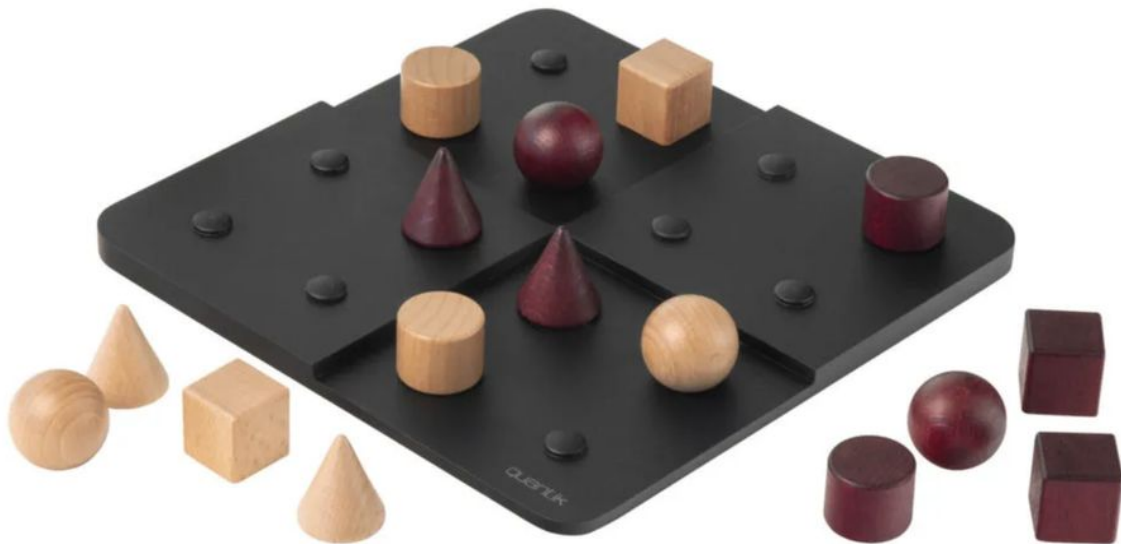


Quantik

Relatório intercalar realizado no âmbito da disciplina de programação em lógica que integra o terceiro ano do Mestrado Integrado em Engenharia Informática e Computação na FEUP



Índice

Descrição do Jogo	3
Tabuleiro	3
Peças	3
Objetivo	3
Regras	4
Representação Interna do Estado do Jogo	4
Estado Inicial	4
Estado Intermédio	4
Estado Final	5
Execução de uma jogada	6
Código desenvolvido	7
Bibliografia	7

Descrição do Jogo

Quantik é um jogo de tabuleiro de estratégia puramente abstrato para dois jogadores. Este jogo foi desenhado por Nouri Khalifat e lançado no ano de 2019 pela desenvolvedora Gigamic.

Tabuleiro

O jogo é jogado num tabuleiro dividido em quatro zonas iguais, cada uma com 2 linhas e 2 colunas, num total de 16 células.

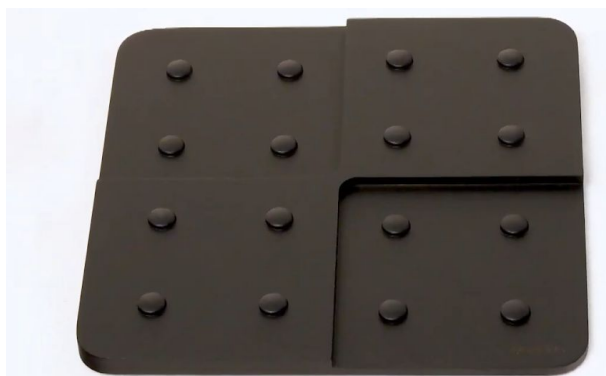


Figura 1 - Tabuleiro vazio.

Peças

Cada jogador tem em sua posse 8 peças de 4 formas diferentes: 2 cubos, 2 cilindros, 2 esferas e 2 cones da sua cor.



Figura 2 - Imagem ilustrativas das peças do jogo.

Objetivo

O objetivo do jogo é ser o primeiro jogador a ocupar uma linha, coluna ou zona quadrada com as quatro diferentes peças: o cilindro, a esfera, o cubo e o cone. As peças que formam a linha, coluna ou quadrado vencedor não têm que ser todas do mesmo jogador.

Regras

Em cada jogada, os jogadores colocam uma das suas peças no tabuleiro. Os jogadores devem jogar alternadamente, no entanto, é possível que um jogador jogue duas vezes seguidas se o adversário não tiver nenhuma jogada possível.

Ao colocar uma peça no tabuleiro, cada jogador deve respeitar a seguinte regra: não é permitido colocar uma peça numa coluna, linha ou região quadrada na qual o adversário já tenha colocado uma peça da mesma forma.

Representação Interna do Estado do Jogo

Será usada uma lista de listas para representar o tabuleiro. Usando átomos para representar o estado de uma célula:

- empty - não ocupada;
- cylinder_<cor> - ocupada por um cilindro;
- sphere_<cor> - ocupada por uma esfera;
- cone_<cor> - ocupada por um cone;
- cube_<cor> - ocupada por um cubo.

Onde está a tag <cor> deverá estar a cor das peças do jogador que ocupou a respectiva célula, podendo ser white ou brown.

Estado Inicial

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

	1	2	3	4
1	X	..
2	X	..
3	X	..
4	X	..

LEGEND:

Piece:	<solid><color>
Solids	- Symbol
Cube	- M
Sphere	- O
Cone	- A
Cylinder	- H

Colors	- Symbol
Brown	- b
White	- w

Figura 3 - Representação do tabuleiro vazio no SISctus

Estado Intermédio

```
board([
  [sphere_white, empty, empty, cylinder_white],
  [cube_brown, empty, empty, empty],
  [empty, cube_white, cone_brown, empty],
  [empty, empty, empty, cube_brown]
]).
```

(Nota: O próximo jogador seria o que possui as peças brancas.)

	1	2	3	4
1	X Ow	.. X ..	Hw X	X
2	X Mb	.. X X	X
3	X ..	Mw X Ab	.. X	X
4	X X ..	Mb X	X

LEGEND:
 Piece: <solid>_<color>
 Solid - Symbol Color - Symbol
 Cone - A Brown - b
 Cube - M White - w
 Cylinder - H
 Sphere - O

Figura 4 - Representação do tabuleiro num estado intermédio no SISCTus

Estado Final

```
board([
  [sphere_white, empty, empty, cylinder_white],
  [cube_brown, cylinder_white, empty, cone_white],
  [empty, cube_white, cone_brown, sphere_brown],
  [empty, empty, empty, cube_brown]
]).
```

O vencedor seria o jogador das peças brancas, vencendo quando coloca a peça cone_white no tabuleiro, fazendo assim uma linha de figuras diferentes, tanto com peças suas como do adversário.

No SICStus, o resultado da visualização do tabuleiro seria este:

	1	2	3	4
1	X Ow	.. X ..	Hw X	X
2	X Mb	Hw X ..	Aw X	X
3	X ..	Mw X Ab	Ob X	X
4	X X ..	Mb X	X

LEGEND:
 Piece: <solid>_<color>
 Solid - Symbol Color - Symbol
 Cone - A Brown - b
 Cube - M White - w
 Cylinder - H
 Sphere - O

Figura 5 - Representação do tabuleiro no estado final no SISCTus

Execução de uma jogada

Para jogar uma peça o jogador terá que colocar “play(1)” ou “play(2)”, consoante é o jogador 1 ou o jogador 2. Se o jogador 1 tentar jogar duas vezes, o Sisc responderá que não. Caso seja a vez do outro jogador, o SISctus perguntará qual a peça que deseja jogar e em que linha e coluna a deseja pôr.

```
| ?- play(1).  
What piece do you want to play?  
cone_brown.  
In which row?  
|: 1.  
In which collumn?  
|: 1.■
```

Figura 6 - Jogada do jogador 1 no SISctus

Código desenvolvido

```
play(P) :-  
    nextPlayer(P),  
    board(X),  
    write('What piece do you want to play?\n '),  
    read(Piece),  
    write('In which row?\n'),  
    read(Row),  
    write('In which collumn?\n'),  
    read(Column),  
    playPiece(Row, Column, Piece, X, Y),  
    assert(board(Y)),  
    clearEverything,  
    display_game(Y),  
    set_next_player(P).  
  
set_next_player(1) :-  
    assert(nextPlayer(2)).
```

Figura 7 - Código desenvolvido para um jogador efetuar uma jogada

Bibliografia

1. 1jour-1jeu.com. (2019). *Quantik (2019) - Board games - 1jour-1jeu.com*. [online] Available at: <https://en.1jour-1jeu.com/boardgame/2019-quantik/> [Accessed 1 Oct. 2019].
2. BoardGameGeek. (2019). *Quantik*. [online] Available at: <https://boardgamegeek.com/boardgame/286295/quantik/credits> [Accessed 6 Oct. 2019].
3. Mastersofgames.com. (2019). *Quantik by Gigamic | Abstract Strategy Game for 2 Players*. [online] Available at: <https://www.mastersofgames.com/cat/board/gigamic-quantik-game.htm> [Accessed 7 Oct. 2019].
4. YouTube. (2019). *GIGAMIC - QUANTIK*. [online] Available at: <https://www.youtube.com/watch?v=Ft-4FjpZG7Q&> [Accessed 9 Oct. 2019].