

# **Trabalho Prático: Quebra-cabeça de 8 peças**

**Atos Brito Omena<sup>1</sup>, Grécia Cristina Garcia Rivera<sup>2</sup>, Joao Victor Nogueira de Souza<sup>3</sup>,**

<sup>1</sup>Instituto de Computação– Universidade Federal do Amazonas (UFAM)  
Av. General Rodrigo Octavio Jordão Ramos, 1200 - Coroado I, Manaus - AM,  
69067-005

atos.omena@icomp.ufam.edu.br, grecia.rivera@icomp.ufam.edu.br,  
joao.souza@icomp.ufam.edu.br

## **1. Introdução**

O problema escolhido para estudo foi o 8-puzzle (ou Quebra-cabeça de 8 peças) que consiste num tabuleiro com peças numeradas e acomodadas de forma aleatória com uma peça faltando. Diante disso, o objetivo consiste na ordenação das peças usando o espaço vazio para deslizá-las.

No quebra-cabeça em estudo, há um tabuleiro formado por uma matriz de ordem 3 com a numeração de 1 a 8 (com o espaço vazio representando o número zero). Além disso, o usuário poderá mover em até quatro direções a célula vazia para tentar resolver, no menor número de movimentos possível, o quebra-cabeça, ou seja, ordená-lo.

## **2. Metodologia**

### **2.1. Procedimento de busca implementado :**

O algoritmo implementado foi baseado no algoritmo de Dijkstra que tem por finalidade calcular o caminho de custo mínimo entre dois estados. Nesse sentido, o algoritmo Dijkstra parte de uma estimativa (heurística) para o custo mínimo e gradativamente a suposição é ajustada.

### **2.2. Decisões de projeto para elaboração de interface :**

A interface foi elaborada utilizando a biblioteca Tkinter. Através dela, foram implementados 6 botões.

“Gerar aleatórios” : O primeiro no canto superior esquerdo, gera números aleatórios no tabuleiro

”Resolver”: O segundo botão no canto superior esquerdo, tem o objetivo de solucionar o jogo mostrando cada estado.

← ↑ → ↓ : 4 botões dispostos de forma adjacente ao tabuleiro, movem a célula vazia no respectivo sentido da seta do botão.

Outra decisão aderida ao planejamento de interface é a possibilidade de controle da célula pelas teclas A, W, S, D, setas do teclado e ao clicar nos botões da interface com o cursor.

### **2.3. As instruções de instalação e uso do aplicativo desenvolvido.**

Comandos de instalações que podem ser necessárias no terminal caso o usuário não tenha pré-instalado na máquina:

```
pip install keyboard
```

Depois de instalado, o usuário precisa fazer clonagem do repositório do gitHub

([https://github.com/AnemoSota/TrabalhoAED2\\_FilaDePrioridades](https://github.com/AnemoSota/TrabalhoAED2_FilaDePrioridades))

e após abrir o terminal do repositório, digitar o comando

```
python main.py
```

Após isso, o programa abrirá uma janela com a interface do jogo e o usuário poderá começar a usá-lo.

Ao clicar em “Gerar aleatórios” o tabuleiro será reformulado de maneira que os números estarão dispostos de forma aleatória.

Ao clicar em ”Resolver”, o tabuleiro será solucionado dinamicamente passo a passo a matriz no tabuleiro e faz aparecer na parte inferior do display uma mensagem de processamento durante o tempo de resolução do programa . Além dele, há 4 botões com setas indicando seu respectivo sentido. Visto isso, cada um desses botões tem a função de mover a célula vazia na respectiva direção da seta do botão caso o usuário queira resolver manualmente o 8 puzzle ou estabelecer uma ordem específica para começar.

Ademais, caso o usuário queira realizar um movimento invalido (mover a célula vazia para um sentido fora dos limites do tabuleiro), na parte inferior do display aparecerá o aviso de “Jogada inválida!”.