

ILIOS

Relatório Final



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

16 de Novembro de 2016

Resumo

Foi proposto pelos docentes da unidade curricular Programação em Lógica o desenvolvimento em Prolog, linguagem geralmente utilizada para desenvolvimento de inteligência artificial e linguística computacional, de um jogo de tabuleiro para 2-4 jogadores.

Foram propostos alguns jogos dos quais escolhemos o Ilios. Sendo o jogo uma adaptação de jogos de tabuleiros da antiga grécia, estudamos tudo que era necessário para jogar, regras, quantidades de peças e a sua influência no jogo e por a maneira como tudo seria implementado em Prolog.

Foi implementado um motor de funcionamento do jogo para 2-4 jogadores, sendo computadores ou não, utilizando a linguagem Prolog. A interface gráfica numa primeira fase foi desenvolvida utilizando apenas a consola no entanto numa segunda fase será desenvolvida numa outra unidade curricular (LAIG) uma interface em três dimensões utilizando o Prolog como meio de comunicação entre a parte lógica e a gráfica.

No final conseguimos atingir o objetivo: desenvolvimento da parte lógica de um jogo, de modo a ser possível um futuro desenvolvimento ao nível da interface gráfica.

Concluimos que Prolog é uma linguagem muito interessante ao nível de implementação lógica neste caso de um jogo, pois permite estruturar de uma forma prática e eficaz o código em relação ao prévio planeamento lógico, isto é, antes de qualquer implementação prática.

Conteúdo

1	Introdução	4
2	O Jogo Ilios	5
2.1	Contextualização	5
2.2	Componentes do jogo	5
2.3	Objetivo do jogo	5
2.4	Regras	5
3	Lógica do Jogo	7
3.1	Representação do Estado do Jogo	7
3.2	Visualização do Tabuleiro	9
3.3	Lista de Jogadas Válidas	10
3.4	Execução de Jogadas	10
3.5	Avaliação do Tabuleiro	10
3.6	Final do Jogo	10
3.7	Jogada do Computador	10
4	Interface com o Utilizador	11
5	Conclusões	12
	Bibliografia	13

1 Introdução

Este trabalho tem como objetivo desenvolver um jogo de tabuleiro para duas pessoas ou até mesmo quatro, sendo que o desenvolvimento da parte lógica será através da utilização da linguagem Prolog e a implementação de uma interface gráfica adequada ao jogo, de modo a numa fase posterior possibilitar o desenvolvimento gráfico da aplicação baseando-se no Prolog como meio de comunicação entre diferentes ferramentas que permitem uma ligação entre a parte lógica e gráfica do jogo.

O relatório está estruturado de forma a dar a conhecer o jogo, primeiramente pela enumeração e explicação das suas regras, conceitos e objetivos depois o modo de implementação, a sua explicação lógica e estados de jogo e, por fim, é abordada a interface gráfica.

2 O Jogo Ilios

Nesta secção será explicada a implementação da parte lógica do Ilios como a representação do estado do tabuleiro e sua visualização, execução de movimentos, verificação do cumprimento das regras do jogo, determinação do final do jogo e cálculo das jogadas a realizar pelo computador utilizando diversos níveis de jogo.

2.1 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.

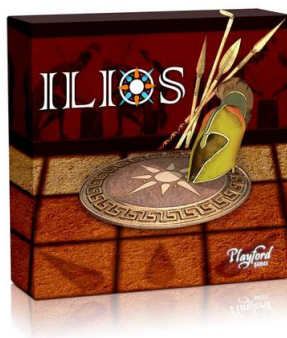


Figura 1: Caixa do Ilios

2.2 Componentes do jogo

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

2.3 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.

2.4 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.

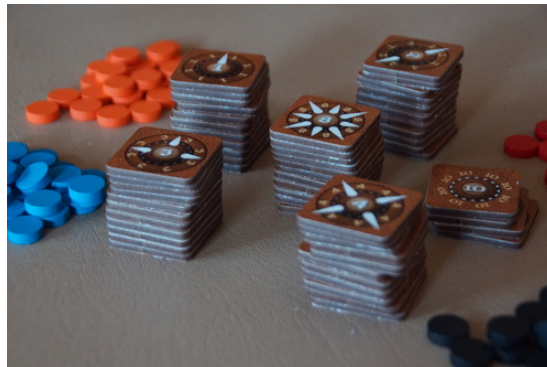


Figura 2: Peças de jogo

- 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);

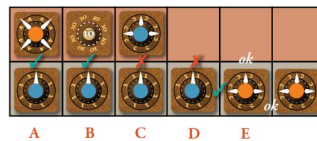


Figura 3: Jogadas válidas

- 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;

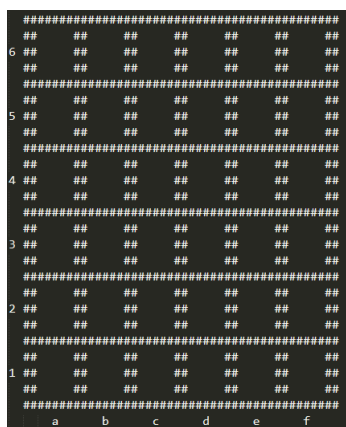
3 Lógica do Jogo

Nesta secção será explicada a implementação da parte lógica do Ilios como a representação do estado do tabuleiro e sua visualização, execução de movimentos, verificação do cumprimento das regras do jogo, determinação do final do jogo e cálculo das jogadas a realizar pelo computador utilizando diversos níveis de jogo.

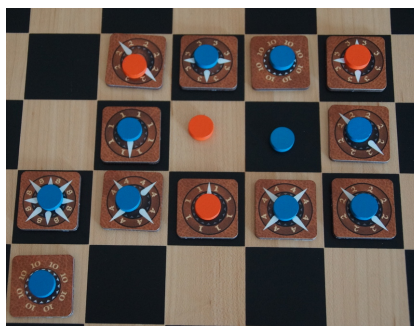
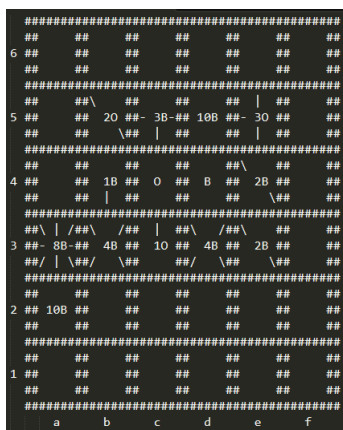
3.1 Representação do Estado do Jogo

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

[illegible]

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



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

6	B	B	O	O	B	O
5	B	O	B	B	O	B
4	O	B	O	B	B	O
4	B	B	O	B	B	B
2	B	O	B	O	O	O
1	O	B	O	B	B	O
	a	b	c	d	e	f

Figura 6: Exemplo de um estado final

3.2 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, N)**: 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(Board,Size,X,Y)**: retorna uma casa do tabuleiro;
- **printElement([Head|Tail])**: imprime uma casa do tabuleiro;
- **printElementTop(Element)**: imprime a 1ª parte de um elemento de uma casa;
- **printElementMid(Element)**: imprime a 2ª parte de um elemento de uma casa;
- **printElementBot(Element)**: imprime a 3ª parte de um elemento de uma casa;
- **printTopRow(Row)**: imprime a 1ª linha das casas de uma linha;
- **printMidRow(Row)**: imprime a 2ª linha das casas de uma linha;
- **printBotRow(Row)**: imprime a 3ª linha das casas de uma linha;
- **printLetters(Size, N)**: imprime letras;

3.3 Lista de Jogadas Válidas

Não foi implementada qualquer função que permitisse validar as jogadas. O próprio motor do jogo faz as verificações necessárias.

3.4 Execução de Jogadas

O predicado `startGame` passa como parâmetros o jogador da próxima jogada. Entre outras, chama o predicado `placePiece` onde passa a linha, a coluna e a nova peça.

```
startGame(NextPlayer).  
placePiece(-1, X,Y,Piece).
```

3.5 Avaliação do Tabuleiro

A cada turno verifica todas as jogadas possíveis e guarda-as na variável `gamePlays/2`. O `gamePlays` é constituído por duas listas, uma onde tem os valores de cada jogada ordenados por ordem crescente, e outra com as jogadas em si.

3.6 Final do Jogo

A verificação do vencedor do jogo ocorre aquando do total preenchimento do tabuleiro de jogo. O vencedor é o jogador com mais pontos ou, em caso de empate, o que tiver conquistado mais peças.

3.7 Jogada do Computador

O jogo apresenta bots com dois níveis de dificuldade o fácil, que joga de forma aleatória e o difícil que avalia todas as jogadas possíveis e escolhe a melhor possível.

4 Interface com o Utilizador

De modo a criar uma interface gráfica em modo de texto, foi criado um ficheiro apenas direccionado para esta questão: `Ilios.pl`. Neste ficheiro encontra-se implementada uma função que imprime o menu do jogo, onde é possível escolher o modo do jogo assim com um pequeno tutorial.

Existem variadíssimas funções que chamadas imprimem o tabuleiro sejam as legendas alfanuméricas, linhas, barras, espaços em branco ou os caracteres de jogada.

5 Conclusões

Concluimos que a linguagem Prolog é muito útil e propicia no desenvolvimento lógico de um jogo, permitindo estruturar de uma forma mais conseguida e correta a ideia do jogo em si, oferecendo ferramentas de desenvolvimento muito úteis.

Gostaríamos de ter tido tempo para implementar acrescentar o que podera ficar por fazer.

Bibliografia

Disponível em: http://www.modestgames.com/ilios/ILIOS_rules_KSv2.pdf.