

Oolong

Relatório Intercalar



Mestrado Integrado em Engenharia Informática e
Computação

Programação em Lógica

Grupo O_1:

José Pedro Borges - up201503603
Miguel Mano Fernandes - up201503538

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

14 de Outubro de 2017

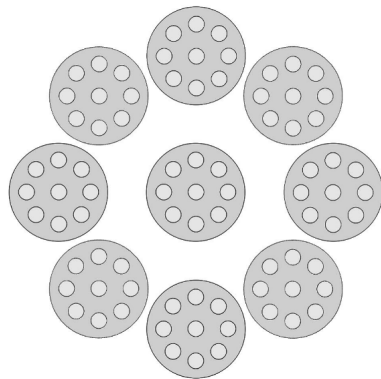
1 O Jogo Oolong

Oolong é um jogo de tabuleiro desenvolvido em 2015 pelo artista John Shulters e designer Sarah Graybill e publicado por Black Straw Games.

Trata-se de um jogo de estratégia para dois jogadores situado numa casa de chá japonesa, em que cada jogador representa um fabricante de chá (preto e verde) tentando servir o máximo da sua marca.

Quando um jogador servir 5 porções numa mesa, haverá atingido a maioria nessa mesa. Vencer o jogo envolve atingir a maioria em 5 mesas e, consequentemente, na casa.

A organização das 9 mesas deverá seguir a seguinte estrutura:



Como cada uma das 9 mesas tem 9 posições para jogar será mais fácil imaginar uma bússola por cima do tabuleiro em que cada lugar na mesa e cada mesa representam uma direção (N, NE, E, SE, S, (...) e centro).

1.1 Regras do Jogo

1.1.1 Começar o jogo

Quem começa o jogo é o jogador com o chá preto e tem de colocar uma das peças dele na mesa do centro, em qualquer posição. Se os jogadores estiverem envolvidos em múltiplos jogos, quem começa é quem perdeu o último. Cada jogada envolve colocar uma peça, mover o empregado e, possivelmente, ativar uma Ação Especial.

1.1.2 Colocação de peças

O lugar em que a peça é colocada indica a mesa em que se vai jogar a seguir. Por exemplo, se o jogador Verde jogar no lugar SE da mesa do centro, o jogador Preto tem de jogar na mesa SE do tabuleiro, em qualquer posição, desde que não esteja já ocupada.

1.1.3 Mover o empregado

O empregado é utilizado para ajudar a saber em que mesa será colocada a próxima peça e a sua posição prévia. As regras de utilização são as seguintes:

- Quando é jogada uma peça, o empregado é movido para a mesa onde vai ser feita a próxima jogada.
- Ao colocar o empregado é preciso ter o cuidado de o pôr na peça que represente a mesa onde foi jogado o turno anterior.

Por exemplo, se o jogador Preto está na mesa do centro e joga na posição W, é preciso colocar o empregado na mesa W e na posição do centro desta.

1.2 Marcadores especiais

Existem 8 marcadores especiais que são designados aleatoriamente às mesas na periferia - cada mesa tem um no máximo.

Cada marcador é ativado imediatamente mal o requisito que exigido seja cumprido. Após a ação ter sido feita não poderá voltar a ser usado até ao final do jogo. Se uma ação ativar outra ação noutra mesa, esta também será executada, ou seja, é permitido encadeamento de ações especiais.

- O jogador representado pode mover uma das suas peças de uma mesa não conquistada para outra mesa não conquistada.
Requisito: 3 peças iguais.



- O jogador representado pode mover o empregado do espaço onde está para o mesmo espaço numa mesa diferente.
Requisito: 5 peças iguais.



- O jogador pode rodar a mesa para uma orientação qualquer (empregado também roda).
Requisito: 4 peças iguais.



- O jogador que ativa a ação pode trocar a posição de duas mesas não conquistadas. O empregado também é movido, se presente.
Requisito: 4 peças iguais.



- O jogador que ativa a ação pode trocar a posição de uma mesa conquistada com uma não conquistada. O empregado também é movido, se presente.
Requisito: 5 peças iguais.



1.3 Conquistar uma mesa

Quando um jogador tiver 5 peças da sua cor numa mesa, conquista essa mesa. A mesa pode continuar a ser utilizada nas jogadas, mas quando todos os espaços vazios forem preenchidos a mesa será considerada completa e qualquer jogada que levaria um jogador a ir para essa mesa será substituída, fazendo com que o jogador possa escolher um lugar qualquer vazio para colocar a sua peça.

1.4 Fim do Jogo

Quando um jogador conquista 5 das 9 mesas o jogo acaba imediatamente.

2 Representação do Estado do Jogo

A representação do estado do jogo não se poderá cingir apenas ao armazenamento do tabuleiro, dada a existência de elementos exteriores dinâmicos - as cartas especiais.

Para tal, considerou-se favorável a utilização de uma estrutura de dados adicional - uma lista que correlaciona os marcadores especiais e as mesas.

De facto, seria possível a reserva de um elemento extra na representação de cada mesa, mas por motivos de simplificação e facilidade de impressão do tabuleiro, dividiu-se em dois arrays distintos.

2.1 Tabuleiro

O tabuleiro é uma matriz multidimensional que preserva as 9 mesas redondas, mapeadas de forma similar ao teclado numérico de um telemóvel:

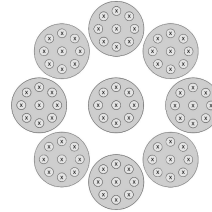
- A **1ª posição** da matriz corresponde à mesa **Noroeste** (NW).
- A **2ª posição** da matriz corresponde à mesa **Norte** (N).
- A **4ª posição** da matriz corresponde à mesa **Oeste** (O).
- ...

A correspondência de cada posição da mesa a um ponto cardeal segue o mesmo padrão.

2.1.1 Representação inicial do tabuleiro

No estado inicial, todas as células estão vazias - representadas por 'x'.

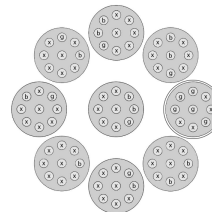
```
[ [ x, x, x, x, x, x, x, x, x ],
  [ x, x, x, x, x, x, x, x, x ],
  [ x, x, x, x, x, x, x, x, x ],
  [ x, x, x, x, x, x, x, x, x ],
  [ x, x, x, x, x, x, x, x, x ],
  [ x, x, x, x, x, x, x, x, x ],
  [ x, x, x, x, x, x, x, x, x ],
  [ x, x, x, x, x, x, x, x, x ],
  [ x, x, x, x, x, x, x, x, x ] ]
```



2.1.2 Representação intermédia do tabuleiro

No estado intermédio, surge o aparecimento de peças pretas - representadas por 'b' - e peças verdes - representadas por 'g'.

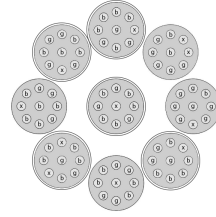
```
[ [ x, g, x, x, x, b, x, x, x ],
  [ b, x, x, b, x, x, g, x, x ],
  [ x, b, x, x, b, x, x, g, x ],
  [ b, x, g, x, x, x, x, x, x ],
  [ x, x, b, x, x, b, x, x, g ],
  [ g, g, x, g, g, x, x, x, g ],
  [ x, x, x, x, x, b, x, x, x ],
  [ x, x, g, x, x, b, x, x, x ],
  [ x, x, x, x, x, b, x, b, x ] ]
```



Neste estado em específico ilustrou-se uma situação de vitória de mesa por parte do jogador verde. Note-se que a mesa Este - posição nº6 da matriz - acumula um total de 5 peças verdes, pelo que a maioria nesse ponto está estabelecida.

2.1.3 Representação final do tabuleiro

```
[ [ g , g , b , b , b , b , g , g , x ] ,
  [ b , b , b , b , g , x , g , g , b ] ,
  [ g , b , x , g , b , x , g , g , g ] ,
  [ b , g , g , x , b , b , b , b , g ] ,
  [ b , g , b , g , x , b , b , b , g ] ,
  [ g , g , b , g , g , g , g , x , g ] ,
  [ b , x , b , b , g , b , x , b , g ] ,
  [ b , g , g , b , x , b , g , g , b ] ,
  [ g , b , x , g , g , b , b , b , b ] ]
```



Nesta situação final, o jogador preto emerge vitorioso, tendo conquistado maioria nas cinco mesas Norte, Oeste, central, Sudoeste e Sudeste.

2.2 Marcadores especiais

Dada a atribuição aleatória dos marcadores especiais no início do jogo e a possibilidade do jogo terminar sem todos os marcadores serem ativados, apresentar-se-á apenas uma possível disposição inicial e intermédia/final.

O mapeamento continua a executar-se do mesmo método descrito anteriormente.

2.2.1 Representação inicial dos marcadores

```
[ black_move , green_move , black_waiter , green_waiter ,
  empty , rotate , rotate , swap_unclaimed , swap_claimed ]
```

Neste exemplo, o marcador **black_move** está associado à mesa Noroeste.

Note-se a repetição da carta especial **rotate**, que efetivamente possui dois exemplares, e também a utilização da etiqueta **empty** para identificar mesas sem marcadores - ora consumidos, ora não atribuídos, como no caso da mesa central.

2.2.2 Representação intermédia/final dos marcadores

```
[ black_move , empty , black_waiter , green_waiter ,
  empty , rotate , rotate , empty , swap_claimed ]
```

Nesta etapa do jogo, os marcadores **green_move** e **swap_unclaimed** terão sido consumidos, daí a sua substituição por **empty**.

3 Visualização do Tabuleiro

A representação do tabuleiro em modo de texto não é feita com a disposição original do jogo para efeitos de simplificação. Em vez disso, é usada uma matriz de 3x3 mesas com 3x3 posições cada uma. As posições representam na mesma os pontos cardeais, para facilitar o jogo.

```

! ?- play.
Welcome to Prolog.



|   |   |   |
|---|---|---|
| x | x | x |
| x | x | x |
| x | x | x |



|   |   |   |
|---|---|---|
| x | x | x |
| x | x | x |
| x | x | x |



|   |   |   |
|---|---|---|
| x | x | x |
| x | x | x |
| x | x | x |



|   |   |   |
|---|---|---|
| x | x | x |
| x | x | x |
| x | x | x |



|   |   |   |
|---|---|---|
| x | x | x |
| x | x | x |
| x | x | x |



|   |   |   |
|---|---|---|
| x | x | x |
| x | x | x |
| x | x | x |



|   |   |   |
|---|---|---|
| x | x | x |
| x | x | x |
| x | x | x |



|   |   |   |
|---|---|---|
| x | x | x |
| x | x | x |
| x | x | x |



|   |   |   |
|---|---|---|
| x | x | x |
| x | x | x |
| x | x | x |


```

Para produzir este efeito é usada o predicado `print_formatted_line(X)` para desenhar as linhas horizontais em cada matriz, que separam as posições. É usada várias vezes o predicado `put_code` com os códigos ASCII 185, 187, 188, 201, 204, 205 e 206. X pode levar os argumentos 0, 1,2 ou 3 dependendo da linha a imprimir. Um dos usos do predicado é o seguinte:

```

print_formatted_line(X) :- X = 1,
    nl, put_code(204), put_code(205), put_code(205), put_code(205),
    put_code(206), put_code(205), put_code(205),
    put_code(205), put_code(206), put_code(205), put_code(205),
    put_code(205), put_code(185), nl.

```

O predicado `print_block([H|T])` junta o predicado acima com a impressão do array previamente definido. São usadas os predicados `nh0`, `write`, `put_code` e `print_formatted_line` várias vezes para cada matriz permitindo desenhar o tabuleiro da forma que queremos.

```

print_block([H|T]) :- nl, print_formatted_line(0), write(' '),
    print_formatted_line(0), write(' '), print_formatted_line(0), nl,
    put_code(186), write(' '), nth0(0, H, E1), write(E1), write(' '),
    put_code(186), (.....).

```

O predicado `print_board([H|T])` é o chamado para imprimir tudo feito pelos predicados mencionados anteriormente.

```

print_board([H|T]) :- length([H|T], Matrix_Size),
    Trim_Size is Matrix_Size - 3,
    trim_tail([H|T], Trim_Size, Block),
    print_block(Block),
    trim_head(T, 2, Remain),
    print_board(Remain).

```

`trim_head(L,N,S)` remove N elementos da lista L e guarda o resultado em S. `trim_tail(L,N,S)` faz o mesmo, só que em vez de cortar N elementos do principio, corta do fim

4 Movimentos

Cada turno, o jogador tem acesso a dois tipos de jogadas distintos:

- Colocação de uma peça em Board, enviando a posição de eleição numa lista cuja cabeça **H** corresponde à mesa e a cauda **T** à cadeira.

`place_token([H|T], Board)`

- Caso os requerimentos para a ativação de um marcador especial se cumpram, a seleção de uma mesa **Table** do **Board** e ativação da sua habilidade é automática.

`activate_special(Table, Board)`

Note-se, porém, que o input adicional exigido ao jogador varia consoante o marcador selecionado.

Por exemplo, o marcador **rotate** permite a rotação da mesa numa direção no sentido dos ponteiros do relógio ou inverso. Para tal, um possível predicado seria `rotate_table(Direction, Table, Board)`.

Traduzindo, a direção **Direction** - introduzida pelo jogador - da rotação da mesa **Table**, adquirida através do predicado **activate_special**. Estas alterações manifestam-se em **Board**.

Com o desenvolvimento do projeto e avanço da lógica de jogo, mais predicados deste género serão desenvolvidos.