas posições iniciais e, posteriormente, verificar se o movimento é possível. Em caso afirmativo prossegue-se com o movimento e em caso negativo avisa-se o jogador que o movimento pretendido é inválido e pede para especificar outra jogada.