

Relatório Intercalar

Azacru

Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

Grupo:

Nuno Miguel Outeiro Pereira – up201506265

Ricardo José Santos Pereira – up201503716

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

14 de outubro de 2017

1 História

O Azacru é um jogo que foi lançado em 2005 como parte integrante de um conjunto de jogos chamado Pacru Series 302 que continha vários jogos que partilhavam o mesmo tabuleiro, assim como as mesmas peças e marcos, conjunto este inventado por Mike Wellman.

Este jogo tem também o seu mestre, que neste caso é Martyn J.Hamer que já foi múltiplas vezes campeão do mundo.

2 Regras do Jogo

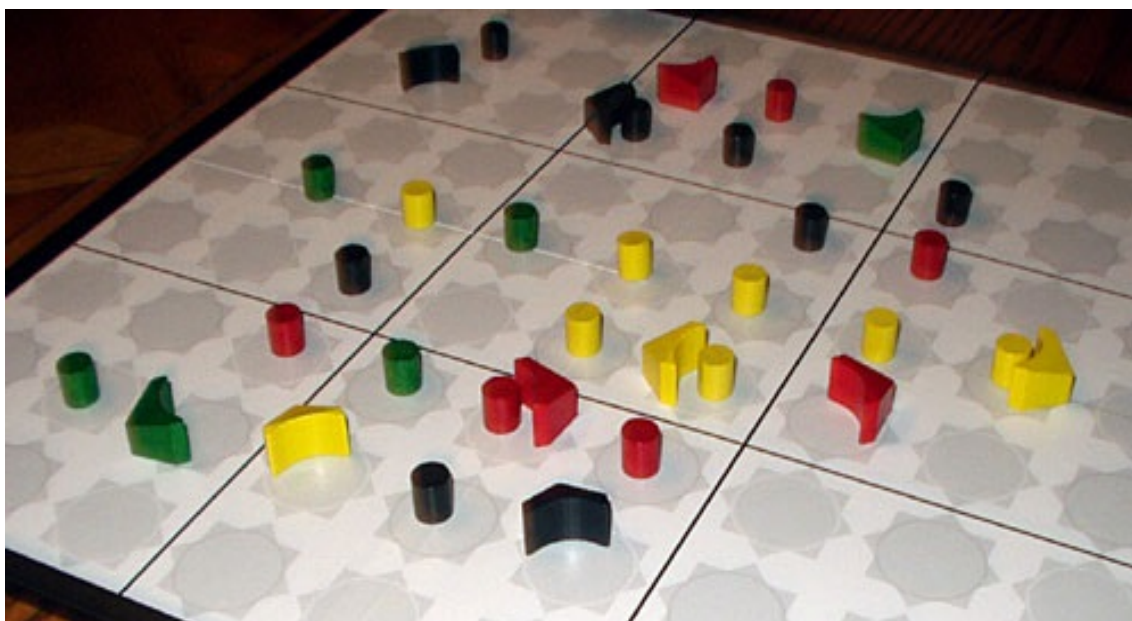
O Azacru é um jogo multi-jogador no qual cada um dos jogadores tem peças e marcadores de cor diferente. A ordem pela qual começam a jogar é deixada ao critério dos jogadores. Neste trabalho, vamos apenas implementar o jogo para 2 jogadores, sendo que por isso, cada jogador começa com 4 peças.

O jogo vai-se jogando alternadamente entre os jogadores, sendo que só é possível passar a vez caso o jogador esteja numa posição onde não consegue realizar qualquer jogada.

Os movimentos permitidos à peça são 3, sendo eles, em relação à posição da peça original, em frente, 45° à esquerda ou 45° à direita. Ao fim da jogada a peça fica a apontar na última direção em que foi jogada, sendo também colocado um marcador nessa posição do tabuleiro, exceto quando no final da jogada a peça tiver parado noutro sector.

O poder de alcance da peça numa jogada depende do sector em que se localiza, sendo que é o mesmo em casas à soma do número de marcadores da sua cor nesse sector. Quando não existem marcadores da cor da peça nesse sector o poder de alcance da peça é 1. Não é obrigatório percorrer todas as casas até ao poder máximo, a jogada pode ser de 1 casa até o número do alcance máximo.

Em baixo segue uma foto dum tabuleiro aquando da realização de um jogo entre 4 jogadores para se perceber melhor do que se fala quando se referem marcadores e sectores.



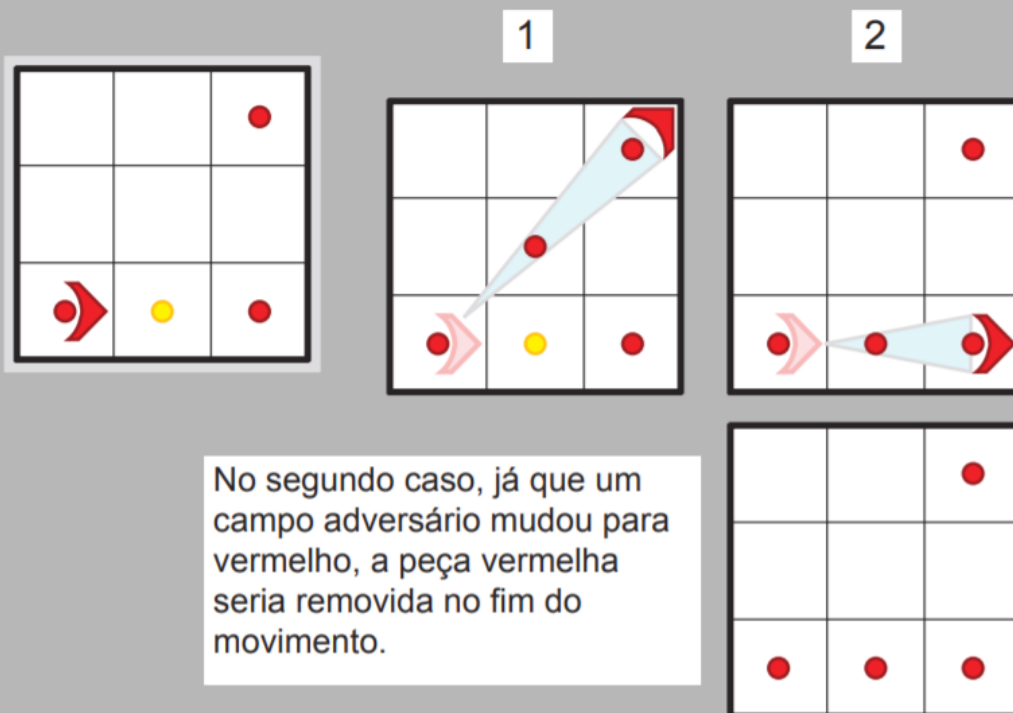
Quando a jogada inclui atravessar mais do que uma casa, não é permitido trocar a direção a meio do movimento, a direção pela qual se inicia o movimento é a direção pela qual a peça passará todas as casas.

Se no final da jogada a peça tiver passado para outro sector pode não terminar o seu movimento na posição em que iniciou a jogada, sendo que neste caso o jogador pode rodar a peça 45° no final da jogada quer seja para a esquerda ou para a direita.

Quando o jogador realiza um movimento da sua peça entre um campo da sua cor e outro dessa mesma cor, e não se passou através de uma peça, faz com que todos os campos intermédios fiquem também da cor da sua peça. Esta jogada pode também ser realizada saltando peças, contudo as casas intermédias entre a 1ª e a última não iram se alterar.

Quando, na jogada, são trocadas casas de cor do adversário para a cor da peça dessa jogada, a peça é então extraída do tabuleiro.

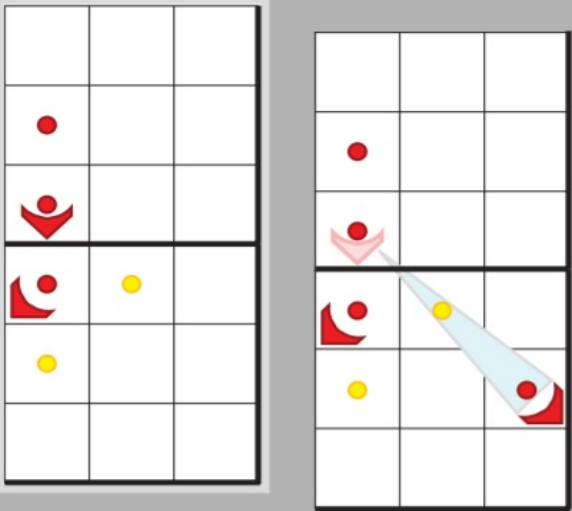
Neste exemplo, a peça vermelha tem dois movimentos possíveis que seriam conexões: cada uma começa e termina num campo vermelho.



É proibido mover uma peça para uma casa com a cor do adversário, não se pode também mover uma peça para um campo ocupado por outra. Só é permitido saltar por cima de uma peça, se a casa onde se for parar já for da cor da peça jogada, de outra forma não é possível.

O jogo termina quando um dos jogadores não for capaz de realizar nenhuma jogada ou quando todas as suas peças foram retiradas do jogo. Quando tal se sucede, o jogador passa a vez, os restantes jogadores jogam mais um turno e depois disso contabilizam-se os marcadores de cada jogador e quem tiver mais acumulados ganha o jogo.

De seguida, segue-se um exemplo de um término de jogo:



Exemplo de fim de jogo (seja a ordem do turno: vermelha, verde, amarela)

Vermelha move-se

Verde move-se (e remove a última peça verde do tabuleiro)

Amarela move-se

Vermelha move-se

Verde anuncia 'não posso mover-me' ou 'passo'

Amarela move-se

Vermelha move-se

O jogo termina e os marcadores de cada jogador são contabilizados.

3 Representação do Estado do Jogo

Detalhando agora a implementação em Prolog do jogo, o tabuleiro definir-se-à como uma lista de listas, especificamente nove listas de nove átomos, onde os átomos serão números inteiros representativos de cada campo no tabuleiro, e contendo informação sobre qualquer peça ou marcador que se possa lá encontrar. Os números serão constituídos por um ou três dígitos e terão a forma de:

$$\text{Átomo} = \text{OPM}$$

- O é um número entre 1 e 8, correspondendo à orientação da peça.
- P é um número entre 1 e 2, correspondendo à identidade da peça que aí se encontra (caso pertença ao jogador 1 ou 2).
- M é um número entre 0 e 2, correspondendo à identidade do marcador que aí se encontra (sendo 0 caso o campo esteja vazio).
- Se não estiver nenhuma peça presente nesse campo, então O e P serão substituídos por 0 (daí um número com apenas um dígito).

Deste modo, os estados iniciais, intermédios e finais do tabuleiro podem ser descritos por algo como as seguintes listas, com o respetivo output:

| Inicial | |
|---|--|
| [[420,000,510,000,000,000,520,000,000], | |
| [000,000,000,000,000,000,000,000,000], | |
| [000,000,000,000,000,000,000,000,000], | |
| [000,000,000,000,000,000,000,000,000], | |
| [310,000,000,000,000,000,000,000,720], | |
| [000,000,000,000,000,000,000,000,000], | |
| [000,000,000,000,000,000,000,000,000], | |
| [000,000,000,000,000,000,000,000,000], | |
| [000,000,110,000,000,000,120,000,810]] | |
| Intermédio | |
| [[000,000,000,000,211,000,520,000,000], | |
| [000,422,000,001,000,000,000,000,000], | |
| [000,000,000,000,000,000,000,000,000], | |
| [000,211,000,000,000,822,000,000,000], | |
| [000,000,000,000,000,000,002,002,000], | |
| [000,000,000,000,000,000,000,000,000], | |
| [000,111,000,000,000,000,000,000,000], | |
| [000,001,000,000,000,000,122,000,000], | |
| [000,000,000,000,000,000,000,000,810]] | |

Final

```
[[000,001,001,001,722,002,002,002,000],
[001,001,001,002,001,001,001,001,002],
[001,002,002,001,000,001,001,001,002],
[001,002,111,002,000,001,001,001,002],
[001,001,001,002,002,001,001,002,002],
[001,002,001,001,001,000,001,001,002],
[001,001,002,002,002,001,001,000,002],
[001,001,000,001,322,002,002,002,001],
[001,000,001,002,002,002,211,001,001]]
```

| | | | | | | | | | |
|---|---|---|---|---|-------|---|-------|---|---|
| | | | | | | | | | |
| | | 1 | 1 | 1 | 2 2 2 | 2 | 2 | 2 | |
| 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 2 |
| 1 | 2 | 2 | 1 | | | 1 | 1 | 1 | 2 |
| 1 | 2 | 1 | 1 | 2 | | 1 | 1 | 1 | 2 |
| 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 |
| 1 | 2 | 1 | 1 | 1 | 1 | | 1 | 1 | 2 |
| 1 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | | 2 |
| 1 | 1 | | | 1 | 2 2 2 | 2 | 2 | 2 | 1 |
| 1 | | | 1 | 2 | 2 | 2 | 1 1 1 | 1 | 1 |

4 Visualização do Tabuleiro

O tabuleiro é representado através de chamadas a *write/1* com átomos que consistem em strings de caracteres ASCII. Estas chamadas são sequenciadas através de uma série de predicados que primeiro efetuam o processamento do input e só depois procedem ao *display* dos átomos resultantes. O predicado que faz a chamada apropriada destes predicados é *print_matrix/1*, que deve receber a lista referida no capítulo anterior (esta lista deve ter exatamente nove listas de nove átomos).

Devido à natureza do jogo, quer no *display* quer nos predicados é transversal a complexidade, já que no primeiro caso recorre-se apenas a números para se efetuar a representação das peças do tabuleiro. A título de exemplo, a chamada ao predicado para visualizar o tabuleiro sem qualquer peça (vazio) teria como argumento:

```
print_matrix([[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],
[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],
[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0]]).
```

Produzindo o seguinte output:

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

Como se pode perceber pela imagem, cada campo é constituído por um espaço de 7 por 3 caracteres, sendo no entanto apenas utilizado um espaço de 3 por 3 caracteres. Assim, conseguimos exprimir elementos como este:

| | | | |
|--|-----|---|--|
| | | | |
| | 2 | | |
| | 2 2 | | |
| | | 2 | |

Que representa um campo com um marcador e uma peça do jogador 2 com orientação sudeste.

5 Movimentos

Os movimentos de cada jogador serão descritos por uma chamada ao predicado *move_piece/2*, que requer *input* do utilizador, levando como argumentos dois átomos, sendo um deles um par de coordenadas (um átomo com o formato CN, onde C é um caractere e N um número inteiro) que indicam a posição, no tabuleiro, da peça que pretendemos mover (de futuro, os valores das coordenadas serão mostrados juntamente com o tabuleiro, alinhados com as colunas e as linhas deste); o segundo argumento é um átomo com formato semelhante: o caractere é limitado a 'l', 'm' ou 'r', que significam respetivamente 'left', 'middle', 'right', indicando a direção, com base na orientação atual da peça, que pretendemos que o movimento tenha; e o número indica o número de campos através dos quais nos pretendemos mover.

Caso o movimento cause a alteração de sector por parte da peça, o predicado *rotate_piece/1* é chamado, pedindo ao utilizador o *input* de um único caractere, semelhante ao caractere do segundo argumento do *move_piece/2*, que efetuará a rotação da peça após esta ter completado o movimento.

A validação de cada um destes *inputs* acontecerá na lógica do jogo, que comparará o movimento descrito com uma lista de movimentos possíveis dado o estado do jogo. Se o movimento for válido, o tabuleiro será novamente representado com as mudanças resultantes do movimento da peça. Caso contrário, uma mensagem de erro será mostrada, juntamente com novo pedido de *input*.

Os turnos de cada jogador serão mantidos internamente, de modo a simplificar a utilização do jogo.

6 Movimentos

- <http://www.boardability.com/game.php?id=azacru>
- <http://www.pacru.com/rulesPT.pdf>