

Manual de Utilizador



Unidade Curricular: Inteligência Artificial 2017/2018

Andreia Pereira nº 150221021

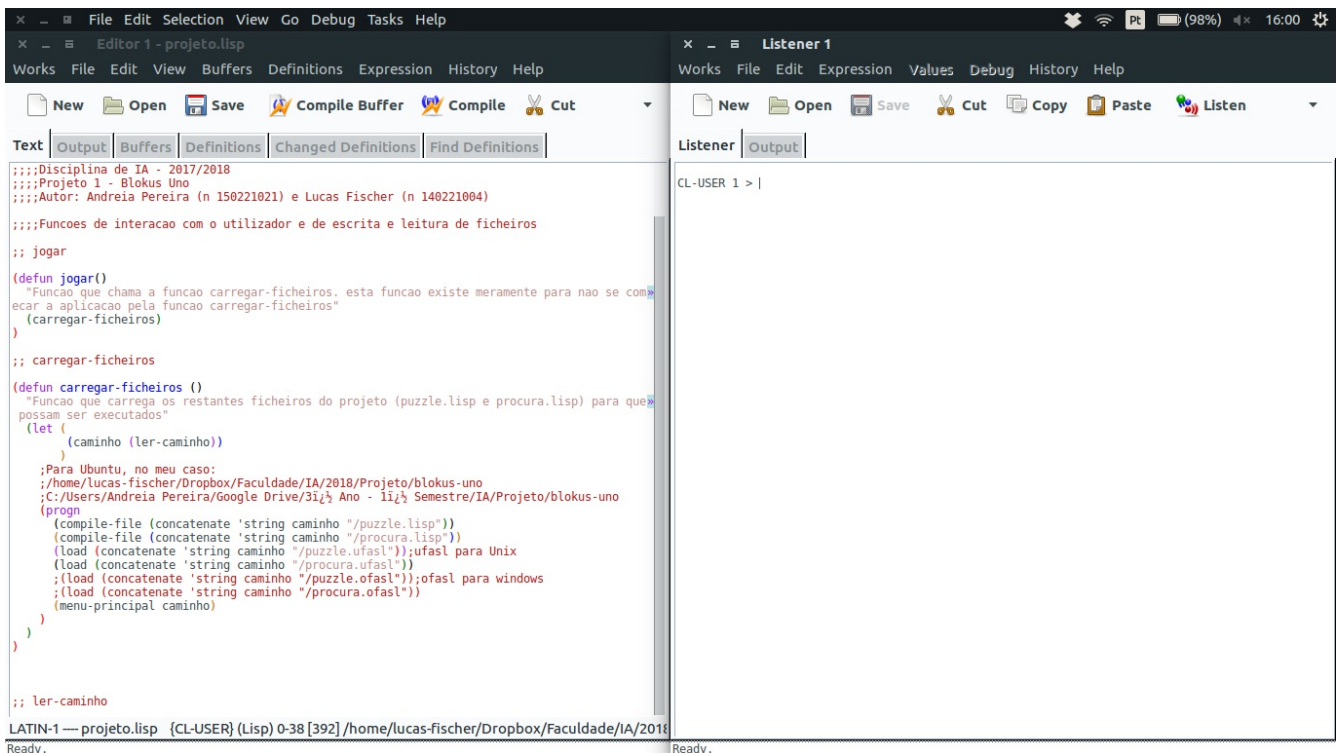
Lucas Fischer nº 140221004

1- Descrição da aplicação


Blokus-Uno é uma aplicação que simula o jogo de blokus com algumas alterações às suas regras. A aplicação tem como objetivo encontrar para, um determinado tabuleiro, qual a solução ótima ou de menor custo para chegar ao fim do jogo.

2- Iniciar a Aplicação

Para poder dar início a aplicação necessita de abrir o ficheiro **projeto.lisp** no IDE **LispWorks**. Uma vez aberto deverá estar à semelhança da seguinte foto:



De modo a que consiga iniciar a aplicação, tem que primeiro compilar

as suas funções, para isso carregue no botão que diga ****"Compile Buffer"**:  [Compile Buffer](./images/compile-buffer.png)

Está agora pronto para iniciar esta fantástica aplicação! Comece por executar a função **(jogar)** no painel chamado **Listener** e de seguida insira o caminho para a diretoria principal da aplicação. (ex: C:/Users/NOME DO UTILIZADOR/Documentos/blokus-uno) e carregue na tecla enter.

```
CL-USER 1 > (jogar)
Introduza o caminho ate ao directorio do projeto
/home/lucas-fischer/Dropbox/Faculdade/IA/2018/Projeto/blokus-uno
```

Irá ver que a aplicação foi bem compilada e de seguida ser-lhe-á apresentado um menu principal onde poderá dar início à aplicação.



Neste magnífico menu principal pode escolher uma de duas opções:

1. **Iniciar** - Esta opção dará início à aplicação, levando-lhe para outros menus onde tem a possibilidade de escolher qual o **tabuleiro inicial** que deseja, qual o **algoritmo** de procura que deseja, qual a **heurística** que deseja e qual a **profundidade**

máxima que deseja (caso tenha selecionado o algoritmo *Depth First*)

2. **Sair** - Como pode deduzir, esta opção leva ao término da aplicação, parando a sua execução.

3- Utilização da Aplicação

Escolhendo a opção de **Iniciar** a aplicação irá levá-lo para outros menus onde irá ser questionado sobre as escolhas que pretende fazer na execução da aplicação.

Estes menus são:

- **Escolha do tabuleiro inicial**

```
-> Escolha um tabuleiro que pretende utilizar como estado do no inicial
-> 1 - Tabuleiro a)
-> 2 - Tabuleiro b)
-> 3 - Tabuleiro c)
-> 4 - Tabuleiro d)
-> 5 - Tabuleiro e)
-> 6 - Tabuleiro f)
6|
```

Onde pode escolher qual o tabuleiro que pretende que seja o tabuleiro inicial

- **Escolha do algoritmo de procura em espaço de estados**

```
-> Escolha o algoritmo de procura em espaco de estados que pretende
-> 1 - Breadth-First (bfs)
-> 2 - Depth-First (dfs)
-> 3 - A*
-> 4 - IDA*
3|
```

Neste menu pode escolher um dos quatro algoritmos de procura em espaço de estados disponíveis na aplicação

- **Escolha da profundidade máxima da árvore de procura (caso tenha escolhido o algoritmo Depth-First)**

```
-> Insira a profundidade maxima que deseja para o algoritmo  
10|
```

Caso tenha escolhido o algoritmo _Depth First_ no menu anterior ser-lhe-á questionado qual a profundidade máxima que pretende que este algoritmo vá. A título de exemplo foi inserido a profundidade máxima de 10 mas o valor inserido é qualquer número positivo que desejar

- **Escolha da heurística a utilizar (caso tenha escolhido o algoritmo A* ou IDA*)**

```
-> Escolha a heuristica que pretende  
-> 1 - Heuristical  
-> 2 - Heuristica2  
2|
```

Neste menu pode escolher qual a heurística que pretende utilizar caso tenha escolhido o algoritmo ****A**** ou ****IDA****. A Heurística é uma função que irá ajudar estes algoritmos de procura a "filtrar" alguns nós que não sejam tão relevantes para o problema.

Após responder a estas questões poderá ver a magia a acontecer, ser-lhe-á apresentado algo com o seguinte aspeto:

```

(1 1 0 0 1 1 0 0 1 1 0 0 2 2)
(1 1 0 0 1 1 0 0 1 1 0 0 2 2)
(0 0 1 1 0 0 1 1 0 0 2 2 0 0)
(0 0 1 1 0 0 1 1 0 0 2 2 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(0 0 1 1 0 0 2 2 0 0 0 0 0 0)
(0 0 1 1 0 0 2 2 0 0 0 0 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(0 0 2 2 0 0 0 0 0 0 0 0 0 0)
(0 0 2 2 0 0 0 0 0 0 0 0 0 0)
(2 2 0 0 0 0 0 0 0 0 0 0 0 0)
(2 2 0 0 0 0 0 0 0 0 0 0 0 0)

-Pecas pequenas: 10
-Pecas medias: 2
-Pecas em cruz: 15

-Profundidade da solucao: 8
-Heuristica do no objetivo: 0
-Custo do no-objetivo: 8

-Nos expandidos: 48
-Nos gerados: 247
-Tempo de Execucao: 1 segundo(s)
-Penetrancia: 0.032388665
-Fator de Ramificacao: 1.8013954

-No pai: (((1 1 0 0 1 1 0 0 0 0 0 0 2 2) (1 1 0 0 1 1 0 0 0 0 0 0 2 2) (0 0 1 1 0 0 1 1 0 0
2 2 0 0) (0 0 1 1 0 0 1 1 0 0 2 2 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (1 1 0 0 2 2 0 0 2 2 0
0 0 0) (0 0 1 1 0 0 2 2 0 0 0 0 0 0) (0 0 1 1 0 0 2 2 0 0 0 0 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0
0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (0 0 2 2 0 0 0 0 0 0 0 0 0 0) (0 0 2 2 0 0 0 0 0 0 0 0 0 0)
(2 2 0 0 0 0 0 0 0 0 0 0 0 0) (2 2 0 0 0 0 0 0 0 0 0 0 0 0)) (10 3 15) 7 3 10 (((1 1 0 0 1 1
0 0 0 0 0 0 2 2) (1 1 0 0 1 1 0 0 0 0 0 0 2 2) (0 0 1 1 0 0 0 0 0 0 2 2 0 0) (0 0 1 1 0 0 0
0 0 2 2 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (0 0 1 1 0 0 2 2 0

```

Muito bem, mas o que quer dizer toda esta informação ? Não se preocupe, ficará a saber tudo já a seguir!

4- Interpretação do Resultado

Agora que já tem o resultado da execução da aplicação falta-lhe apenas saber interpretar os dados fornecidos por este resultado. Vamos passo a passo explicar-lhe cada um.

- **Estado do nó objetivo**

```

(1 1 0 0 1 1 0 0 1 1 0 0 2 2)
(1 1 0 0 1 1 0 0 1 1 0 0 2 2)
(0 0 1 1 0 0 1 1 0 0 2 2 0 0)
(0 0 1 1 0 0 1 1 0 0 2 2 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(0 0 1 1 0 0 2 2 0 0 0 0 0 0)
(0 0 1 1 0 0 2 2 0 0 0 0 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(0 0 2 2 0 0 0 0 0 0 0 0 0 0)
(0 0 2 2 0 0 0 0 0 0 0 0 0 0)
(2 2 0 0 0 0 0 0 0 0 0 0 0 0)
(2 2 0 0 0 0 0 0 0 0 0 0 0 0)

```

A primeira informação que encontramos refere-se ao estado do nó objetivo, isto é, para este problema, o tabuleiro no qual o algoritmo de procura terminou, que corresponde a um tabuleiro onde o jogador não consegue por mais nenhuma das suas peças (representadas pelo número 1).

- **Peças restantes do nó objetivo**

```

-Pecas pequenas: 10
-Pecas medias: 2
-Pecas em cruz: 15

```

Estas três linhas indicam-lhe as peças que o jogador ainda tinha quando chegou ao nó objetivo. O jogo inicia-se com **10 peças pequenas, 10 peças médias e 15 peças em cruz** por isso deixamos-lhe fazer as contas para determinar quantas peças foram usadas.

- **Informação sobre o custo, heurística e profundidade**


```
-Profundidade da solucao: 8  
-Heuristica do no objetivo: 0  
-Custo do no-objetivo: 8
```

As seguintes três linhas do resultado dão-lhe informação sobre a profundidade em que se encontrava o nó objetivo, o valor heurístico que esse nó objetivo tinha (o ideal é que o valor seja **0** nos nós objetivos), e o custo necessário para que se consiga chegar desde o nó inicial até este nó objetivo.

- **Informação sobre a eficiência do algoritmo**

```
-Nos expandidos: 48  
-Nos gerados: 247  
-Tempo de Execucao: 1 segundo(s)  
-Penetrancia: 0.032388665  
-Fator de Ramificacao: 1.8013954
```

Esta porção do resultado dá-lhe informação sobre a eficiência que o algoritmo obteve, calculada através de métricas como **Penetrância** e **Fator de ramificação médio**, estas métricas envolvem fórmulas matemáticas complexas por isso poupamos-lhe o trabalho de ter que fazer as contas, não tem que agradecer.

Nós expandidos - Este valor indica quantos nós o algoritmo expandiu (gerou os seus sucessores) até que encontrasse um nó que fosse um nó objetivo (quanto menor este valor melhor)

Nós gerados - Os nós gerados são todos os nós que resultam da expansão de um nó, logo, quantos menos nós gerados menor será a memória utilizada pelo algoritmo.

Tempo de Execução - O tempo de execução, como é dito, é medido em segundos e como pode deduzir, quanto menor for o tempo de

execução melhor é a eficiência do algoritmo.

Penetrância - A penetrância é um valor entre **0** e **1** que permite-lhe ter uma noção sobre a relação entre quantos nós foram gerados e quantos nós fazem realmente parte do caminho da solução do problema. O ideal será ter uma penetrância de valor 1 o que significa que todos os nós que foram gerados fazem parte da solução do problema.

Fator de ramificação médio - Esta métrica consiste num valor entre **1** e **mais infinito (+∞)** e representa o número de sucessores que, em média, cada nó possui.

- **Informação sobre o nó pai do nó objetivo**

```
-No pai: (((1 1 0 0 1 1 0 0 0 0 0 0 2 2) (1 1 0 0 1 1 0 0 0 0 0 0 2 2) (0 0 1 1 0 0 1 1 0 0
2 2 0 0) (0 0 1 1 0 0 1 1 0 0 2 2 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (1 1 0 0 2 2 0 0 2 2 0
0 0 0) (0 0 1 1 0 0 2 2 0 0 0 0 0 0) (0 0 1 1 0 0 2 2 0 0 0 0 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0
0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (0 0 2 2 0 0 0 0 0 0 0 0 0 0) (0 0 2 2 0 0 0 0 0 0 0 0 0 0)
(2 2 0 0 0 0 0 0 0 0 0 0 0 0) (2 2 0 0 0 0 0 0 0 0 0 0 0 0)) (10 3 15) 7 3 10 (((1 1 0 0 1 1
0 0 0 0 0 0 2 2) (1 1 0 0 1 1 0 0 0 0 0 0 2 2) (0 0 1 1 0 0 0 0 0 0 2 2 0 0) (0 0 1 1 0 0 0
0 0 0 2 2 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (0 0 1 1 0 0 2 2 0
0 0 0 0 0) (0 0 1 1 0 0 2 2 0 0 0 0 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (1 1 0 0 2 2 0 0 2 2
0 0 0 0) (0 0 2 2 0 0 0 0 0 0 0 0 0 0) (0 0 2 2 0 0 0 0 0 0 0 0 0 0) (2 2 0 0 0 0 0 0 0 0 0 0
0 0) (2 2 0 0 0 0 0 0 0 0 0 0 0 0)) (10 4 15) 6 4 10 (((1 1 0 0 0 0 0 0 0 0 0 0 2 2) (1 1 0
0 0 0 0 0 0 0 0 0 0 2 2) (0 0 1 1 0 0 0 0 0 0 2 2 0 0) (1 1 0 0 2
2 0 0 2 2 0 0 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (0 0 1 1 0 0 2 2 0 0 0 0 0 0) (0 0 1 1 0 0
2 2 0 0 0 0 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (0 0 2 2 0 0 0 0
0 0 0 0 0 0) (0 0 2 2 0 0 0 0 0 0 0 0 0 0) (2 2 0 0 0 0 0 0 0 0 0 0 0 0) (2 2 0 0 0 0 0 0 0 0
0 0 0 0 0 0)) (10 5 15) 5 3 8 (((1 1 0 0 0 0 0 0 0 0 0 0 2 2) (1 1 0 0 0 0 0 0 0 0 0 0 2 2) (0
0 1 1 0 0 0 0 0 0 2 2 0 0) (0 0 1 1 0 0 0 0 0 0 2 2 0 0) (1 1 0 0 2 2 0 0 2 2 0 0 0 0) (1 1 0
0 2 2 0 0 2 2 0 0 0 0) (0 0 1 1 0 0 2 2 0 0 0 0 0 0) (0 0 1 1 0 0 2 2 0 0 0 0 0 0) (0 0 0 0
```

A última informação que o resultado lhe apresenta é uma lista (que pode ser enorme) que corresponde ao nó pai do nó objetivo. Neste nó pai pode por sua vez ver o nó pai desse nó e assim sucessivamente até chegar ao nó raiz por isso é uma informação útil que lhe permite obter o caminho de solução que o algoritmo percorreu até encontrar o nó objetivo.

5- Ficheiro estatisticas.dat

Para que não tenha que estar sempre a repetir todos estes passos e estar constantemente a executar a aplicação, esta mesma escreve os resultados tal como apresentados anteriormente, num ficheiro situado na diretoria principal do projeto denominado ***estatisticas.dat***.

```
(1 1 0 0 1 1 0 0 1 1 0 0 2 2)
(1 1 0 0 1 1 0 0 1 1 0 0 2 2)
(0 0 1 1 0 0 1 1 0 0 2 2 0 0)
(0 0 1 1 0 0 1 1 0 0 2 2 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(0 0 1 1 0 0 2 2 0 0 0 0 0 0)
(0 0 1 1 0 0 2 2 0 0 0 0 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(1 1 0 0 2 2 0 0 2 2 0 0 0 0)
(0 0 2 2 0 0 0 0 0 0 0 0 0 0)
(0 0 2 2 0 0 0 0 0 0 0 0 0 0)
(2 2 0 0 0 0 0 0 0 0 0 0 0 0)
(2 2 0 0 0 0 0 0 0 0 0 0 0 0)
```

```
-Pecas pequenas: 10
-Pecas medias: 2
-Pecas em cruz: 15
```

```
-Profundidade da solucao: 8
-Heuristica do no objetivo: 0
-Custo do no-objetivo: 8
```

```
-Nos expandidos: 48
-Nos gerados: 247
-Tempo de Execucao: 1 segundo(s)
-Penetrancia: 0.032388665
-Fator de Ramificacao: 1.8013954
```

6- Conclusão

Agora que já concluiu a leitura deste manual já possui todas as capacidades para executar a aplicação e tirar toda a informação que precisa sobre ela, espero que goste!

Happy coding from **Andreia Pereira & Lucas Fischer!**