

Manual de Utilizador

Projeto de Inteligência Artificial

Blokus Uno - Primeira Fase

Autores:

Tiago Farinha (201802235)

Francisco Moura (201802033)

Docente:

Eng. Filipe Mariano

Índice do Manual

- [Introdução](#)
- [Características do Jogo](#)
 - [Peças](#)
 - [Tabuleiro](#)
 - [Regras do Jogo](#)
- [Execução](#)
- [Estatísticas](#)

Introdução

Este Manual de Utilizador tem como objetivo mostrar ao *end user* como utilizar o nosso programa para resolver o problema do Blokus Uno.

Características

Peças

O jogo tem 4 peças disponíveis:

A - Quadrado 1x1



B - Quadrado 2x2



C-1 - S variação 1



C-2 - S variação 2



Tabuleiro

Existem 5 tabuleiros disponíveis para escolher mais um sexto completamente vazio:

[illegible]

Regras do Jogo

- Os jogadores escolhem uma das suas peças e colocam-nas de modo a que um dos quadrados da peça cubra um dos quadrados de canto do tabuleiro de jogo (posição inicial).
- As jogadas são feitas à vez e, em cada turno, o jogador coloca uma peça de modo a que toque pelo menos numa das suas peças já existente, mas apenas nos cantos. Peças do mesmo jogador nunca se podem tocar nas laterais, mas podem tocar lateralmente em outras peças.
- Uma vez colocada, a posição da peça não poderá ser alterada até ao final do jogo.
- Quando um dos jogadores não consegue colocar uma peça no tabuleiro de jogo, deverá passar a vez.
- O jogo termina quando nenhum dos jogadores consegue colocar mais peças.
- É definido um objetivo para o jogador atingir, em termos do número mínimo de casas preenchidas.

Execução

O programa inicia-se com o comando (**iniciar**):

```
CL-USER 1 > (iniciar)
```

```
|          Blokus Uno          |
|                               |
|      1 - Resolver problemas  |
|      2 - Ver Tabuleiros      |
|      0 - Sair                |
|                               |
>
```

Quando introduzida a opção 1, **1 - Resolver problemas**, o utilizador será levado para a página de escolha do problema:

$$> 1$$

```
|          Blokus Uno
|
|      Escolha um problema:
|          1 - A
|          2 - B
|          3 - C
|          4 - D
|          5 - E
|          6 - F
|          0 - Voltar
|_____
>
```

Caso tivesse escolhido a opção 2, **2 - Ver Tabuleiros**, seria levado para a página de visualização desses mesmos:

Blokus Uno													
Tabuleiros (Desafios) :													
A													
(0	0	0	0	2	2	2	2	2	2	2	2	2	2)
(0	0	0	0	2	2	2	2	2	2	2	2	2	2)
(0	0	0	0	2	2	2	2	2	2	2	2	2	2)
(0	0	0	0	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
B													
(0	0	0	0	0	0	0	2	2	2	2	2	2	2)
(0	0	0	0	0	0	0	2	2	2	2	2	2	2)
(0	0	0	0	0	0	0	2	2	2	2	2	2	2)
(0	0	0	0	0	0	0	2	2	2	2	2	2	2)
(0	0	0	0	0	0	0	2	2	2	2	2	2	2)
(0	0	0	0	0	0	0	2	2	2	2	2	2	2)
(0	0	0	0	0	0	0	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)
(2	2	2	2	2	2	2	2	2	2	2	2	2	2)

Após escolher o problema, será levado para a página de escolha do algoritmo de resolução:

> 1

```
|          Blokus Uno          |
|                               |
|      Escolha um algoritmo:   |
|      1 - BFS                 |
|      2 - DFS                 |
|      3 - DFS (modificado)    |
|      0 - Voltar              |
|                               |
```

Aqui são apresentadas duas opções de resolução, **BFS** e **DFS**

- Introduzida a opção 1, é selecionado o algoritmo *Breath-first Search*. Este algoritmo procura pela solução na lista de nós abertos de forma recursiva, ao mesmo tempo que vai expandindo cada um dos nós em que entra sequencialmente, colocando-os no final da lista de nós abertos. Fará isto até encontrar a solução.
- Introduzida a opção 2, é selecionado o algoritmo *Depth-first Search*. Este algoritmo procura pela solução no primeiro nó expandido da lista de nós abertos recursivamente, expandindo esse, colocando os seus sucessores no início da lista aberta e expandindo novamente o primeiro nó dessa lista. Fará isto até atingir a profundidade máxima pretendida. Caso a solução não seja encontrada até esse valor, repetirá o processo para o nó aberto seguinte que esteja na lista e não ultrapasse a profundidade máxima. O utilizador deve especificar a profundidade máxima pretendida:

> 2

Qual a profundidade limite (Insira 0 para voltar atrás)?
>

- A opção 3 apenas corre o algoritmo DFS com 100 de profundidade máxima, assegurando a resolução de todos os problemas.

Estatísticas

Depois de executados os algoritmos, são guardadas as estatísticas desses no caminho *C:\Projeto IA\estatisticas.dat*

10	(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
11	(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
12	(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
13	(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
14	(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)