

BUSCA DE SOLUÇÕES UTILIZANDO OS ALGORITMOS DE BUSCA

(BUSCA EM LARGURA E BUSCA EM PROFUNDIDADE)

Matheus Cândido Teixeira¹

1-INSTITUTO FEDERAL DE MATO GROSSO

Para executar esse programa é necessário executar os seguintes comandos:

```
pip install Pillow
pip install appJar
```

Para testar o programa há duas possibilidades, utilizando o prompt de comando e utilizando GUI.

1. UTILIZANDO PROMPT DE COMANDO

Edite o estado inicial no arquivo **modelo.py**:

```
#--Estado inicial
E0 = [ [1,2,3] , [4,5,'x'] , [7,8,6] ]
```

E o objetivo é obter o seguinte:

```
#--Estado final
Eobj = [ [1,2,3] , [4,5,6] , [7,8,'x'] ]
```

1.1. MÉTODO DE BUSCA EM LARGURA

Para este teste utilize os seguintes comandos a partir do prompt de comando:

```
python game.py largura
```

A saída será:

```
Solucao encontrada [largura = 3, profundidade = 2]

Solucao:

[[[1, 2, 3], [4, 5, 'x'], [7, 8, 6]], [[1, 2, 3], [4, 5, 6], [7, 8, 'x']]]

-----
```

O programa também gera uma saída gráfica contida no arquivo **solucao.png**:

1	2	3	1	2	3
4	5	x	4	5	6
7	8	6	7	8	x

1.2. MÉTODO DE BUSCA EM PROFUNDIDADE

Para este teste utilize os seguintes comandos a partir do prompt de comando:

```
python game.py profundidade
```

A saída será:

```
Solucao encontrada [largura = 3, profundidade = 2]

Solucao:

[[[1, 2, 3], [4, 5, 'x'], [7, 8, 6]], [[1, 2, 3], [4, 5, 6], [7, 8, 'x']]]
-----
```

O programa também gera uma saída gráfica contida no arquivo **solucao.png**:

1	2	3
4	5	x
7	8	6

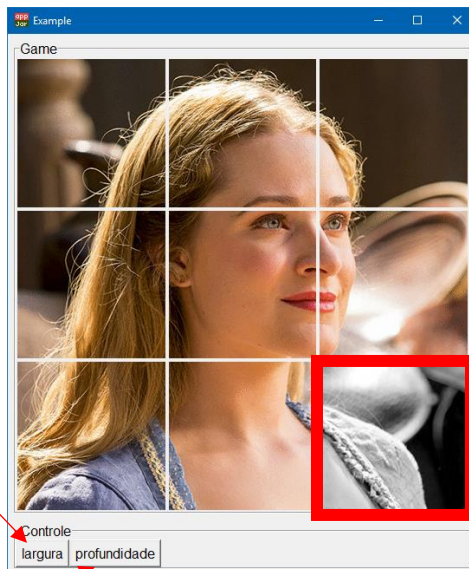
1	2	3
4	5	6
7	8	x

2. UTILIZANDO MÉTODO GRÁFICO

Para utilizando o modo gráfico utilize o seguinte comando:

```
python game.py GUI
```

O resultado será:



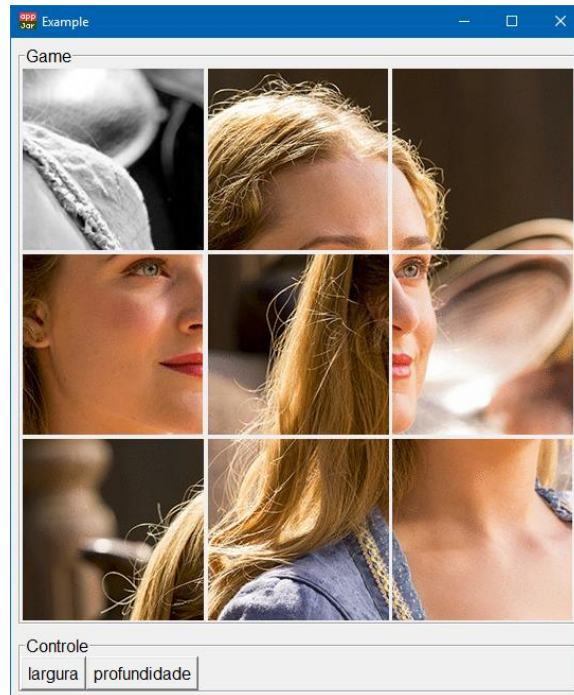
Busca resolver utilizando o método da largura

A peça que pode ser movida

Busca resolver utilizando o método da profundidade

Para mover uma peça clique nela e se o movimento for válido ele será realizado.

Exemplo (Utilizando o método da largura):



Após alguns segundos a solução será gerada no arquivo **solucao.png**:

x	2	3	2	x	3	2	4	3	2	4	3	2	4	3	2	4	3	2	x	3	x	2	3	1	2	3	1	2	3	1	2	3						
5	4	6	5	4	6	5	x	6	x	5	6	1	5	6	1	5	6	1	x	6	1	4	6	1	4	6	x	4	6	4	x	6	4	5	6	4	5	6
1	7	8	1	7	8	1	7	8	1	7	8	x	7	8	7	x	8	7	5	8	7	5	8	7	5	8	7	5	8	7	5	8	7	x	8	7	8	x

Solucao encontrada [largura = 4, profundidade = 13]

Para a mesma ordenação anterior, utilizando o método da profundidade não alcança um solução mesmo após 10000 iterações.