

FACULDADE DE ENGENHARIA  
DA UNIVERSIDADE DO PORTO



RELATÓRIO INTERMÉDIO

PROGRAMAÇÃO EM LÓGICA

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E  
COMPUTAÇÃO

---

**Zurero**

---

*Autores:*

Afonso Jorge Ramos

João Dias Conde Azevedo

up201506239@fe.up.pt

up201503256@fe.up.pt

Outubro 2018

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Zurero</b>	<b>3</b>
<b>3</b>	<b>Lógica do Jogo</b>	<b>3</b>
3.1	Representação do Estado do Jogo . . . . .	3
3.2	Visualização do Tabuleiro . . . . .	6
<b>4</b>	<b>Bibliografia</b>	<b>9</b>

# 1 Introdução

No âmbito da unidade curricular de Programação em Lógica, do curso Mestrado Integrado em Engenharia Informática e Computação, foi-nos sugerido o desenvolvimento de um jogo utilizando a linguagem de programação PROLOG. De entre as várias opções que nos foram disponibilizados pelos docentes foi escolhido o jogo Zurero. Após alguma investigação percebemos que o Zurero é um jogo interessante com características peculiares para o que é um jogo de tabuleiro clássico, como o "deslizar" das pedras.

O objetivo deste trabalho foi a aplicação dos primeiros conceitos interiorizados nas aulas teóricas e desenvolvidos nas aulas práticas da cadeira. Este método de avaliação torna-se importante pois permite-nos avaliar os conhecimentos que adquirimos até então e saber se somos ou não capazes de, com uma linguagem de programação nova e um paradigma completamente diferente do que estamos habituados, produzir algo de útil para o quotidiano e futuro.

## 2 Zurero

Zurero é jogado num tabuleiro de 19x19, inicialmente vazio. O jogo foi inventado por Jordan Goldstein em 2009. É uma interessante variação do clássico jogo "Go" onde dois jogadores, com cores distintas, devem "atirar" e fazer "deslizar" pedras ao longo do tabuleiro, desde as bordas do tabuleiro até que atinjam e parem noutra pedra.

O interessante e peculiar neste jogo é o facto de que caso uma pedra deslize e atinja outra que tem um espaço vazio atrás, esta última é empurrada.

O objetivo do jogo é colocar 5 pedras da cor do jogador seguidas, quer numa direção horizontal, vertical ou diagonal.

Usualmente, as peças pretas jogam primeiro, colocando a primeira peça no tabuleiro com total liberdade de escolha. Nas jogadas seguintes, alternando entre o preto e o branco, um jogador faz deslizar uma pedra de uma borda do tabuleiro para o centro, seguindo as regras de colisão já mencionadas.

Se ao empurrar uma pedra adversária o jogador coloca 5 peças do adversário numa coluna, fila ou diagonal, perde o jogo, a não ser que ele próprio faça uma coluna, fila ou diagonal.

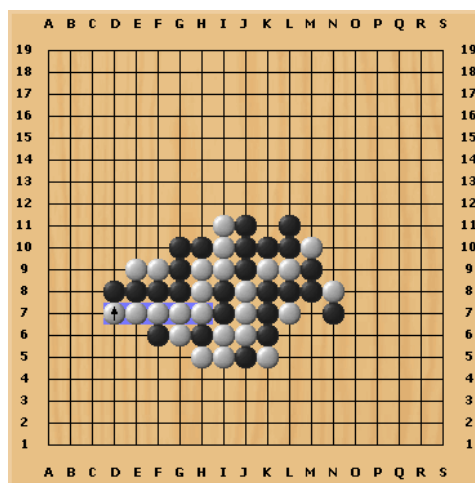


Figura 1: Exemplo de um tabuleiro a meio de jogo

## 3 Lógica do Jogo

### 3.1 Representação do Estado do Jogo

O estado de jogo é guardado no tabuleiro, representado por uma lista de listas. O tabuleiro é de 19x19 **interseções** e, por isso, a primeira lista conterá outras 19, cada uma dessas com 19 elementos.

Para exemplificação, o código e comentários abaixo representam em linguagem PROLOG o estado inicial do tabuleiro e possíveis estados ao longo do jogo. Cada número de representação interna ao programa é traduzido em um ou mais caracteres na consola do SICStus.

A chave de tradução é também apresentada em baixo.

```

1 /* Starting game board */
2 initial_board ([[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
3                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
4                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
5                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
6                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
7                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
8                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
9                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
10                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
11                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
12                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
13                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
14                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
15                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
16                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
17                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
18                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
19                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
20                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]]).

```

Listing 1: Board inicial.

```

1 /* Mid game board */
2 midgame_board ([[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
3                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
4                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
5                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
6                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
7                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
8                 [0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0],
9                 [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0],
10                [0,0,0,0,0,0,0,0,2,1,1,1,2,0,0,0,0,0,0],
11                [0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,0],
12                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
13                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
14                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
15                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
16                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
17                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
18                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
19                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
20                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]]).

```

Listing 2: Board intermédio.

```

1 /* End game board */
2 endgame_board ([[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
3                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
4                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
5                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
6                 [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
7                 [0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0],
8                 [0,0,0,0,0,0,0,1,2,0,0,0,0,0,0,0,0,0,0,0],
9                 [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0],
10                [0,0,0,0,0,0,0,2,1,1,1,2,0,0,0,0,0,0,0,0],
11                [0,0,0,0,0,0,0,0,0,2,1,2,0,0,0,0,0,0,0,0],
12                [0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0],
13                [0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0],
14                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
15                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
16                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
17                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
18                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
19                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
20                [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]]).

```

Listing 3: Board final.

```

1 /* Associates each number with each piece */
2 translate(0,'-+-').
3 translate(1,'-B-'). %Black Pieces
4 translate(2,'-W-'). %White Pieces

```

Listing 4: Tradução da representação interna.

### 3.2 Visualização do Tabuleiro

A representação interna do tabuleiro inicial será como a anteriormente apresentada, traduzindo-se no seguinte output de consola no SICStus Prolog:

```

| ?- print_initial_board.
      x-----x
      | | | | | | | | | | | | | | | | | | | | |
19|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
18|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
17|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
16|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
15|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
14|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
13|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
12|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
11|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
10|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
9|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
8|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
7|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
6|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
5|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
4|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
3|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
2|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
1|---+---+---+---+---+---+---+---+---+---+---+---+
      | | | | | | | | | | | | | | | | | | | | |
      x-----x
      A B C D E F G H I J K L M N O P Q R S

```

Figura 2: Layout de jogo em estado inicial.

Uma representação interna do tabuleiro a meio de um jogo como a acima apresentada traduz-se no seguinte output de consola no SICStus:

```

| ?- print_mid_board.
x-----x
| | | | | | | | | | | | | | | | | | | | |
19|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
18|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
17|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
16|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
15|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
14|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
13|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
12|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
11|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
10|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
9|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
8|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
7|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
6|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
5|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
4|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
3|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
2|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
1|+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | | | | | | | | | | | |
x-----x
  A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S

```

Figura 3: Layout de jogo em estado intermédio.



Ainda, uma possível representação do tabuleiro num estado de jogo final será a seguinte:

```

I ?- print_end_board.
x-----x
19|-----|
18|-----|
17|-----|
16|-----|
15|-----|
14|-----B-B-----|
13|-----B-W-----|
12|-----B-----|
11|-----W-B-B-B-W-----|
10|-----W-B-W-----|
9|-----W-----|
8|-----W-----|
7|-----|
6|-----|
5|-----|
4|-----|
3|-----|
2|-----|
1|-----|
x-----x
  A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S

```

Figura 4: Layout de jogo em estado final.

Face a esta representação, necessária, devido ao facto de um tabuleiro de zurero ser disposto desta forma, com letras (crescentes da esquerda para a direita) para identificar as colunas e números (crescentes de baixo para cima) para identificar as linhas, conseguimos que para o utilizador, não haja dificuldades de adaptação a um tabuleiro diferente do utilizado em outros jogos de tabuleiro.

O tabuleiro inicial de jogo é criado usando o predicado *initial\_board(-Board)* em que *Board* contem o tabuleiro inicial.

Para efeitos de apresentação foi construído o predicado *print\_board(+Board)* que recebe uma matriz Board (lista de listas) e a imprime na consola, sendo que o tabuleiro atual é sempre representado no final de cada jogada, depois de a jogada ter sido aprovada pelos vários predicados de verificação. É um predicado que chama outro auxiliar, *print\_board\_aux(+Board, +LineNumbers)* que é chamado de forma recursiva, que se auxilia noutro predicado, também ele recursivo, *print\_line(+Line)*.

## 4 Bibliografia

[1] igGameCenter

[2] Board Game Geek