Adaptoid

Relatório Intercalar

Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

Grupo Adaptoid_2:

Marcelo Diocleciano Rodrigues Ferreira - up201405323
Pedro Daniel Oliveira Pacheco - up201406316

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

16 de Outubro de 2016

O Jogo Adaptoid

Introdução ao jogo:

O **Adaptoid** é um jogo de tabuleiro para duas pessoas e foi criado por Néstor Romeral Andrés em 2007. O tempo de duração do jogo ronda os 20 minutos.

Um "**adaptoid**" é uma criatura que está em constante evolução para se adaptar ao ambiente. Para sobreviver esta precisa de se manter alimentada. O jogo envolve dois exércitos de **adaptoids** que lutam entre si com o objetivo de eliminar os oponentes.

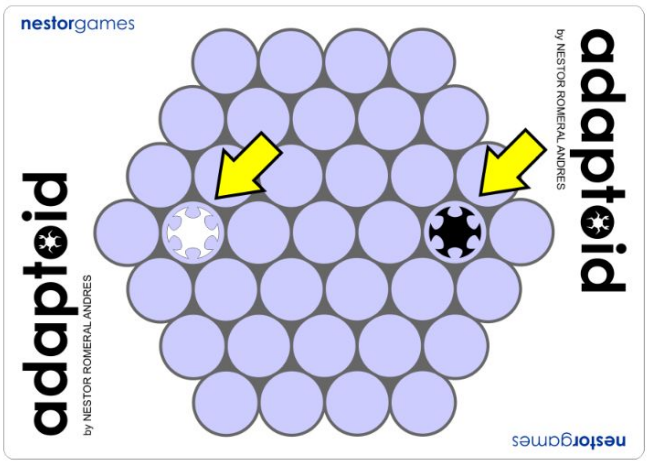
Sobre o jogo:

O tabuleiro de jogo é hexagonal e é constituído por 37 casas. Cada jogador começa com 12 corpos de **adaptoids**, 12 patas e 12 garras. Um jogador tem estas peças de cor preta e o outro de cor branca. Os corpos têm espaços para que as patas ou garras sejam colocadas ao longo do jogo. Por norma, cada corpo de um **adaptoid** tem 6 espaços que podem ser preenchidos.

Estas imagens representam, respectivamente, um corpo de um adaptoid, uma pata e uma garra de cada cor.

Preparação do jogo:

Antes de começar a jogar deve ser escolhida aleatoriamente a cor de cada jogador. Depois disso, cada jogador deve colocar um corpo de adaptoid no tabuleiro. A posição inicial por norma é a seguinte:



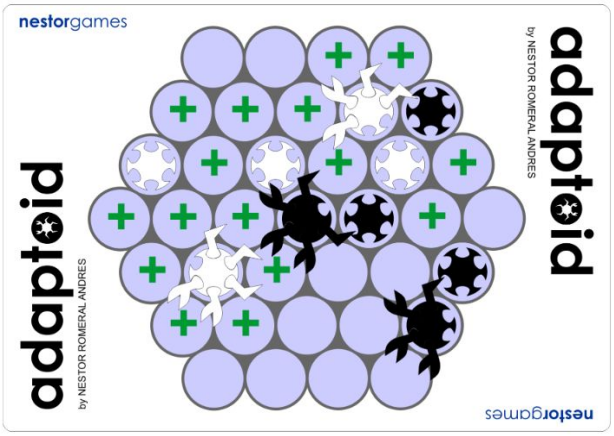
Estas posições iniciais podem ser alteradas se ambos os jogadores concordarem. Após o posicionamento das peças, o jogador de cor branca começa a jogar.

Regras do jogo:

O turno de cada jogar está dividido em três fases.

Crescimento:

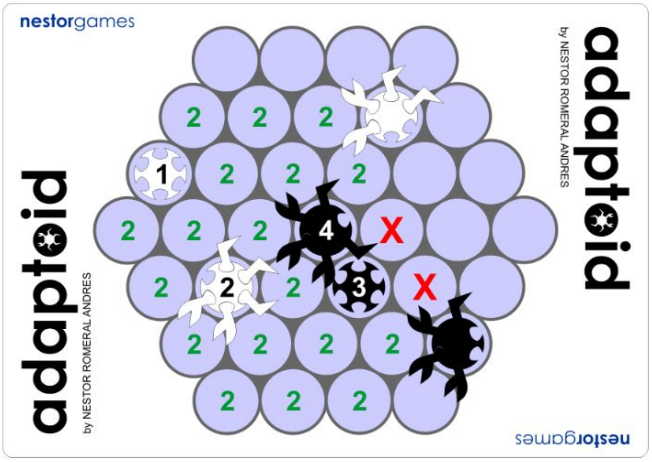
Nesta fase o jogador pode optar por adicionar um corpo vazio do adaptoid(sem garras nem patas) ao tabuleiro, sendo que apenas o pode posicionar de forma adjacente a um **adaptoid** existente ou pode adicionar uma pata ou uma garra a um adaptoid já criado anteriormente.Como já foi dito, cada **adaptoid** só pode ter 6 membros no máximo.



Neste exemplo, um novo **adaptoid** branco pode ser criado nas casas com o símbolo “+”.

Movimento:

Na fase de movimento, o jogador pode mover um **adaptoid** o número de casas correspondente ao número de patas que este tem. Por exemplo, um **adaptoid** com três patas pode ser movido por três casas, sendo que não pode passar por casas ocupadas por outros **adaptoids.** É possível terminar o movimento numa casa ocupada, depois o **adaptoid** que tiver mais garras é o que fica nessa casa. Se ambos tiverem o mesmo número de garras, são os dois removidos do tabuleiro.



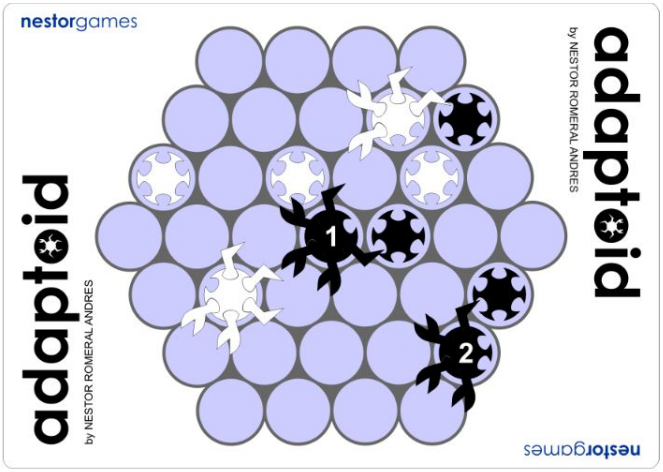
Neste exemplo, o **adaptoid 1** não se consegue mover porque não tem patas.

O **adaptoid 2** pode se mover 3 casas pois tem 3 patas, essas possibilidades estão marcadas com o número 2 verde, sendo que não pode ir para as casas que estão marcadas com um “X”, pois teria de passar por casas já ocupadas por outros **adaptoids**.

O **adaptoid 2** pode capturar o **adaptoid 3,** terminando o movimento na casa deste. A captura é possível dado que o **adaptoid 2** tem mais garras que o **adaptoid 3.** Já a captura do **adaptoid 4** não seria possível, pois tem mais garras que o **adaptoid 2.**

Alimentação:

Nesta fase, os **adaptoids** precisam de se alimentar, isto significa, que o número de casas livres à sua volta tem de ser no mínimo equivalente ao número de membros que estes possuem no momento. Um **adaptoid** com 3 membros, precisa de 3 casas livres adjacentes a ele para sobreviver. Caso contrário, todos os **adaptoids** do adversário que não se encontrem nestas condições são removidos do tabuleiro e contam como capturas.



Neste exemplo, o jogo está no turno do jogador branco e está na altura de capturar os **adaptoids** pretos que não foram alimentados. O **adaptoid** com o número 1,tem 5 membros mas apenas 4 casas livres adjacentes a ele. Sendo assim, este será capturado. O mesmo acontece com o **adaptoid** número 2 e ambos são removidos do tabuleiro e o jogador branco pontua 2 capturas.

Final do Jogo:

O jogo termina quando um dos jogadores consegue no mínimo 5 capturas. Por outro lado, se algum jogador ficar sem **adaptoids** no tabuleiro, perde automaticamente. Em caso de empate, ganha o jogador que fez a última jogada.

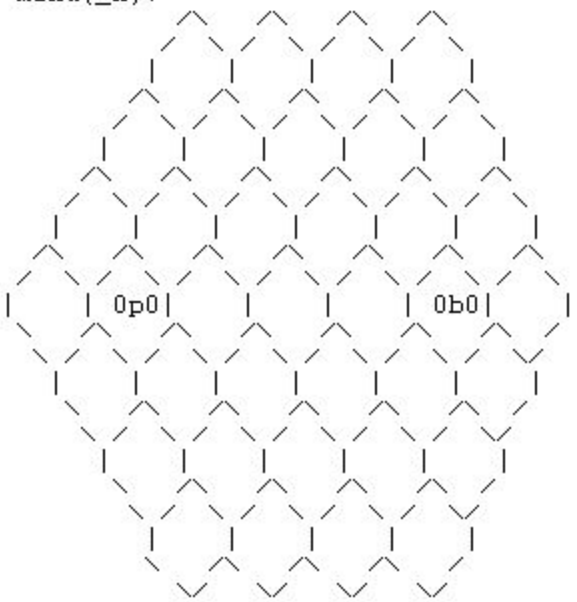
Representação do Estado do Jogo

Tradução dos símbolos presentes nas células do tabuleiro impresso:

-> n p y

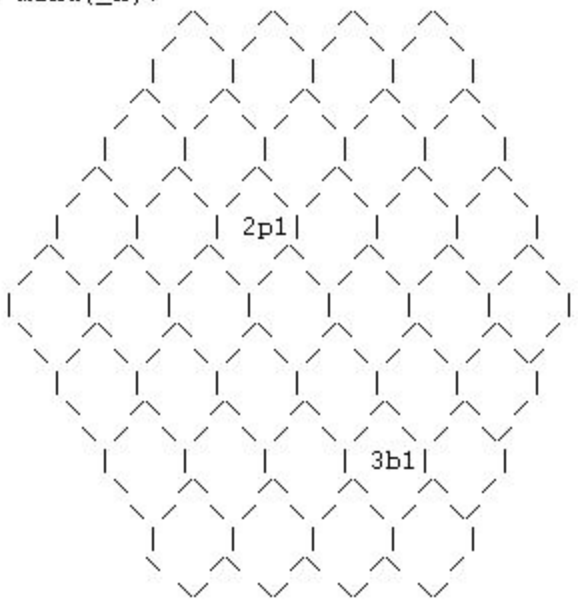
- n - indica o número de patas do adaptoid em causa.
- p - identifica a cor preta do adaptoid (caso fosse b, identificaria a cor branca) .
- y - identifica o número de pinças do adaptoid em causa.

```
?- menu(_X).
```




```
tabuleiro1( [
            ['!','!','x',' ',' ',' ',' '],
            ['!','x',' ',' ',' ',' ',' '],
            ['x',' ',' ',' ',' ',' ',' '],
            [' ',[0 , 'p', 0],' ',' ',' ',[0 ,'b',0],' '],
            ['x',' ',' ',' ',' ',' ',' '],
            ['!','x',' ',' ',' ',' ',' '],
            ['!','!','x',' ',' ',' ',' ']
             ]
          ).
```

```
?- menu(_X).
```



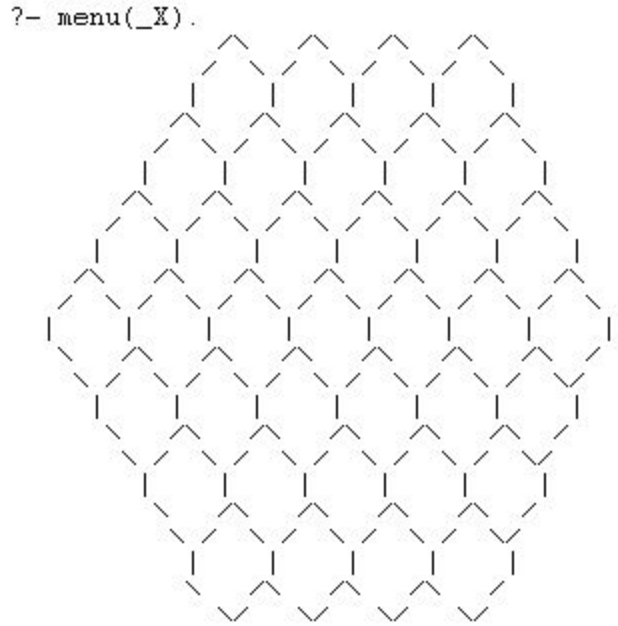

```
tabuleiro( [
            ['!','!','x',' ',' ',' ',' '],
            ['!','x',' ',' ',' ',' ',' '],
            ['x',' ',' ',[2,'p',1],' ',' ',' '],
            [' ',' ',' ',' ',' ',' ',' '],
            ['x',' ',' ',' ',' ',' ',' '],
            ['!','x',' ',' ',' ',[3, 'b',1],' '],
            ['!','!','x',' ',' ',' ',' ']
             ]
          ).
```

Visualização do Tabuleiro

```
list([_X|_Xs]).
list([]).

giveSpace(N) :-
 (N =:= 0; N =:= 6),
 write(' ').

giveSpace(N) :-
 (N =:= 1; N =:= 5),
write('  ').

giveSpace(N) :-
 (N =:= 2; N =:= 4),
 write('   ').

giveSpace(3) :- write('    ').
giveSpace(7) :- write('            ').
giveSpace(8) :- write('              ').


displayMember([List|Rest]) :-
    write(List),
    displayMember(Rest).

displayMember([]).

displayA('x', N):- N > 3, write('  \\ ').
displayA(X, _):-(X = 'x'; X = '!'), write('    ').
displayA(X, _):- (X = ' '; list(X)) , write('  /\\ ').

displayB('x', N):-  N > 3, write('   \\').
displayB(X, _):-(X = 'x'; X = '!'), write('    ').
displayB(X, _):- (X = ' '; list(X)), write(' /  \\').
```

```
displayC(X, _):-(X = 'x'; X = '!'), write('    ').
displayC(' ',_):- write('|    ').
displayC(List, _):- write('| '), displayMember(List), write('').

displayLineA([X | Xs], Value) :- displayA(X, Value) , displayLineA(Xs, Value).
displayLineA([], N):- N > 3, write(' /'), nl.
displayLineA([], _T):- nl.

displayLineB([X | Xs], Value) :- displayB(X,Value), displayLineB(Xs,Value).
displayLineB([] ,N ):- N > 3, write('/'), nl.
displayLineB([], _T ):- nl.

displayLineC([X | Xs], Value) :- displayC(X, Value), displayLineC(Xs, Value).
displayLineC([], _T ):- write('|'), nl.

displayEnd1(0):-nl , giveSpace(8), displayEnd2(4).
displayEnd1(Counter) :-
      Counter > 0,
      Counter1 is Counter - 1,
      write('\\   /'),
      displayEnd1(Counter1).


displayEnd2(0) :- nl.
displayEnd2(Counter) :-
      Counter > 0,
      Counter1 is Counter - 1,
      write('\\/   '),
      displayEnd2(Counter1).


displayLine(L, Value) :-
        giveSpace(Value),
        displayLineA(L, Value),
        giveSpace(Value),
        displayLineB(L, Value),
        giveSpace(Value),
        displayLineC(L, Value).

displayLine([], _T):- nl.


displayBoard([H | T], Count) :-
    displayLine(H,Count ),
    Count1 is Count + 1,
    displayBoard(T ,Count1).

displayBoard([], Count):- giveSpace(Count), displayEnd1(4).


menu(_X) :-
  tabuleiro(_X),
  displayBoard(_X, 0).
```

Output :



Movimentos

- moveAdaptoid(+Board, + Player, + ListInitialCoords, + ListFinalCoords, - NewBoard)
- addLeg(+Board, + Player, + Coords, -NewBoard)
- addPincer(+Board, +Player, +Coords, -NewBoard)
- getAdaptoid(+Board, +Player, +Coords, -Adaptoid)
- removeAdaptoid(+Board, +Player, +Coords, -NewBoard)
- putAdaptoid(+Board, +Player, +Coords, -NewBoard)