

ILIOS

Relatório Intercalar



Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

Grupo Ilios_2:

João Henrique Araújo - 201303962

José Pedro Teles - 201305101

Faculdade de Engenharia da Universidade do Porto

16 de Outubro de 2016

1 O Jogo Ilios

1.1 Contextualização

O Ilios é um jogo de tabuleiro, recomendado para jogadores com mais de 13 anos de idade. É um jogo de estratégia, divertido e fácil de aprender. Cada partida reúne 2 a 4 jogadores e tem a duração prevista de 30 minutos.

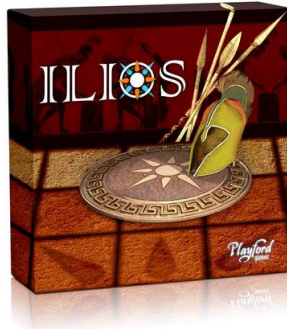


Figura 1: Caixa do Ilios

1.2 Componentes do jogo

- 1 Tabuleiro 6x6, 7x7 ou 8x8;
- 39 peças de jogo (divididas por 6 tipos diferentes);
- 96 discos (24 de cada cor);

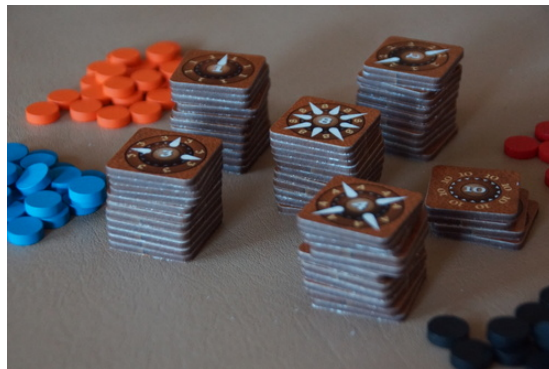


Figura 2: Peças de jogo

1.3 Objetivo do jogo

O objetivo deste jogo é conquistar as peças numeradas que tanto o jogador como o adversário colocam no tabuleiro. O jogo termina quando todo o tabuleiro estiver ocupado, ganhando o jogador que reunir a maior soma dos valores das peças ou, em caso de igualdade, o maior número de peças conquistadas.

1.4 Regras

- Define-se a ordem dos jogadores, o primeiro jogador é aquele que tiver estado mais perto da Grécia. Para jogos com dois jogadores o primeiro jogador recebe duas armas e o segundo uma. Para jogos com mais de dois jogadores, cada um recebe uma arma.
- Cada jogador tira 3 guerreiros, de forma aleatória e secreta;
- No seu turno, cada jogador coloca uma peça no tabuleiro, em qualquer posição livre, o guerreiro tem que apontar para pelo menos um quadrado ocupado por um inimigo (se não conseguir colocar no tabuleiro tem que mostrar o jogo aos outros jogadores antes de jogar numa casa livre);

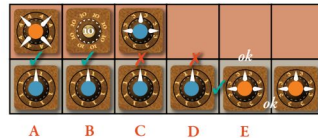


Figura 3: Jogadas válidas

- Um jogador para assaltar as casa dos adversários tem que colocar um guerreiro a apontar para um inimigo conquistando-o de seguida, depois é preciso substituir as peças identificadoras(peças coloridas);
- Quando uma peça estiver completamente cercada por peças identificadoras, o guerreiro pode ser removido do tabuleiro e guardado pelo jogador a que pertence, somando o valor da peça à sua pontuação. No seu lugar é colocado uma peça identificadora do jogador que rodeou o guerreiro removido;
- Para colocar reforços o jogador retira um guerreiro dos que sobraram da partilha inicial e coloca na sua mão;
- O jogo termina quando todo o tabuleiro estiver completo;

2 Representação do Estado do Jogo

Para a representação dos dados do jogo usamos uma lista de listas, cada uma representando uma casa do tabuleiro e tendo três valores para os seguintes respectivos atributos:

- Estado:
 - -1 = Não ocupada ou uma arma ainda não conquistada por um jogador
 - 'O' = Jogador Laranja
 - 'B' = Jogador Azul
 - 'G' = Jogador Verde
 - 'Y' = Jogador Amarelo
- Peça:
 - -1 = Não existe uma peça nessa casa
 - 1, 2, 3, 4, 8 ou 10 = Valor da peça que ocupa essa casa
- Orientação da peça:
 - -1 = Não existe uma peça ou a orientação é indiferente (no caso das peças 4, 8 e 10)
 - 'N' = Norte
 - 'S' = Sul
 - 'W' = Oeste
 - 'E' = Este

2.1 Representação de um estado inicial do tabuleiro

$[[-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1],$
 $[-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1],$
 $[-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1],$
 $[-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1],$
 $[-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1]]$

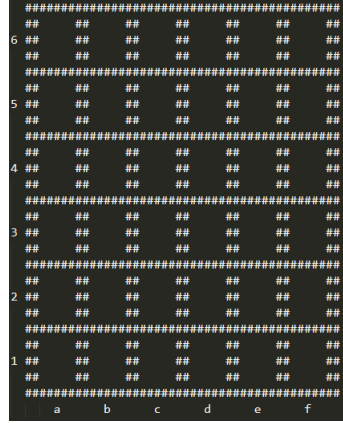


Figura 4: Exemplo de um estado inicial

2.2 Representação de um estado intermédio do tabuleiro

$[[-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1],$
 $[-1, -1, -1], ['O', 2, 'N'], ['B', 3, 'S'], ['B', 10, -1], ['O', 3, 'W'], [-1, -1, -1],$
 $[1, -1, -1], ['B', 1, 'S'], ['O', -1, -1], ['B', -1, -1], ['B', 2, 'N'], [-1, -1, -1],$
 $['B', 8, -1], ['B', 4, -1], ['O', 1, 'N'], ['B', 4, -1], ['B', 2, 'N'], [-1, -1, -1],$
 $['B', 10, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1],$
 $[-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1], [-1, -1, -1]]$

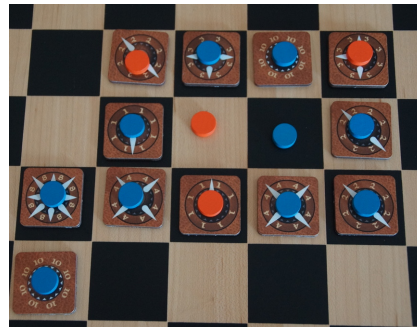
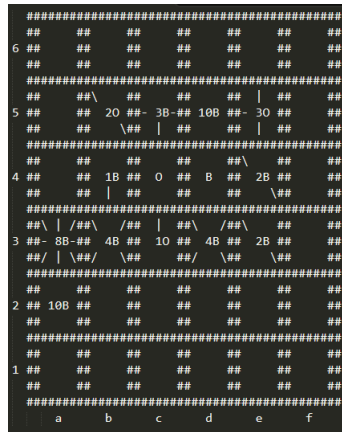


Figura 5: Exemplo de um estado intermédio

2.3 Representação de um estado final do tabuleiro

```
[['B', -1, -1], ['B', -1, -1], ['O', -1, -1], ['O', -1, -1], ['B', -1, -1], ['O', -1, -1],
['B', -1, -1], ['O', -1, -1], ['B', -1, -1], ['B', -1, -1], ['O', -1, -1], ['B', -1, -1],
['O', -1, -1], ['B', -1, -1], ['O', -1, -1], ['B', -1, -1], ['B', -1, -1], ['B', -1, -1],
['B', -1, -1], ['O', -1, -1], ['B', -1, -1], ['O', -1, -1], ['O', -1, -1], ['O', -1, -1],
['O', -1, -1], ['B', -1, -1], ['O', -1, -1], ['B', -1, -1], ['B', -1, -1], ['O', -1, -1]]
```

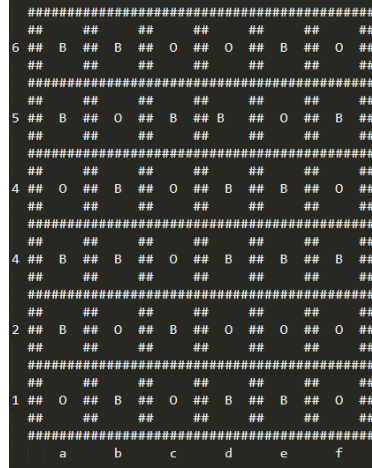


Figura 6: Exemplo de um estado final

3 Visualização do Tabuleiro

A visualização do tabuleiro em modo de texto é feita através da composição de caracteres ASCII, como visto anteriormente nas figuras 4, 5 e 6. Para este efeito, foi desenvolvido um conjunto de predicados.

O predicado principal é:

- **printBoard(List, Size)**: recebe uma lista descritiva das peças que constituem o tabuleiro e imprime-o;

Os predicados auxiliares são:

- **printBorder(N)**: imprime a borda horizontal entre as casas do tabuleiro
- **getElement([Head|Tail], X, Y)**: retorna uma casa do tabuleiro
- **printElement([Head|Tail])**: imprime uma casa do tabuleiro
- **getRow([Head|Tail],[Head|Rest],N,End)**: retorna uma linha do tabuleiro
- **printRow([Head|Tail])**: imprime uma linha do tabuleiro

Cada casa é representada por um quadrado 5x3, usando-se os três predicados seguintes para imprimir uma linha do tabuleiro:

- **printTopRow([Head|Tail])**: imprime a 1ª linha das casas de uma linha
- **printMidRow([Head|Tail])**: imprime a 2ª linha das casas de uma linha
- **printBotRow([Head|Tail])**: imprime a 3ª linha das casas de uma linha
- **convertCode([Cross|Tile], X, Y)**: traduz o átomo em peça de jogo;
 - **convertCodeAux(Cross, [Tile|_], X, Y)**: predicado auxiliar de *convertCode*;
 - **translateCodeToChar(X, Y)**: traduz o código X para o código Y;
- **printInfo(_)**: apresenta informação acerca da representação do jogo;

```
#####
##  ##  ##  ##  ##  ##
6 ##  ##  ##  ##  ##  ##
##  ##  ##  ##  ##  ##
#####
##  ##\  ##  ##  ## |  ##
5 ##  ## 20 ##- 3B-## 10B ##- 30 ##  ##
##  ##  \## |  ##  ## |  ##
#####
##  ##  ##  ##  ##\  ##
4 ##  ## 1B ## 0  ## B  ## 2B ##
##  ## |  ##  ##  ##  \##
#####
##\ | /##\ /## | ##\ /##\  ##
3 ##- 8B-## 4B ## 10 ## 4B ## 2B ##
##/ | \##/ \## ##/ \## \##
#####
##  ##  ##  ##  ##  ##
2 ## 10B ##  ##  ##  ##  ##
##  ##  ##  ##  ##  ##
#####
##  ##  ##  ##  ##  ##
1 ##  ##  ##  ##  ##  ##
##  ##  ##  ##  ##  ##
#####
a b c d e f

Current Score:
BLUE 4 (1p)
ORANGE 10 (1p)

-> ORANGE's turn to play
Current hand:
#####
##\ /# ##\  ## |  #
##- 8 -# # 2  #- 3 -#
##/ \# #  \# #  #
#####
(a) (b) (c)
```

Figura 7: Informação geral apresentada durante cada turno

4 Movimentos

placeWarrior(Board, Player, TileNumber, Orientation).

Função usada para colocar o guerreiro numa posição específica. Falha se for escolhida numa posição inválida.

validMoves(Board, Player, ListOfMoves).

Devolve as jogadas possíveis em *ListOfMoves*.

value(Board, Player, TileNumber, Value).

Avalia a jogada e devolve o seu valor em *Value*.

gameOver(Board, Winner).

Caso o jogo tenha acabado, devolve o vencedor em *Winner*.

chooseMove(Board, Level, Move).

Função em que consoante os diferentes níveis de dificuldade (*Level*) devolve jogadas possíveis.

removeWarrior(Board, Player, TileNumber).

Substitui as peças de jogo pelos marcadores do *Player*.