

Moai

Relatório Intercalar



Universidade do Porto

Faculdade de Engenharia

FEUP

Mestrado Integrado em Engenharia Informática e
Computação

Programação em Lógica

Grupo 5:

Paulo Jorge de Faria dos Reis - 080509037

Miguel Rossi Seabra - 060509054

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

7 de Outubro de 2012

Resumo

Pretende-se com este relatório apresentar o procedimento seguido pelo grupo de trabalho no desenvolvimento em Prolog do jogo "Moai".

Será disponibilizada informação relativamente ao jogo em termos de regras, descrição do tabuleiro, opções tomadas pelo grupo em pontos onde a informação do criador do daquele não é clara e também uma apresentação dos predicados de Prolog que vão ser utilizados para o desenvolvimento do projeto.

1 Introdução

A implementação das regras de um jogo de tabuleiro recente, utilizando programação em lógica apresenta um desafio aliciante, ao mesmo tempo que possibilita o contacto e desenvolvimento de capacidades nesta área aos elementos envolvidos no desenvolvimento deste sistema. É também de ressaltar a criação de um programa funcional que permite a realização de partidas de "Moai" onde os intervenientes podem ser humanos ou computadores, e neste caso tendo o sistema capacidades mínimas de realizar jogadas válidas de acordo com as regras. A versão base apenas permitirá a interação em modo de texto através da consola de Prolog, podendo no entanto ser providenciada a integração com interface gráfico.

2 Descrição do Problema

O jogo foi criado por Rey Alicea em 2012 [2], e não existe qualquer referência histórica deste. Pelo que nos apercebemos o jogo encontra-se ainda num estágio de revisão do mesmo. A descrição das regras [1, 2] não são claras o suficiente, deixando muito espaço para a interpretação das mesmas, assim a equipa de desenvolvimento das regras em Prolog decidiu utilizar as regras conforme se explica mais adiante.

São necessários dois jogadores para realizar uma partida de "Moai".

2.1 Elementos do jogo

Para jogar "Moai" é necessário um tabuleiro com 8 filas e 8 colunas não se utilizando os cantos daquele, num total de 60 casas de jogo, ou em alternativa um tabuleiro hexagonal com lados de 6 casas de dimensão, para um total de 91 casas de jogo. Cada jogador é representado no tabuleiro por uma peça, "Moai", branca para o primeiro a jogar, preta para o segundo, para evitar confusões com o nome do jogo, passaremos a chamar estas peças de peões. São ainda utilizados um conjunto de peças que chamaremos de "bloqueadores". No caso do tabuleiro quadrado existem 4 direções de movimento de peões (num total de 8 sentidos de movimento), no entanto pela forma como é criado o tabuleiro hexagonal, neste apenas existem 3 direções (6 sentidos) de movimento, no tabuleiro hexagonal não existe movimento por colunas.

Por defeito iremos utilizar o tabuleiro quadrado.

Peão ("Moai") - Peça móvel, que representa o jogador no tabuleiro. Existem duas, uma para cada jogador, podendo sendo distinguidas pela sua cor (branca e preta). Em cada jogada apenas um peão é movido, de acordo com regras. No entanto o peão do jogador cuja jogada está em curso tem de ser

capaz de se mover de acordo com as regras, mesmo não sendo esse o movimento pretendido pelo jogador.

Bloqueador - Peça sem movimento no tabuleiro, cada jogador coloca uma por jogada numa qualquer casa de jogo de acordo com as regras especificadas mais adiante.

2.2 Regras

As regras que serão utilizadas são descritas de seguida:

- Cada casa de jogo pode ter nenhuma ou uma peça de jogo.
- O tabuleiro começa vazio e o jogador branco coloca o seu peão numa qualquer casa de jogo, segue-se o jogador preto a colocar o seu peão.
- O jogo termina quando na sua vez um jogador não for capaz de mover o seu próprio peão de acordo com as regras.
- Os bloqueadores uma vez colocados no tabuleiro ficam fixos, não sendo permitida qualquer movimentação dos mesmos, nem a sua remoção.
- Começando no jogador branco os jogadores vão alternado entre si, realizando uma jogada de cada vez, até o jogo terminar.
- Uma jogada consiste em:
 - Colocação de um bloqueador numa casa de jogo, garantindo que:
 - * O bloqueador tem de ficar em linha com um dos peões, diagonal ou ortogonalmente.
 - * Deve existir pelo menos uma casa de jogo vaga entre o peão e o bloqueador.
 - Mover um peão (o seu ou o do outro jogador) na direção do bloqueador colocado anteriormente, verificando:
 - * O peão move-se ao longo da linha diagonal ou ortogonal que une as duas peças.
 - * O peão move-se no sentido do bloqueador ou no sentido oposto a este, até encontrar um obstáculo (peão, bloqueador ou fronteira do tabuleiro).

A representação do tabuleiro seguinte não espelha uma situação real de jogo. O peão branco pode ser movimentado em qualquer um dos sentidos das setas, dependendo de qual o último bloqueador que foi colocado no tabuleiro.

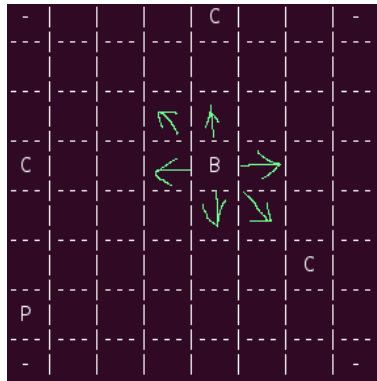


Figura 1: Tabuleiro com Movimentos possíveis

3 Representação do Estado do Jogo

A forma escolhida para representar o tabuleiro de jogo ao longo das jogadas consiste na utilização de uma lista de listas, Estas últimas representam cada uma delas uma linha do tabuleiro. O índice de cada uma das posições destas listas está relacionada com a uma coluna específica do tabuleiro.

```
[
  ['-',' ',' ',' ',' ',' ',' ','C',' ',' ',' ','-'],
  [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
  [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
  ['C',' ',' ',' ',' ',' ','B',' ',' ',' ',' '],
  [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
  [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '],
  ['P',' ',' ',' ',' ',' ',' ',' ',' ','C',' ',' '],
  ['-',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','-']
]
```

Figura 2: Representação do tabuleiro

4 Representação de um Movimento

Um movimento é representado fornecendo um tabuleiro, a posição onde pretendemos colocar o bloqueador, qual o peão a mover, e o sentido do movimento, como resultado recebemos o novo tabuleiro:

```
mover(InBoard, XCounter, YCounter, Cor, PushPull, OutBoard).
```

5 Visualização do Tabuleiro

Em modo de texto o tabuleiro, em particular a divisão das casas de jogo vai ser representado com recurso ao caracteres "-" e "|". O peão do jogador branco pelo carácter "B" e o do jogador preto com "P", os bloqueadores são representados com "C".

O código está disponibilizado no final deste documento no Anexo A.
 Por defeito considera-se um tabuleiro quadrado de dimensão de 8 por 8 casas, sendo que os cantos não são utilizados para efeitos de jogo.

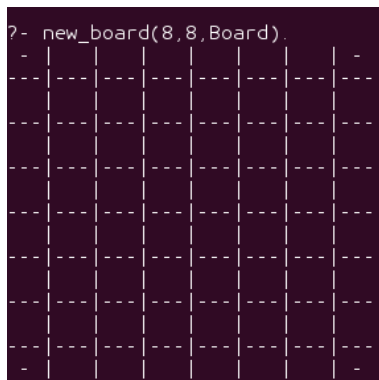


Figura 3: Tabuleiro vazio

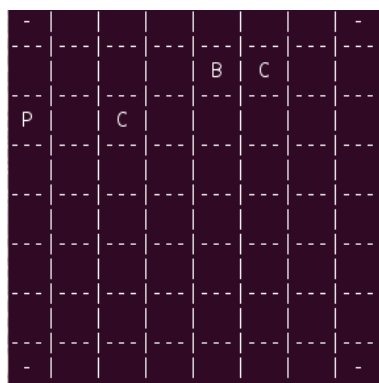


Figura 4: Tabuleiro durante o jogo

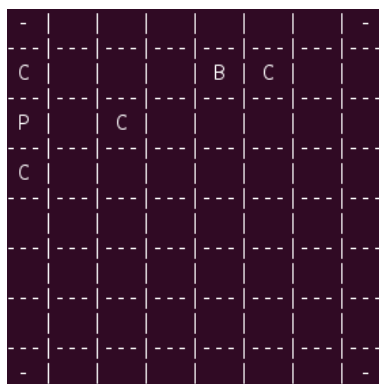


Figura 5: Tabuleiro no final do jogo

6 Conclusões e Perspetivas de Desenvolvimento

As regras do jogo não parecem ser complicadas, no entanto a forma como são descritas não são explícitas o suficiente. Sendo um jogo criado recentemente não existem quase referências ao mesmo excetuando as apresentadas na sitografia e que são da autoria do próprio autor. O trabalho vai consistir na elaboração dos predicados apresentados neste documento e outros que possam vir a ser necessários e que não tenham sido previstos. Consideramos que esteja feito uma décima parte do trabalho total necessário para a conclusão do projeto.

Bibliografia

- [1] Rey Alicea. Blog do autor. <http://reyaliceagames.wordpress.com/>, 2012. Online em Outubro 2012.
- [2] Rey Alicea. Boardgamegeek. <http://boardgamegeek.com/image/1405108/moai>, 2012. Online em Outubro 2012.

Lista de Figuras

1	Tabuleiro com Movimentos possíveis	4
2	Representação do tabuleiro	4
3	Tabuleiro vazio	5
4	Tabuleiro durante o jogo	5
5	Tabuleiro no final do jogo	5

A Anexo A - Código implementado

```
%Create a new board
new_board( X, Y, Board ):- create_board(X,Y,IBoard), blank_corners(IBoard,Board),
                           print_board(Board).

create_board( _, 0, FLINES ):- FLINES = [], !.
create_board( X, Y, [Line | Lines] ):- YN is Y-1, create_line(X,0, Line),
                                       create_board(X, YN, Lines).
create_line( X, X, L ):- L=[], !.
create_line( X, LX, [L | Ls] ):- LLX is LX+1, L = ' ', create_line( X, LLX, Ls).

blank_corners([Iline|Ilines],Board ):- blank_first_corners(Iline,Oline),
                                       blank_last_corners(Oline, Ilines, Board).

blank_first_corners( [_|IT], [OH|OT] ):- OH = '- ', blank_corner(IT,OT).

blank_corner( [_|[]], [OH|[]] ):- OH = '- ', !.
blank_corner( [IH|IT] , [IH|OT] ):- blank_corner(IT,OT).

blank_last_corners(Oline, Ilines, [Oline|OT] ):- blank_last(Ilines,OT).

blank_last( [IH|[]], [OH|[]] ):- blank_first_corners(IH,OH), !.
blank_last( [IH|IT], [IH|OT] ):- blank_last(IT,OT).

print_board( [LastLine | [] ] ):- print_line( LastLine), write('\n\n'), !.
print_board( [Line | Lines ] ):- print_line( Line), print_separator(Line),
                                print_board(Lines).

print_line( [LastPiece | [] ] ):- write(' '), write(LastPiece), write('\n'), !.
print_line( [Piece | Pieces ] ):- write(' '), write(Piece), write(' |'),
                                print_line( Pieces ).

print_separator( [_|[]] ):- write('---\n'), !.
print_separator( [_|T] ):- write('---|'), print_separator(T).
```