A thick dark gray vertical bar is on the left. A brown arrow points right from it, containing the date.

15-10-2019

Relatório de avaliação intercalar

Programação em Lógica

quantik

Several thin, curved, light gray lines sweep upwards from the bottom left corner.

João Henrique Afonso Marques Reguengo da Luz - 201703782

Liliana Natacha Nogueira Almeida - 201706908

Índice

Sobre o jogo	2
Representação interna do estado do jogo.....	3
Estado inicial:	3
Estado intermédio:.....	3
Estado final:	3
Visualização do tabuleiro em modo de jogo.....	4

Sobre o jogo

Quantik é um jogo de tabuleiro puramente estratégico lançado em agosto de 2019. O objetivo de cada jogador é ser o primeiro a colocar uma peça que forme uma linha, coluna ou quadrante constituído pelas quatro peças distintas.

Por cada jogador são distribuídas duas peças de cada sólido geométrico diferente: 2 cilindros, 2 cubos, 2 cones e 2 esferas. Em cada jogada, os jogadores colocam alternadamente uma das suas peças no tabuleiro. Não é permitido colocar uma peça numa linha, coluna ou quadrante no qual já exista uma peça do adversário com a mesma forma. No entanto, a peça pode ser colocada se ambas pertencerem ao mesmo jogador.

O jogo termina quando um dos jogadores colocar a quarta peça distinta numa linha, coluna ou quadrante, independentemente das peças pertencerem todas a esse jogador.

Source: <https://www.boardgamegeek.com/boardgame/286295/quantik>

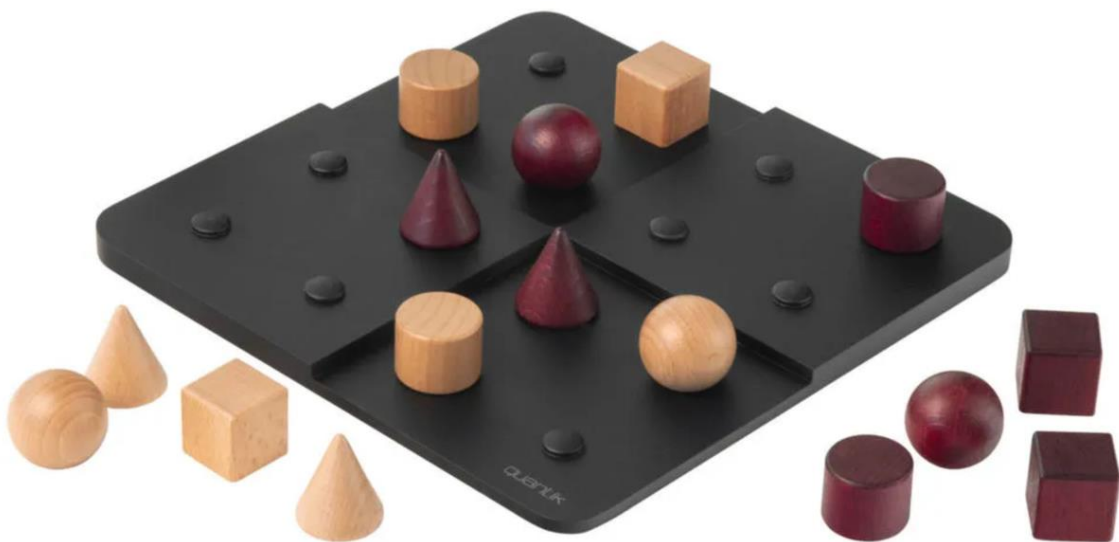


Figura 1 - Exemplo do tabuleiro de um jogo a decorrer

Representação interna do estado do jogo

CADA TIPO DE PEÇA TEM UMA NOMENCLATURA ÚNICA QUE INDICA DE UMA FORMA INTUITIVA O TIPO DE SÓLIDO E O JOGADOR A QUE A PEÇA PERTENCE.

A correspondência entre os átomos e as peças é a seguinte:

wCyl → Cilindro branco	bCyl → Cilindro preto
wCub → Cubo branco	bCub → Cubo preto
wCon → Cone branco	bCon → Cone preto
wSph → Esfera branca	bSph → Esfera preta

Estado inicial:

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

Figura 2 - Representação em Prolog do tabuleiro do jogo no estado inicial.

Board			

Figura 3 - Representação visual do estado inicial do tabuleiro.

Estado intermédio:

```
mid_board([
    [empty, cube_w, empty, cil_b],
    [cil_w, sph_b, empty, empty],
    [empty, cone_b, cone_b, sph_w],
    [empty, empty, cil_w, empty]
]).
```

Figura 3 - Representação em Prolog do tabuleiro do jogo numa fase intermédia.

Board			
	wCub		bCyl
wCyl	bSph		
	bCon	bCon	wSph
		wCyl	

Figura 2 - Representação visual de um estado intermédio do tabuleiro.

Estado final:

```
final_board([
    [empty, cube_w, empty, cil_b],
    [cil_w, sph_b, empty, empty],
    [empty, cone_b, cone_b, sph_w],
    [empty, empty, cil_w, cube_w]
]).
```

Figura 5 - Representação em Prolog do tabuleiro do jogo numa possível fase final.

Board			
	wCub		bCyl
wCyl	bSph		
	bCon	bCon	wSph
		wCyl	wCub

Figura 4 - Representação visual de um estado final do tabuleiro.

Visualização do tabuleiro em modo de jogo

Para visualizar o tabuleiro em modo de texto é necessário utilizar o predicado *display_game* que recebe como argumentos o tabuleiro num dos seus estados e o próximo jogador a colocar uma peça. O output deste predicado pode ser visto nas figuras 3, 5 e 7.

```
display_game(Board, Player):-  
    display_board(Board),  
    nl,  
    write('Next player: '),  
    write(Player).
```

Figura 6 - Predicado de visualização do tabuleiro do jogo.