

Programação em Lógica

# *Aplicação em Prolog do Jogo de Tabuleiro Exo*

*MIEIC – Turma 5 – Grupo Exo\_2  
(20 de outubro de 2019)*

# Índice

|  |   |
|--|---|
| Descrição Do Jogo.....                             | 3 |
| 1. Objetivo Do Jogo.....                           | 3 |
| 2. Configuração.....                               | 3 |
| 3. Como Jogar .....                                | 4 |
| 4. Finalização Do Jogo E Condições De Vitória..... | 5 |
| Modelação Do Jogo Em Prolog .....                  | 6 |
| Representação Interna Do Estado Do Jogo .....      | 6 |
| Situação Inicial .....                             | 6 |
| Situação Intermedia.....                           | 6 |
| Situação Final .....                               | 7 |
| Imagens Ilustrativas .....                         | 8 |
| Visualização Do Tabuleiro Em Modo De Texto .....   | 9 |

# Descrição do Jogo



## 1. História

O jogo Exo foi criado em março de 2019 e faz parte de um conjunto de 5 jogos pertencentes a uma coleção de nome “Cut and Play” criada por Léandre Proust, um criador de jogos de tabuleiro francês.

## 2. Objetivo do Jogo

O objetivo do Jogo Exo é criar combinações de planetas alinhando-os.

## 3. Configuração

No início do jogo, cada jogador cria a sua galáxia ao colocar uma estrela à sua frente. Formámos o universo ao misturar os 27 planetas entre as estrelas (ver a figura 1).

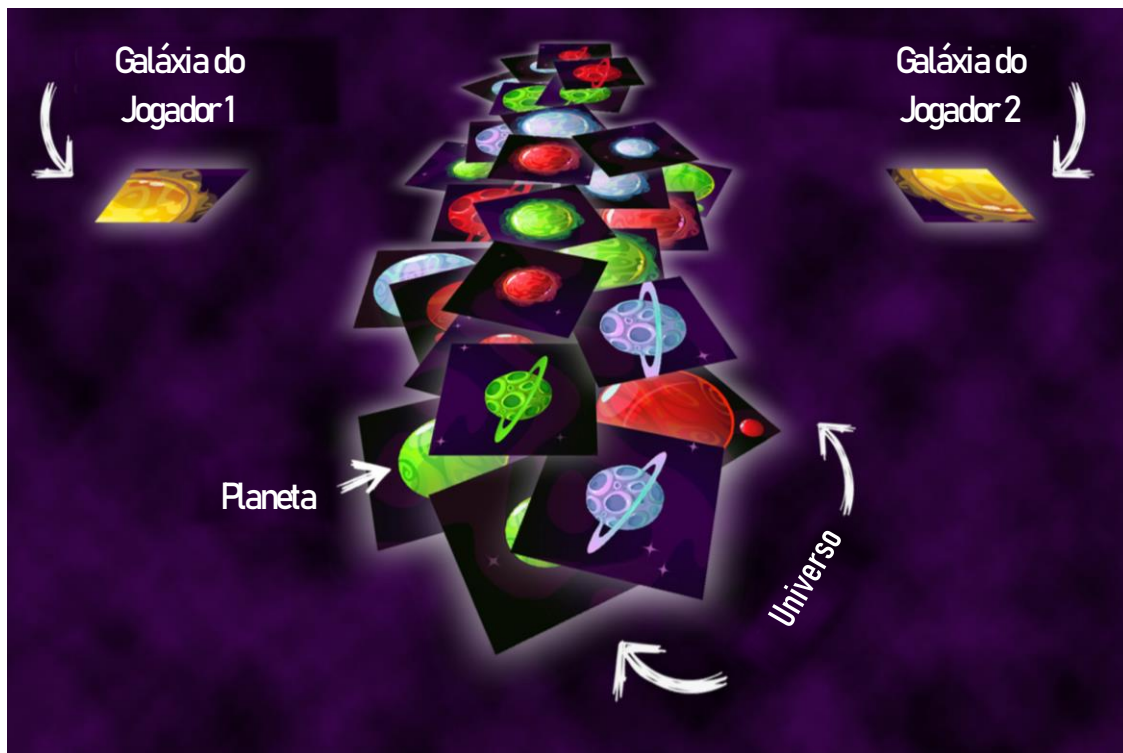


Figura 1 – Estado Inicial do Jogo.

#### 4. Como jogar

Os jogadores revezam-se no sentido dos ponteiros do relógio. Durante a sua vez, o jogador deve escolher um planeta disponível (um que não esteja coberto por outro planeta) do universo e colocá-lo na sua galáxia.

O primeiro planeta na galáxia do jogador deve ser colocado num dos 8 quadrados adjacentes à estrela. Posteriormente, os planetas devem ser colocados ou adjacentes à estrela ou a um planeta da galáxia. Um planeta não pode ser movido depois de colocado na galáxia.

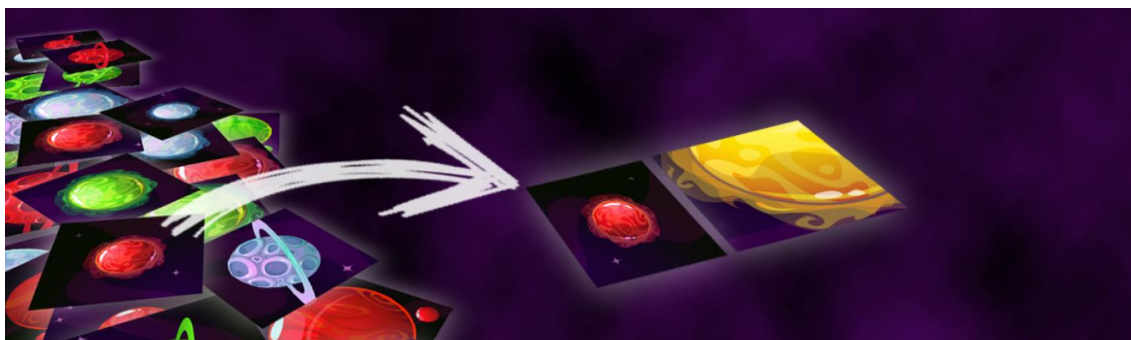


Figura 2 – Exemplo de uma colocação correcta do 1º planeta.

Os planetas possuem diferentes características: tamanho (pequeno, médio, grande), cor (azul, vermelho, verde) e tipo (terrestre, gasoso, anelado). O jogador ganha pontos quando cria linhas na sua galáxia com 3 planetas com características comuns.

### 5. Finalização do Jogo e Condições de Vitória

O jogo acaba quando cada jogador possui 13 planetas num jogo com dois jogadores ou 9 planetas num jogo com 3 jogadores. Os jogadores ganham 1 ponto por cada linha (horizontal, vertical ou diagonal) de três planetas com uma característica em comum. Se uma linha de planetas partilha mais de que uma característica, um ponto é alocado por cada característica partilhada.

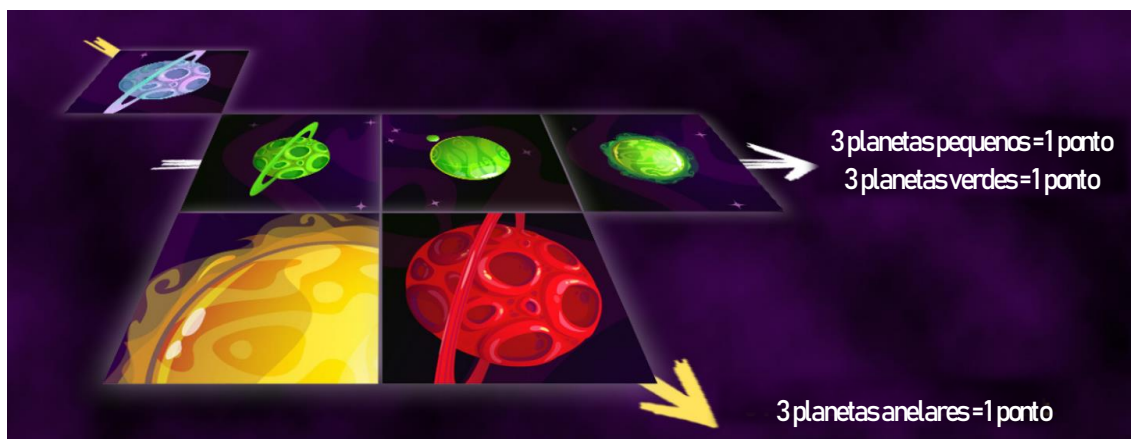


Figura 3 – Contagem dos pontos.

O jogador com mais pontos é declarado vencedor. Em casa de empate, o jogador com mais planetas adjacentes à sua estrela ganha.

# Modelação do Jogo em Prolog

## *Representação interna do estado do jogo*

Uma vez que neste jogo cada jogador tem o seu tabuleiro de jogo, são exemplificadas a seguir 6 representações do estado de jogo: duas iniciais, duas intermédias e duas finais, uma para cada jogador.

### Situação inicial

Jogador 1:

```
initialBoardP1 ([  
[sun]]).
```

Jogador 2:

```
initialBoardP2 ([  
[sun]]).
```

### Situação intermedia

Jogador 1:

```
intermediateBoardP1 ([  
[empty, planet (medium, white, terrestrial), empty],  
[empty, planet (medium, green, terrestrial), planet (small, green,  
gaseous)],  
[empty, planet (small, red, terrestrial), planet (large, green,  
gaseous)],  
[planet (small, red, ringed), sun, empty]]).
```

Jogador 2:

```
intermediateBoardP2 ([  
[planet (large, green, terrestrial), empty, empty],  
[planet (small, green, terrestrial), empty, planet (medium, white,  
gaseous)],  
[sun, planet (medium, green, gaseous), empty],  
[planet (small, red, gaseous), empty, planet (small, green, ringed)]]).
```

## Situação final

Jogador 1:

```
finalBoardP1 ([
[empty, empty, planet (small, white, ringed), planet (medium, white,
ringed)],
[empty, empty, planet (medium, white, terrestrial), planet (medium,
white, gaseous)],
[empty, planet (medium, green, ringed), planet (medium, green,
terrestrial), planet (small, green, gaseous)],
[empty, empty, planet (small, red, terrestrial), planet (large, green,
gaseous)],
[empty, planet (small, red, ringed), sun, planet (large, red, ringed)],
[planet (medium, red, gaseous), planet (small, white, terrestrial),
empty, empty]]).
```

Jogador 2:

```
finalBoardP2 ([
[planet (large, white, ringed), empty, empty],
[planet (large, green, ringed), empty, empty],
[planet (large, green, terrestrial), empty, planet (medium, white,
terrestrial)],
[planet (small, green, terrestrial), empty, planet (medium, white,
gaseous)],
[sun, planet (medium, green, gaseous), planet (medium, red, ringed)],
[planet (small, red, gaseous), planet (small, white, gaseous), planet
(small, green, ringed)],
[planet (large, red, gaseous), empty, planet (large, white, ringed)]]).
```

## Imagens Ilustrativas

**Jogador 1:**

|     |  |
|-----|--|
| --- |  |
| sun |  |

**Jogador 2:**

|     |  |
|-----|--|
| --- |  |
| sun |  |

Figure 4 – Situação Inicial.

**Jogador 1:**

|     |     |     |
|-----|-----|-----|
| --- | MWT | --- |
| --- | MGT | SGG |
| --- | SRT | LGG |
| SRR | sun | --- |

**Jogador 2:**

|     |     |     |
|-----|-----|-----|
| LGT | --- | --- |
| SGT | --- | MWG |
| sun | MGG | --- |
| SRG | --- | SGR |

Figure 5 – Situação Intermédia.

**Jogador 1:**

|     |     |     |     |
|-----|-----|-----|-----|
| --- | --- | SWR | MWR |
| --- | --- | MWT | MWG |
| --- | MGR | MGT | SGG |
| --- | --- | SRT | LGG |
| --- | SRR | sun | LRR |
| MRG | SWT | --- | --- |

**Jogador 2:**

|     |     |     |
|-----|-----|-----|
| LWR | --- | --- |
| LGR | --- | --- |
| LGT | --- | MWT |
| SGT | --- | MWG |
| sun | MGG | MRR |
| SRG | SWG | SGR |
| LRG | --- | LWR |

Figure 6 – Situação Final.



## *Visualização do tabuleiro em modo de texto*

```
/* Prints the Matrix */
display_game([]).

display_game([Head | Tail]) :-
    length(Head, LenList),
    write('\n'),
    writeDivisions(LenList),
    write(' '),
    printLine(Head),
    display_game(Tail).

/* Writing Divisions */
writeDivisions(0) :- write('\n').

writeDivisions(N) :-
    N > 0,
    N1 is N - 1,
    write('-----|'),
    writeDivisions(N1).

/* Prints each Line */
printLine([]).

printLine([Head|Tail]) :-
    symbol(Head, S),
    write(S),
    write(' | '),
    printLine(Tail).

/*Representing a play */

symbol(empty, '  ').

symbol(sun, 'sun').

symbol(planet(Size, Colour, Type), S):-
    symbolSize(Size, S1),
    symbolColour(Colour, S2),
    symbolType(Type, S3),
    atom_concat(S1, S2, SF1),
    atom_concat(SF1, S3, S).

symbolSize(small, 'S').
symbolSize(medium, 'M').
symbolSize(large, 'L').
```

```
symbolColour(red, 'R').  
symbolColour(green, 'G').  
symbolColour(white, 'W').  
  
symbolType(terrestrial, 'T').  
symbolType(gaseous, 'G').  
symbolType(ringed, 'R').
```

**NOTA:** O output produzido está ilustrado na secção anterior.

# Bibliografia

- [https://www.boardgamegeek.com/boardgame/276489/exo/files?fbclid=IwAR2DQZ6yomaBZqpMHjPAit3emuuC3zapW7X6AgUXDql\\_x8528rJE2yP5wtQ](https://www.boardgamegeek.com/boardgame/276489/exo/files?fbclid=IwAR2DQZ6yomaBZqpMHjPAit3emuuC3zapW7X6AgUXDql_x8528rJE2yP5wtQ)
- <https://fr.tipeee.com/leandreproust?fbclid=IwAR1boZ-Zu6N8fxWCMVqpGVcU8XKKFaDiSZExnXcLBJ2TJyvfdilGgkMVXBq>
- <https://www.kickstarter.com/projects/924686715/cut-and-play-collection-of-free-micro-games?fbclid=IwAR3m5lrB4o3sIFsdsdPC573NJ0Yfvtsc0e9kuLYUVHsECx6YWvzWrDXBDs8>