



# Otrio

## **Aplicação em Prolog para um Jogo de Tabuleiro**

### **Relatório Final**

Mestrado Integrado em Engenharia Informática e Computação  
Programação em Lógica

Grupo **Otrio\_4** Turma: 3MIEIC06

João Almeida – ei10099@fe.up.pt

Lázaro Costa – up201405342@fe.up.pt

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

13 de Novembro de 2016

# Resumo

Este relatório diz respeito ao trabalho prático desenvolvido para a unidade curricular de Programação em Lógica, com o objetivo de desenvolver uma aplicação para jogar um jogo de tabuleiro, usando Prolog como linguagem de implementação. O jogo em causa é o Otrio, a ser apresentado em detalhe posteriormente.

Numa fase inicial, foi realizado um estudo das regras do jogo, de modo a garantir que o funcionamento do mesmo estaria compreendido em toda a sua extensão pelos estudantes. Assim como requisitado no enunciado do Trabalho Prático, foram implementados três modos de utilização: Humano contra Humano; Humano contra Computador; Computador contra Computador. Também de acordo com a especificação, implementou-se uma interface em modo de texto que simula o tabuleiro físico do jogo original.

# Índice

1.	Introdução .....	4
2.	O Jogo Otrio.....	5
3.	Lógica do Jogo .....	6
3.1	Representação do Estado Tabuleiro .....	6
3.2	Visualização do Tabuleiro.....	6
3.3	Validação de Jogadas .....	8
3.4	Execução e validação de jogadas .....	8
3.5	A validação do tabuleiro .....	8
3.6	Final do Jogo .....	9
4.	Interface com o utilizador .....	10
5.	Conclusões .....	11
6.	Bibliografia .....	12

# 1. Introdução

O trabalho prático apresentava como objetivo a implementação de um jogo de tabuleiro na linguagem de programação em lógica Prolog. Mediante as opções de jogos de tabuleiro disponibilizadas, a nossa escolha recaiu sobre o Otrio, que captou a nossa atenção por ser um jogo moderno, com regras simples e fáceis de compreender, e que apresenta uma curva de aprendizagem bastante acessível para novos jogadores.

O presente relatório foi estruturado com base nas indicações fornecidas no enunciado do Trabalho Prático e no relatório intercalar: serão apresentados o Otrio e as respetivas regras de jogo; todos os aspetos analisados e implementados da lógica de jogo – da representação do estado até às jogadas do computador; a interface textual; e as conclusões alcançadas.

## 2. O Jogo Otrio

O Otrio é um jogo de estratégia para 2 a 4 jogadores, baseado num tabuleiro interno constituído por 3 linhas de 3 colunas. Em cada posição do tabuleiro podem ser colocadas 3 peças concêntricas.

Numa região externa ao tabuleiro existe uma secção onde são colocadas as peças no estado inicial do jogo. Cada jogador tem 3 conjuntos de peças concêntricas de 3 tamanhos diferentes e da mesma cor. O jogo, ao ocorrer com apenas dois jogadores, cada um escolhe duas cores e coloca as peças no tabuleiro, na região externa. As peças da mesma cor devem ser colocadas juntas.

No caso de o jogo ter três jogadores, cada um escolhe uma cor e são removidas do tabuleiro as peças da cor não utilizada.

Durante as jogadas, cada jogador retira uma peça da respetiva parte externa do tabuleiro e coloca-a no tabuleiro principal. Uma vez a peça colocada no tabuleiro, essa não pode ser movida.

No caso de o jogador não poder jogar, por falta de peças ou por falta de espaço para colocar a peça, deve deixar o jogador seguinte realizar a respetiva jogada.

O jogo termina quando nenhum jogador pode mover alguma peça ou quando algum jogador fizer “Otrio”.

Fazer “Otrio” resulta em vitória: basta ter uma combinação de 3 peças do mesmo tamanho numa linha na horizontal, vertical (Fig. 1) ou mesmo na diagonal.

E ainda pode ser uma combinação de 3 peças alinhadas de tamanho crescente ou decrescente, em qualquer direção (Fig. 2). Por último, também se pode fazer “Otrio” colocando 3 peças concêntricas no mesmo espaço do tabuleiro (Fig. 3).

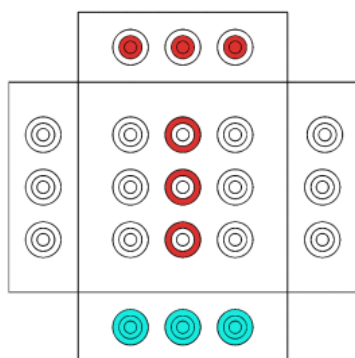


Fig. 1 - Fazer “Otrio” por colocação das peças do mesmo tamanho numa linha.<sup>1</sup>

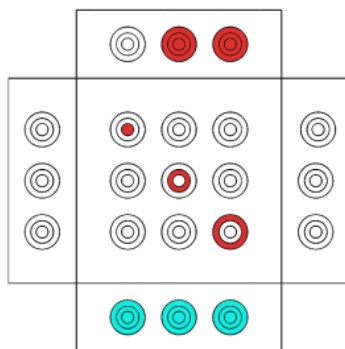


Fig. 2 - Fazer “Otrio” por colocação das peças por ordem crescente ou decrescente numa linha.<sup>1</sup>

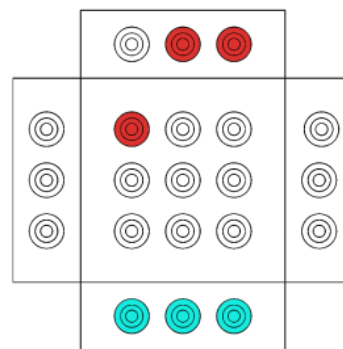


Fig. 3 - Fazer “Otrio” por colocação de 3 peças concêntricas numa posição do tabuleiro.<sup>1</sup>

1. Figuras obtidas em <https://barronwasteland.wordpress.com/2015/12/28/solving-otrio/>

## 3. Lógica do Jogo

### 3.1 Representação do Estado Tabuleiro

Para a representação do estado do tabuleiro, optou-se pela utilização de cinco listas, representando cada uma das mesmas um “sub-tabuleiro”: quatro correspondem à região inicial das peças de cada jogador (as secções de cima/baixo/esquerda/direita do tabuleiro de jogo); e a lista final refere-se ao “sub-tabuleiro” central, ou seja, a área de jogo.

As listas de cada jogador são compostas por três elementos que, por sua vez, são também uma lista de 3 elementos – as 3 peças de cada conjunto.

O tabuleiro central é composto por 3 elementos, sendo cada um a lista que corresponde à linha superior, intermédia, e inferior do tabuleiro, respetivamente. Cada uma das listas mencionadas contém também 3 listas, correspondendo respetivamente às primeira, segunda e terceira posições de cada conjunto de peças na linha correspondente do tabuleiro.

Os elementos correspondem, por ordem, às peças de fora (círculo de maior raio), do meio (círculo menor), cilindro interior. Para cada posição no tabuleiro utiliza-se a letra inicial da cor correspondente (**g**reen; **b**lue; **p**urple; **r**ed).

### 3.2 Visualização do Tabuleiro

Na Figura 4 apresenta-se a situação inicial de um tabuleiro para quatro jogadores, e de seguida as listas correspondentes:

Jogador 1: [ [r, r, r], [r, r, r], [r, r, r] ]

Jogador 2: [ [g, g, g], [g, g, g], [g, g, g] ]

Jogador 3: [ [b, b, b], [b, b, b], [b, b, b] ]

Jogador 4: [ [p, p, p], [p, p, p], [p, p, p] ]

Tabuleiro de Jogo:

[ [ [e, e, e], [e, e, e], [e, e, e] ],

[e, e, e], [e, e, e], [e, e, e] ],

[ [e, e, e], [e, e, e], [e, e, e] ] ]



Fig. 4 - Tabuleiro Inicial<sup>2</sup>

Jogador 1: [ [e, r, r], [e, e, r], [r, r, r] ]  
 Jogador 2: [ [g, g, g], [g, g, g], [e, e, e] ]  
 Jogador 3: [ [b, b, b], [e, e, e], [b, b, e] ]  
 Jogador 4: [ [e, p, p], [e, p, p], [e, p, p] ]



Fig. 5 - Exemplo de Situação Intermédia de Jogo<sup>2</sup>

Tabuleiro de Jogo:

$$\begin{bmatrix} [ [p, e, e], [g, g, b], [e, e, e] ], \\ [ [b, e, e], [r, r, g], [p, b, b] ], \\ [ [p, e, e], [e, e, e], [r, e, e] ] \end{bmatrix}$$

Nas Figuras 6 e 7 apresentam-se as situações de jogo apresentadas anteriormente, no formato de output resultante da implementação atual.

		G - G - G			G - G - G			G - G - G			
3	R - R - R	- -			- -			- -			B - B - B
2	R - R - R	- -			- -			- -			B - B - B
1	R - R - R	- -			- -			- -			B - B - B
		P - P - P			P - P - P			P - P - P			
		A			B			C			

Fig. 6 - Output de Exemplo de Situação Inicial de Jogo

		-----						
		G - G - G		G - G - G		- -		
		-----						
3	- P - P	P - -		G - G - B		- -		B - B - B
		-----						
2	- P - P	B - -		R - R - G		P - B - B		- -
		-----						
1	- P - P	P - -		- -		R - -		B - B -
		-----						
		- R - R		- - R		R - R - R		
		-----						
		A		B		C		

Fig. 7 - Output de Exemplo de Situação Intermédia de Jogo

Para visualizar o tabuleiro, foi implementado o predicado *display*, que tem 5 argumentos, correspondentes aos “sub-tabuleiros” mencionados anteriormente: *Top*; *Left*; *Center*; *Right*; e *Bottom*. *Top*, *Left*, *Right* e *Bottom* correspondem às listas que contêm as peças disponíveis de cada jogador; *Center* corresponde à lista que contém a zona de jogo – com uma lista para cada linha do tabuleiro.

O predicado *display* depende dos seguintes predicados adicionais:

- *printboardTop* – faz output da secção superior do tabuleiro, correspondendo às peças do jogador com as peças verdes, recorrendo a *printLine*
- *printboard* – utilizando os predicados adicionais *printPecas* e *printLine*, efetua o output das secções central, esquerda e direita do tabuleiro, que são mostradas nas mesmas linhas da interface de texto
- *printBottom* – faz output do número de cada coluna da zona de jogo, de modo a facilitar a interação com o utilizador
- *printLine* – faz output de caracteres adicionais para facilitar a interpretação pelo utilizador
- *printPecas* – faz o output do texto correspondente a cada uma das peças de jogo

### 3.3 Validação de Jogadas

A implementação efetuada não envolve uma geração de uma lista com jogadas válidas, alternativamente, sendo que qualquer jogada é válida se a posição indicada pelo jogador estiver vazia, utiliza-se o predicado *verifica* para pedir ao jogador as coordenadas da peça que quer mover, e as coordenadas de destino, e confirmar se as mesmas são válidas.

### 3.4 Execução e validação de jogadas

Os jogadores fazem jogadas à vez enquanto não se verificar nenhuma condição de fim de jogo (alguém ter realizado “Otrio” ou empate).

Cada jogada corresponde a retirar uma peça da área de jogo do jogador em questão, e em colocá-la de forma permanente na área de jogo. São utilizados os predicados *joga* ou *jogapc*, respetivamente para jogadores humanos ou artificiais. Estes têm como parâmetros o tipo de peça correspondente ao jogador; o número de peças do mesmo ao iniciar a jogada; o número resultante de peças do jogador após a mesma; a lista de peças da área do jogador; a lista de peças da área do jogador resultante da jogada efetuada; a zona central (área de jogo do tabuleiro) antes da jogada; e a zona central do tabuleiro resultante.

### 3.5 A validação do tabuleiro

De modo a evitar situações de jogo inválidas, as jogadas são validadas e só ocorrem se a posição do tabuleiro escolhida pelo jogador estiver vazia; sendo isto a única verificação necessária de jogadas no Otrio. Sendo que todas as jogadas são pré-validadas, conclui-se que os tabuleiros serão sempre válidos.



### 3.6 Final do Jogo

O Jogo termina quando algum jogador consegue fazer “Otrio” ou quando ocorre um empate – ou seja, quando todos os jogadores já jogaram todas as respectivas peças.

Para verificar se se observa alguma das duas condições de fim de jogo é utilizado o predicado *check\_victory*, que utiliza os seguintes predicados auxiliares:

- *check\_victory\_horizontal* – verifica as situações de “Otrio” (victória) em que o vencedor alinhou as peças na horizontal, este pode alinhar horizontalmente 3 peças iguais ou 3 peças em ordem crescente

- Padrões:

```
[[A, _, _], [A, _, _], [A, _, _]]
[[_ A, _], [_ A, _], [_ A, _]]
[[_ _ A], [_ _ A], [_ _ A]]
[[A, _, _], [_ A, _], [_ _ A]]
[[_ _ A], [_ A, _], [A, _ _]]
```

- *check\_victory\_concentric* - verifica as situações de “Otrio” (victória) em que o vencedor colocou 3 peças concêntricas numa determinada posição do tabuleiro, com o padrão [A,A,A]

- *check\_victory\_diagonal* - verifica as situações de “Otrio” (victória) em que o vencedor alinhou as peças na diagonal, este pode alinhar diagonalmente 3 peças iguais ou 3 peças em ordem crescente

- Padrões:

[[[A, _, _], [_ _ _], [_ _ _]],	[[[_ _ _], [_ _ _], [A, _ _]],
[[[_ _ _], [A, _ _], [_ _ _]],	[[[_ _ _], [A, _ _], [_ _ _]],
[[[_ _ _], [_ _ _], [A, _ _]]]	[[[A, _ _], [_ _ _], [_ _ _]]]

[[[_ A, _], [_ _ _], [_ _ _]],	[[[_ _ _], [_ _ _], [_ A, _]],
[[[_ _ _], [_ A, _], [_ _ _]],	[[[_ _ _], [_ A, _], [_ _ _]],
[[[_ _ _], [_ _ _], [_ A, _]]]	[[[_ A, _], [_ _ _], [_ _ _]]]

[[[_ _ A], [_ _ _], [_ _ _]],	[[[_ _ _], [_ _ _], [_ _ A]],
[[[_ _ _], [_ _ A], [_ _ _]],	[[[_ _ _], [_ _ A], [_ _ _]],
[[[_ _ _], [_ _ _], [_ _ A]]]	[[[_ _ A], [_ _ _], [_ _ _]]]

[[[A, _ _], [_ _ _], [_ _ _]],	[[[_ _ _], [_ _ _], [_ _ A]],
[[[_ _ _], [_ A, _], [_ _ _]],	[[[_ _ _], [_ A, _], [_ _ _]],
[[[_ _ _], [_ _ _], [_ _ A]]]	[[[A, _ _], [_ _ _], [_ _ _]]]

[[[_ _ A], [_ _ _], [_ _ _]],	[[[_ _ _], [_ _ _], [A, _ _]],
[[[_ _ _], [_ A, _], [_ _ _]],	[[[_ _ _], [_ A, _], [_ _ _]],
[[[_ _ _], [_ _ _], [A, _ _]]]	[[[_ _ A], [_ _ _], [_ _ _]]]

Decidiu-se utilizar os padrões apresentados após cada um dos predicados de modo a utilizar um método simultaneamente eficaz e consistente com o paradigma de programação em uso. De modo a evitar a redundância na criação de padrões semelhantes para identificar “Otrio” na vertical, optou-se por fazer a transposta da matriz com o predicado *transpose* na *library(lists)*, aplicando-se de seguida as verificações utilizadas para a horizontal.

## 4. Interface com o utilizador

Para correr o jogo, o utilizador deve correr o comando 'otrio.', que lança o jogo. De seguida, o utilizador é saudado e é-lhe pedida a informação em relação ao número e tipo de jogadores.

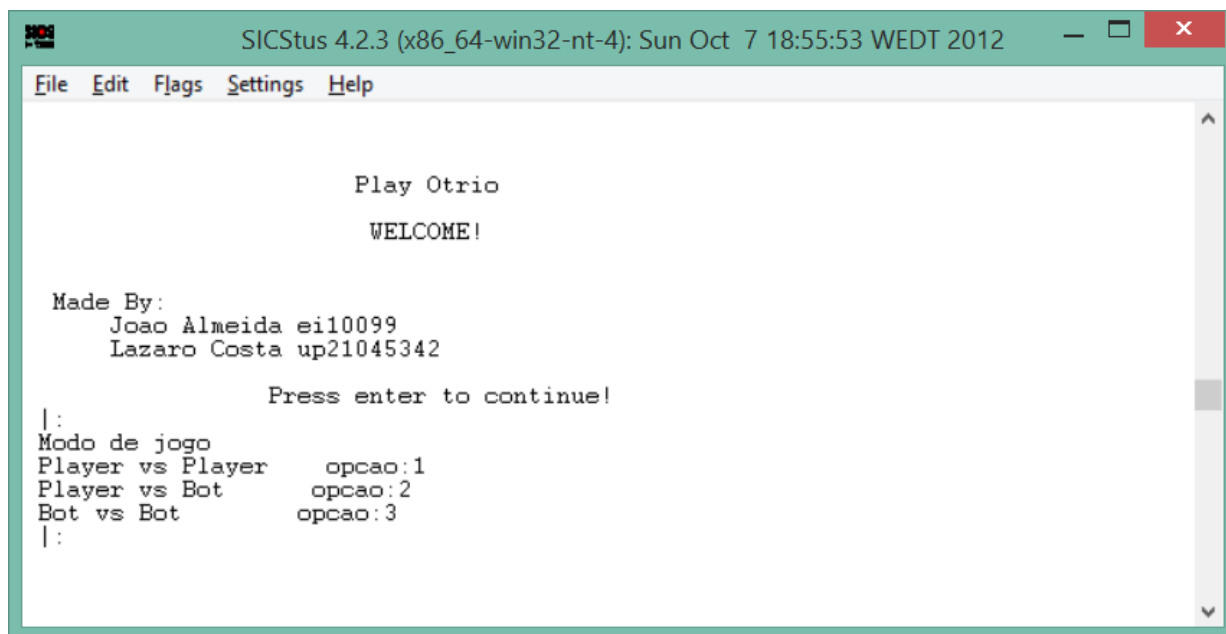


Fig. 8 – Interface – Boas Vindas

De seguida é mostrado o tabuleiro, e solicita-se a jogada do primeiro jogador. No caso de ser um jogador humano, pede-se as coordenadas da peça que este deseja mover (coordenadas específicas, correspondendo às peças dele no lado correspondente do tabuleiro), e de seguida as coordenadas de onde o jogador pretende colocar a peça na área de jogo. (Fig. 9)

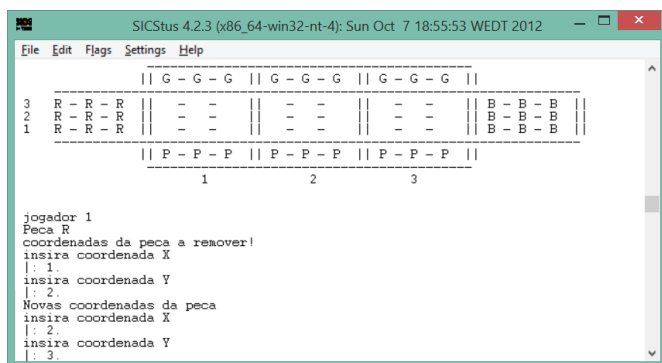


Fig. 9 – Interface – Coordenadas (1/2)

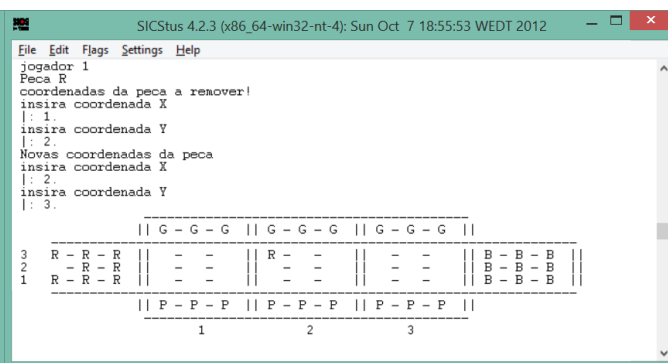


Fig. 10 – Interface – Coordenadas (2/2)

O tabuleiro atualizado é então mostrado na consola (Fig. 10) e solicita-se o movimento dos jogadores seguintes até ao fim do jogo. Para manter a interface uniforme, também se mostra o tabuleiro antes do 'computador' fazer a sua jogada; momentos depois, a jogada é realizada e passa a vez ao jogador seguinte.

Atingindo-se o final de um jogo, é então indicado o jogador vencedor. (Fig. 11)

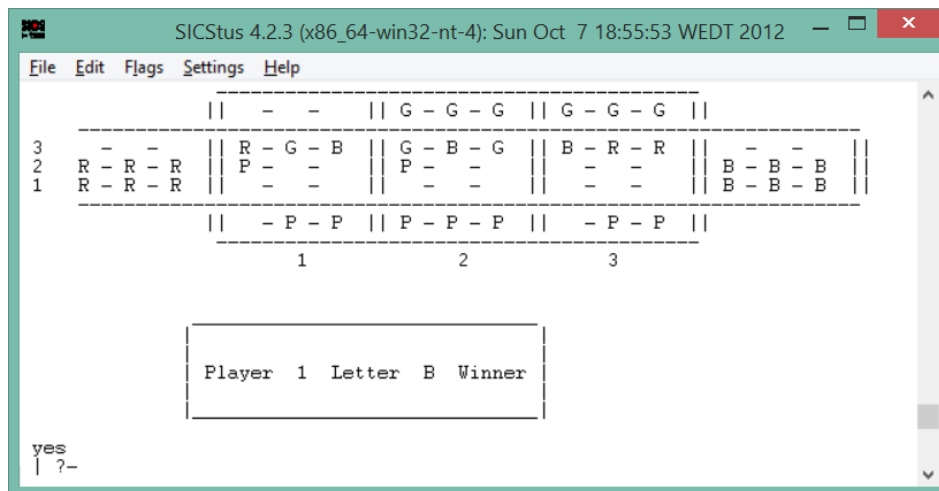


Fig. 11 – Interface – Fim de Jogo

## 5. Conclusões

O Jogo Otrio apesar da sua simplicidade pode ser um jogo bastante interativo e apelativo com o utilizador.

O presente trabalho, para além de ter permitido aprofundar o conhecimento da linguagem Prolog dos elementos do grupo, permitiu compreender e verificar as potencialidades da Programação em Lógica, e das formas como este paradigma se adequa melhor a determinados tipos de problemas.

A grande dificuldade do desenvolvimento do projeto foi a adaptação a uma linguagem puramente lógica e funcional, para além disso foram também encontradas algumas dificuldades na repartição do tabuleiro em pequenas partes.

Seria ainda possível introduzir algum grau de “inteligência artificial” para aumentar o grau de dificuldade dos jogadores artificiais, no entanto infelizmente o grupo não conseguiu atingir esse objetivo dentro do prazo indicado, mas considerando o propósito deste trabalho prático, a implementação do jogo apresentada parece-nos cumprir os objetivos pretendidos. Foi realizado algum progresso no sentido de selecionar uma jogada que pudesse levar à vitória, com o predicado *jogaPcInteligente*, no entanto o resultado pretendido não foi atingido.

## 6. Bibliografia

Os seguintes *websites* foram utilizados para obter informação e recursos sobre o Jogo de Tabuleiro:

Loja *online* especializada em jogos de tabuleiro, com imagens e regras:

<http://www.marblethebrainstore.com/otrio.htm>

Solving Otrio (imagens e detalhes):

<https://barronwasteland.wordpress.com/2015/12/28/solving-otrio/>

Otrio no boardgamegeek:

<https://boardgamegeek.com/boardgame/188465/otrio>