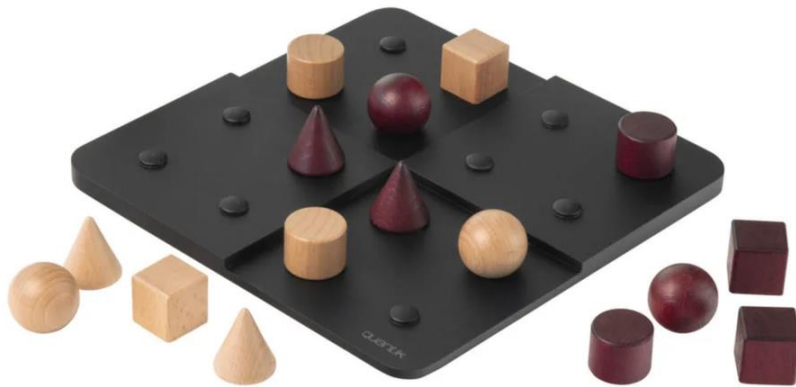


Programação Lógica

Quantik

Aplicação em Prolog para um Jogo de Tabuleiro



Grupo Quantik_1:

João Alberto Preto Rodrigues Praça - up201704748

Lucas Tomás Martins Ribeiro - up201705227

Outubro 2019

Introdução

Este projeto no âmbito da Unidade Curricular de Programação Lógica consiste no desenvolvimento de um jogo de tabuleiro para dois jogadores, em linguagem Prolog. O jogo escolhido foi o *Quantik*, e deverá contar com três modos de utilização: Humano contra Humano, Humano contra Computador e Computador contra Computador, tendo o Computador dois níveis de dificuldade.

Descrição do jogo

História

O *Quantik* é um jogo desenvolvido pela publicadora *Gigamic*, de origem francesa, fundada por 3 irmãos e tendo já criado mais de 400 jogos de tabuleiro que foram adaptados para mais de 45 países.

O designer do jogo foi *Nouri Khalifa*, o mesmo criador dos jogos *Damix* e *Ordo*.



O jogo está indicado como pertencente ao género de estratégia abstrata, sendo descrito como um jogo para dois jogadores, com um tempo de duração média de 20 minutos e recomendado para qualquer pessoa com mais de 8 anos.

Regras

O tabuleiro do jogo divide-se em 4 quadrantes, sendo que em cada um existem 4 pontos onde se podem posicionar as peças, podendo assim dividir-se também o tabuleiro em 4 colunas e 4 linhas.

As peças do jogo consistem em sólidos geométricos, podendo tomar a forma de um cone, cubo, cilindro ou esfera. As peças podem também ser claras ou escuras dependendo do jogador a que pertencem. Cada jogador tem inicialmente direito a 8 peças: duas de cada sólido geométrico.

O objetivo do jogo é formar uma linha, coluna ou preencher um quadrante com 4 peças diferentes. Cada jogador coloca uma peça de cada vez, sendo proibido colocar uma peça numa linha, coluna ou quadrante que já tenha uma peça do oponente com a mesma forma. Apenas é possível repetir formas caso ambas sejam do mesmo jogador. Ganha quem colocar a quarta peça que completa um conjunto, independentemente de quem colocou as restantes peças que formam o conjunto de quatro peças distintas.

Abordagem inicial à modelação do jogo

Representação interna do estado do jogo

O estado do tabuleiro será representado em Prolog através de uma lista de listas, sendo que cada lista dentro da lista principal representa uma linha do tabuleiro, e em que o índice de cada elemento dentro da lista de segundo nível representa a coluna a que este pertence.

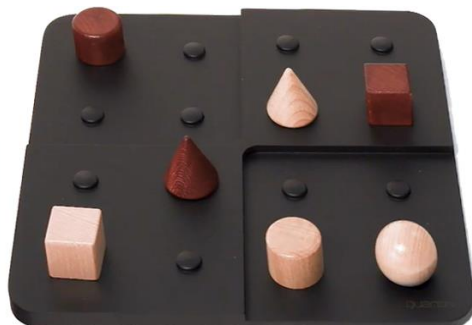
Cada jogador será representado por uma lista, sendo o primeiro elemento correspondente ao número do jogador (1 ou 2), e os restantes elementos referentes às peças que o jogador ainda tem disponíveis.

Representação do estado inicial do tabuleiro:



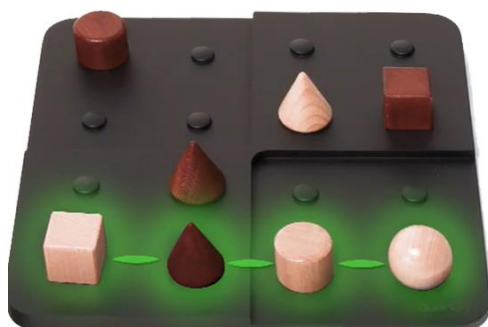
```
[[blank, blank, blank, blank],  
[blank, blank, blank, blank],  
[blank, blank, blank, blank],  
[blank, blank, blank, blank]]
```

Representação de um possível estado intermédio do tabuleiro:



```
[[by, blank, blank, blank],  
[blank, blank, wo, bc ],  
[blank, bo, blank, blank],  
[wc, blank, wy , ws  ]]
```

Representação de um possível estado final do tabuleiro:



```
[[by, blank, blank, blank],  
[blank, blank, wo, bc ],  
[blank, bo, blank, blank],  
[wc, bo, wy , ws  ]]
```

Visualização do tabuleiro em modo de texto

O seguinte código é um protótipo do código utilizado para visualizar o tabuleiro:

```
draw_column_letters :- write('      a      b      c      d      \n').
get_piece(blank, X) :- X='..'.
get_piece(by, X) :- X='BY'.
get_piece(bo, X) :- X='BO'.
get_piece(bc, X) :- X='BC'.
get_piece(bs, X) :- X='BS'.
get_piece(wy, X) :- X='WY'.
get_piece(wo, X) :- X='WO'.
get_piece(wc, X) :- X='WC'.
get_piece(ws, X) :- X='WS'.
write_row_number(N):-
    write(' '),
    print(N),
    write(' ').
draw_separator:- write('      -----\n').
draw_line([H|T]) :-
    write(' '),
    get_piece(H, X),
    print(X),
    write(' |'),
    draw_line(T).
draw_line([]).
draw_board([H|T], Z) :-
    write_row_number(Z),
    write('|'),
    draw_line(H),
    write('\n'),
    draw_separator,
    X is Z+1,
    draw_board(T, X).
draw_board([],_).

draw_number(N):-
    write('Player '),
    print(N),
    write('\n').
draw_pieces([]) :-
    write('\n\n').
draw_pieces([H|T]):-
    get_piece(H, X),
    print(X),
    write(' '),
    draw_pieces(T).
draw_player([H|T]):-
    draw_number(H),
    write('Pieces : '),
    draw_pieces(T).
display_game(Board,Player) :-
    write('\n'),
    draw_column_letters,
    draw_separator,
    draw_board(Board, 1),
    draw_player(Player).
```

A seguinte imagem é um exemplo da aparência do tabuleiro em modo de texto, em que foram utilizados os argumentos [[by, bc, bs, bo], [blank, blank, blank, blank], [blank, blank, blank, blank], [blank, blank, blank, blank]] e [1, by, bc, bs, bo], para o Board e Player, respetivamente.

```
      a      b      c      d
-----
1 |  BY |  BC |  BS |  BO |
-----
2 |  .. |  .. |  .. |  .. |
-----
3 |  .. |  .. |  .. |  .. |
-----
4 |  .. |  .. |  .. |  .. |
-----

Player 1
Pieces : BY BC BS BO
```

Bibliografia

- <https://en.wikipedia.org/wiki/Gigamic>
- <https://www.boardgamegeek.com/boardgame/286295/quantik>
- <https://en.gigamic.com/game/quantik>