**Breakthru**

**Relatório Intercalar**



Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

**Grupo Breakthru\_1:**

João David Gonçalves Baião - 201305195

Pedro Vieira de Castro - 201305337

Faculdade de Engenharia da Universidade do Porto

Rua Roberto Frias, sn, 4200-465 Porto, Portugal

7 de Outubro de 2014

# O Jogo Breakthru

* 1. **Introdução**

O objectivo deste trabalho é realizar um programa em PROLOG onde será possível jogar um jogo de tabuleiro de forma virtual. Através deste trabalho procuramos adquirir competências na área da Programação em Lógica. O jogo escolhido para ser desenvolvido é o Breakthru.

O Breakthru é um jogo de mesa para duas pessoas. Foi criado por Alex Randolph. Foi comercialmente lançado por 3M Company na década de 60.

**1.2 Regras do Jogo**

O jogo tem como objetivo um dos jogadores conseguir fazer a *MotherShip* atravessar as defesas adversárias e chegar aos limites do tabuleiro. O outro jogador tem que impedir que isto seja possível de se concretizar. O jogador **amarelo** (o tiver que escoltar a *MotherShip)* tem ao seu dispor 12 embarcações para o ajudar enquanto o jogador **cinzento** (defensor) tem 20.

A cada jogada o jogador tem duas opções:

* Move horizontalmente ou verticalmente uma das suas embarcações duas posição ou duas das suas embarcações uma única vez cada uma. Caso o jogador mova a *MotherShip* fica impossibilitado de fazer qualquer outra movimentação.
* Destrói uma embarcação inimiga movendo uma das suas numa direção obliqua.

A posição inicial da *MotherShip* é no centro do tabuleiro. De seguida o jogador **amarelo** deverá escolher as posições iniciais das suas 12 embarcações dentro dos limites do quadrado branco (fig. 1). Posteriormente a esta disposição inicial do jogador **amarelo**, o jogador **cinzento** irá distribuir as suas 20 embarcações na zona exterior ao quadrado branco (fig. 1). Em alternativa os jogadores podem escolher uma distribuição *default* como está apresentado na fig. 1.

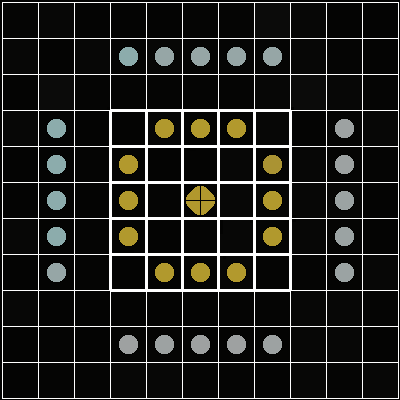


Figura 1

O Jogo acaba quando uma das seguintes condições é atingida:

* O jogador **amarelo** consegue fazer a sua *MotherShip* chegar ao limite do tabuleiro garantido a vitória para si.
* O jogador **cinzento** consegue destruir a *MotherShip*.

# Representação do Estado do Jogo

Sendo o tabuleiro de 11x11 e as listas a melhor estrutura de dados a ser usada em Prolog escolhemos uma lista de listas para representar o tabuleiro. Cada elemento dessa lista será um espaço vazio ou uma peça.

Para distinguir as peças dos dois jogadores, estas têm valores diferentes, sendo que as peças do jogador cinzento correspondem a "1" no tabuleiro, as peças do jogador amarelo são "2" e a *MotherShip*, também esta do jogador amarelo corresponde ao "5". Os espaços vazios são "0". Portanto, a representação inicial do tabuleiro no caso de ter uma disposição default (Fig. 1) corresponderia à seguinte initial\_board.

final\_board(

[[0,0,0,0,1,0,2,0,5,0,0],

[0,0,0,1,0,0,0,0,0,0,0],

[0,1,2,0,0,2,0,0,0,1,0],

[0,0,1,0,0,0,0,0,1,0,0],

[0,0,0,0,0,0,0,0,0,1,0],

[0,0,0,0,0,0,0,0,0,0,0],

[0,1,0,2,0,0,0,2,0,1,0],

[0,1,0,0,1,1,2,0,0,0,0],

[0,0,0,0,0,0,1,0,0,0,0],

[0,0,0,0,0,0,0,1,0,0,0],

[0,0,0,0,0,0,0,0,0,0,0]]).

initial\_board(

[[0,0,0,0,0,0,0,0,0,0,0],

[0,0,0,1,1,1,1,1,0,0,0],

[0,0,0,0,0,0,0,0,0,0,0],

[0,1,0,0,2,2,2,0,0,1,0],

[0,1,0,2,0,0,0,2,0,1,0],

[0,1,0,2,0,5,0,2,0,1,0],

[0,1,0,2,0,0,0,2,0,1,0],

[0,1,0,0,2,2,2,0,0,1,0],

[0,0,0,0,0,0,0,0,0,0,0],

[0,0,0,1,1,1,1,1,0,0,0],

[0,0,0,0,0,0,0,0,0,0,0]]).

O final\_board é um exemplo de um possível final de jogo. Neste caso o jogador **amarelo** ganha visto que a *MotherShip* conseguiu atingir o limite superior do tabuleiro.

# Visualização do Tabuleiro

Para mostrar o tabuleiro aos jogadores irá ser necessá*r*io haver uma impressão do tabuleiro na consola. O tabuleiro impresso na consola irá numerar as colunas e linhas para facilitar as jogadas dos jogadores. Irá também ser impresso uma bussola para ajudar os jogadores a tomarem a suas decisões (fig. 2).

Quanto à simbologia usada a *MotherShip* irá ser simbolizada com um ‘M’, as embarcações do jogador **amarelo** com ‘A’ e as embarcações do jogador **cinzento** com ‘D’.

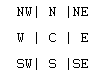
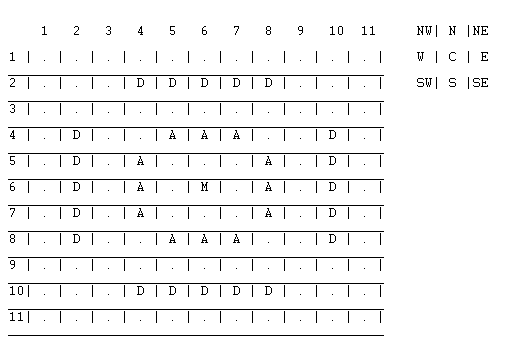
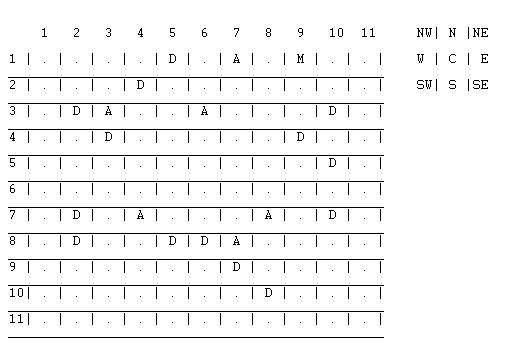


Figura 2 - Bussola

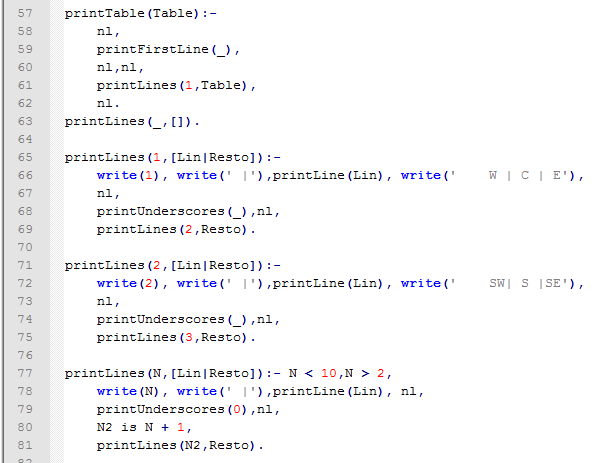


initial\_board

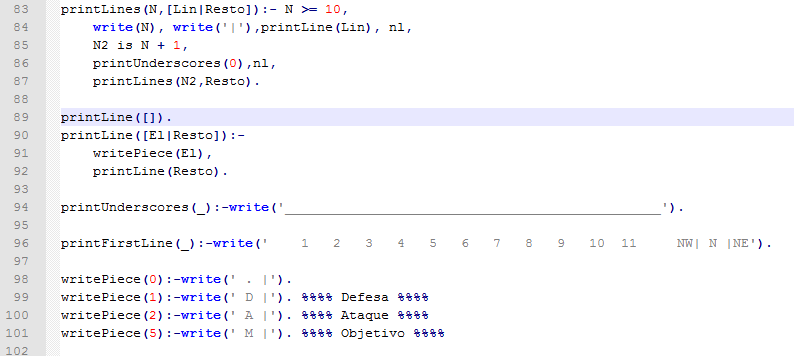


final\_board

Esta impressão vai ser possível correndo o seguinte código sobre as listas previamente apresentadas:



Parte 1



Parte 2

# Movimentos

Para movimentar as peças vão ser pedidas aos jogadores as coordenadas originais da peça em questão e a direção a ser tomada pela peça.

MovePiece(Table,X,Y,Dir,Player,NumPlays)

(NumPlays: numero de jogadas possíveis, partindo do principio que uma moviementação obliqua/*MotherShip* vale 2 jogadas)

MovePiece vai ser responsável por confirmar as seguintes condições:

1. Confirmar que existe uma peça em (X,Y) e que essa peça pertence ao Jogador.
2. Confirmar que a direção é válida (N, S, E, W,NE,NW,SE,SW).
3. Confirmar que se na posição final não se encontra uma peça amigável.
4. Confirmar que o movimento não ultrapassa o número de Jogadas ainda possíveis.

Fontes: https://en.wikipedia.org/wiki/Breakthru\_(board\_game)