



UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO

DANIELE ALVES
FERNANDA ALICE
JO PEREIRA
LUIS GUSTAVO
SARAH JÚLIA

QUEBRA-CABEÇA DE 8 PEÇAS

MANAUS

2023

DESCRIÇÃO DO PROBLEMA ESCOLHIDO

O 8 puzzle é um jogo de quebra-cabeça com blocos deslizantes jogado em um tabuleiro de 3x3 quadrado, que contém oito peças numeradas aleatoriamente e uma peça vazia. O objetivo do jogo é mover as peças para que fiquem em ordem numérica, com a peça vazia sendo movida para permitir que as outras peças sejam movidas.

O jogo começa com as peças misturadas aleatoriamente, com a peça vazia em uma posição qualquer. O jogador pode mover uma peça para uma posição adjacente vazia, fazendo com que a peça movida ocupe a posição vazia. O objetivo é encontrar uma sequência de movimentos que permitam que as peças sejam colocadas em ordem numérica, começando com o 1 no canto superior esquerdo e terminando com a peça vazia no canto inferior direito.

O jogo é um desafio de lógica que pode ser resolvido através de uma variedade de métodos, incluindo a aplicação de algoritmos de busca, como a busca em largura ou a busca em profundidade. O jogo é frequentemente usado como um problema de teste para algoritmos de inteligência artificial e programação. O jogo também pode ser jogado como um passatempo, com o objetivo de resolvê-lo no menor número de movimentos possível.

EXPLICAÇÃO DO PROCEDIMENTO DE BUSCA IMPLEMENTADO

Para realizar a busca do melhor caminho para a solução do problema foi utilizado um algoritmo de busca informada, conhecido como A*, que basicamente consiste na criação de duas listas, uma para guardar as possibilidades de caminho e outra para guardar os estados já passados.

A função começa com um valor representando o estado final, no caso, o valor “12345678”, com uma fila de prioridade vazia e um conjunto de valores separados para armazenar os estados que já passaram pela função.

A busca em si é realizada dentro de um laço de repetição que ocorre enquanto a fila de prioridade não encontra-se vazia, a cada iteração do laço é retirado o primeiro valor da fila de prioridade e é atribuído tal valor ao estado atual e então é feita uma verificação se o estado atual é equivalente ao estado final, se for, é retornado então uma lista com o caminho do estado inicial até o final e o número de passos dados. Se o estado não for o final, então é verificada todas as possíveis transições a serem realizadas e estas são adicionadas na fila de prioridade, para então retornar para o começo do laço de repetição.

Note-se que um fator importante deste algoritmo é a utilização do conjunto para guardar os estados passados, que faz com que não haja verificação em estados que já foram passados e

permite que o algoritmo não entre num eterno loop e acabe encontrando o melhor caminho a ser retornado.

Para a utilização desse algoritmo foi necessária a criação de uma classe Estado, representando o estado atual, com os atributos ordem, número, antecessor, h, g e f. O atributo ordem e número simbolizam a ordem numérica do estado, são atributos únicos de cada objeto dessa classe, a utilização do atributo numero vem a ser necessário somente para que pudesse ser inserido no conjunto de estados passados. Já o atributo antecessor vem a ser utilizado para retornar o estado anterior que deu origem ao objeto atual. Por fim, os atributos h, g e f representam a heurística utilizada, o número de passos e o custo de cada estado, respectivamente.

Como indicado nas especificações, a heurística foi calculada como sendo a quantidade de peças fora do lugar adequado, desconsiderando o espaço vazio e o custo seria o valor da heurística somado à quantidade de passos.

DECISÕES DE PROJETO PARA ELABORAÇÃO DA INTERFACE

Por se tratar de uma interface própria para jogos em python, como já diz o nome, decidimos por usar o pygame. Além da alta quantidade de materiais já disponíveis sobre o uso da interface pela internet já que o pygame possui uma comunidade grande e ativa, o que facilitou a aprendizagem e a aplicação da interface ao código. Iríamos usar a biblioteca Curses, uma biblioteca do próprio Python para a interface do jogo, porém a falta de material em português e a dificuldade da equipe em aplicá-lo ao código do trabalho - visto que seriam necessárias várias instalações para fazer a biblioteca funcionar adequadamente - fizeram a equipe decidir pelo uso do Pygame, o qual é de fácil instalação e portátil.

INSTRUÇÕES DE INSTALAÇÃO E USO DO APLICATIVO DESENVOLVIDO

Para preparar o ambiente para a execução do jogo, é preciso, primeiramente, fazer o download e a instalação da versão mais recente do Python, acessando o seguinte link: <https://www.python.org/downloads/> e baixando o arquivo indicado. Após a devida instalação, basta ir ao ambiente virtual e digitar: `python -m pip install -U pygame`.

E para poder jogar o quebra-cabeça em si, é necessário fazer o download da pasta contendo todos os arquivos do jogo e executar o arquivo Main.py pelo próprio terminal.