Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

Quantick

Relatório Intermédio

3MIEIC06 - Quantik4

Andreia Gouveia up201706430@fe.up.pt
João Filipe Carvalho de Araújo up201705577@fe.up.pt

Outubro 2019

Index

1.Introdução

1.1 História do jogo

Quantik foi inventado por Nouri Khalifa . O jogo foi publicado e distribuído pela Gigamic em setembro de 2019.

1.2 Constituição do jogo

É um jogo para 2 pessoas. Cada jogador tem à sua disposição 8 peças (2 cubos, 2 cilindros, 2 cones e 2 esferas), sendo que cada jogador é diferenciado pela cor das suas peças (podendo ser brancas ou pretas). O jogo também inclui um tabuleiro,dividido em 4 quadrantes, que possui 16 posições onde os jogadores vão poder colocar as suas peças.

Figura 1: Imagem do tabuleiro

1.3 Regras e objectivos do jogo

As regras do jogo são relativamente simples. A cada ronda, os jogadores vão colocar uma das suas peças no tabuleiro em espaços vazios. É proibido colocar uma peça de determinada forma, numa linha, coluna ou quadrante onde já exista uma peça com essa mesma forma do adversário. Caso a peça repetida seja do mesmo jogador, então a jogada é válida. O primeiro jogador a conseguir ter uma fila, coluna ou quadrante do tabuleiro com todas as formas diferentes (independentemente se forem ou não da cor do jogador) ganha o jogo imediatamente.

2.Representação interna

2.1 Tabuleiro

O nosso tabuleiro é representado por uma matriz de 4x4. Ou seja, uma lista de listas de chars.

```
initialBoard(
            [[empty,empty,empty,empty],[empty,empty,empty,empty],[empty,empty,empty,empty],[empty,empty,empty,empty]]).
```

Inicialmente o tabuleiro está vazio, sendo que o vazio (ou seja, sem a presença de peças) é representado pelo char ‘*’. Este char é representado pelo átomo empty.

2.2 Peças

Existem no total 16 peças neste jogo, todas representadas por chars. Estas 16 dividem-se em 8 pretas (representadas por letras minúsculas) e 8 brancas (representadas por letras maiúsculas). Cada jogador terá 2 cones(‘p’ ou ‘P’ dependendo da cor) , 2 cubos(‘c’ ou ‘C’ dependendo da cor), 2 esferas(‘e’ ou ‘E’ dependendo da cor) e 2 cilindros(‘l’ ou ‘L’ dependendo da cor).

Segue-se o código que representa esta atribuição:

```
piece(empty, V) :- V = '*'.
piece(coneB, V) :- V ='p'.
piece(coneW, V) :- V ='P'.
piece(cubeB, V) :- V ='c'.
piece(cubeW, V) :- V ='C'.
piece(cylinderB, V) :- V ='l'.
piece(cylinderW, V) :- V ='L'.
piece(sphereB, V) :- V ='e'.
piece(sphereW, V) :- V ='E'.
```

2.3 Jogadores

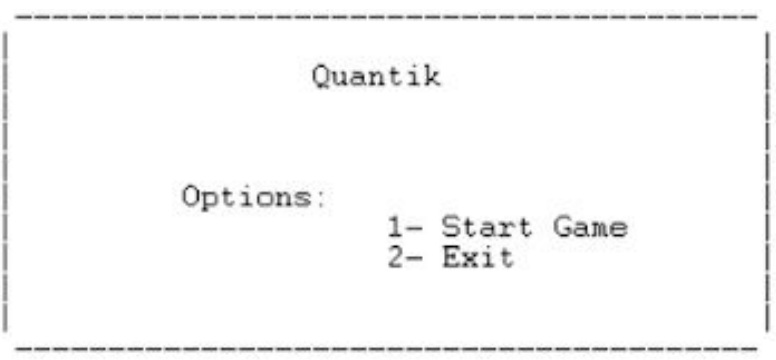
Os jogadores são representados por átomos denominados de white e black, com respectivo valor de 1 e 2. Por “default”, colocamos o jogador branco a ser o primeiro a jogar.

```
player(white , V) :- V = 1.
player(black , V) :- V = 2.
```

3.Visualização

Quando o jogo é inicializado, surge um menu principal que dá ao jogador 2 opções:

- Começar o jogo
- Sair do jogo

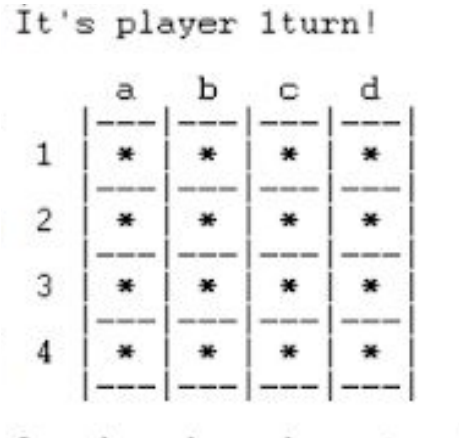


Segue-se o código responsável pelo display do menu principal:

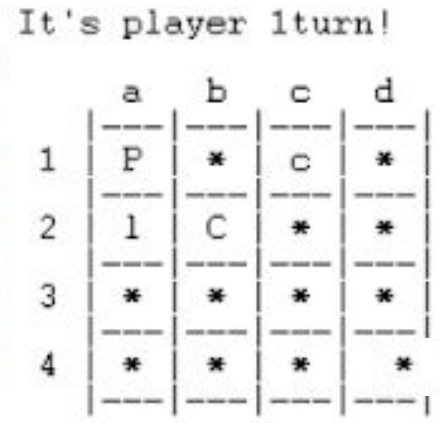
```
menus:-
        displayMainMenu,
        nl,
        read(Input),
        menuChoice(Input).
```

```
displayMainMenu:-
        write('\n ---------------------------------------\n'),
        write('|                                       |\n'),
        write('|               Quantik                 |\n'),
        write('|                                       |\n'),
        write('|                                       |\n'),
        write('|                                       |\n'),
        write('|        Options:                       |\n'),
        write('|                  1- Start Game        |\n'),
        write('|                  2- Exit              |\n'),
        write('|                                       |\n'),
        write('|                                       |\n'),
        write(' ---------------------------------------\n').
```

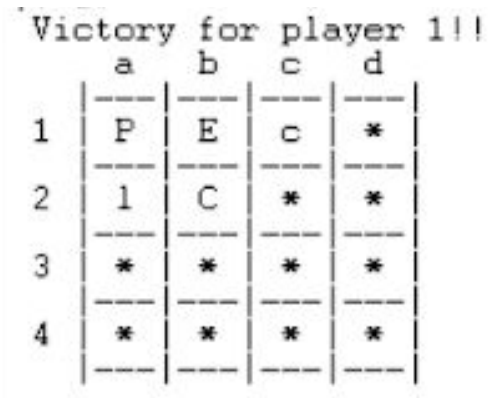
Caso a opção escolhida seja começar o jogo, é apresentado ao jogador um tabuleiro vazio. Em cima do tabuleiro, o programa indica de quem é a vez de jogar.



A imagem em cima, refere-se ao tabuleiro num estado inicial. Num estado mais avançado do jogo, iremos encontrar um tabuleiro do género:



No final do jogo, neste caso, numa situação de vitória, o tabuleiro poderia se apresentar da seguinte maneira:



Segue-se o código responsável pelo display do tabuleiro:

```
showBoard(Player, Board) :-
        player(Player,Num),
        write( 'It\'s player '),
        write( Num ),
        write( 'turn!\n' ),
        printBoard(Board).

printBoard(Board):-
        write('\n     a   b   c   d  '),
        write('\n   |---|---|---|---|') ,
        printBoard(Board, 1).

printBoard([],_):-
        nl,
        nl.

printBoard([H|T],Num):-
        write('\n '),
        write(Num),
        printLine(H),
        nl,
        write('   |---|---|---|---|'),
        Num1 is Num+1,
        printBoard(T,Num1).

printLine([]):-
        write(' |').

printLine([H|T]):-
        write(' | '),
        piece(H,Piece),
        write(Piece),
        printLine(T).
```

Caso a opção escolhida seja sair do jogo, é apresentada na consola uma mensagem a agradecer ao jogador por ter jogado.

```
Thanks for playing!
```

Segue-se o código responsável pela apresentação desta mensagem:

```
menuChoice(2):-
        write('Thanks for playing!').
```

4.Conclusão

O objetivo desta entrega intermédia foi dar a conhecer a base de funcionamento da linguagem Prolog, sendo que os principais pontos a atingir eram conhecer o funcionamento do jogo e implementar a estrutura básica do mesmo.

Sentimos algumas dificuldades inicialmente a perceber o funcionamento desta nova linguagem, mas com a prática e com ajuda das aulas conseguimos superá-las.

Tendo em conta os objectivos, achamos que conseguimos atingir o pretendido nesta entrega.

5.Bibliografia

- https://en.1jour-1jeu.com/people/nouri-khalifa/designer
- https://en.1jour-1jeu.com/boardgame/2019-quantik/
- https://en.gigamic.com/game/quantik