



Universidade do Oeste Paulista
Sistemas de Informação

João Vitor Dias da Silva – 262113651

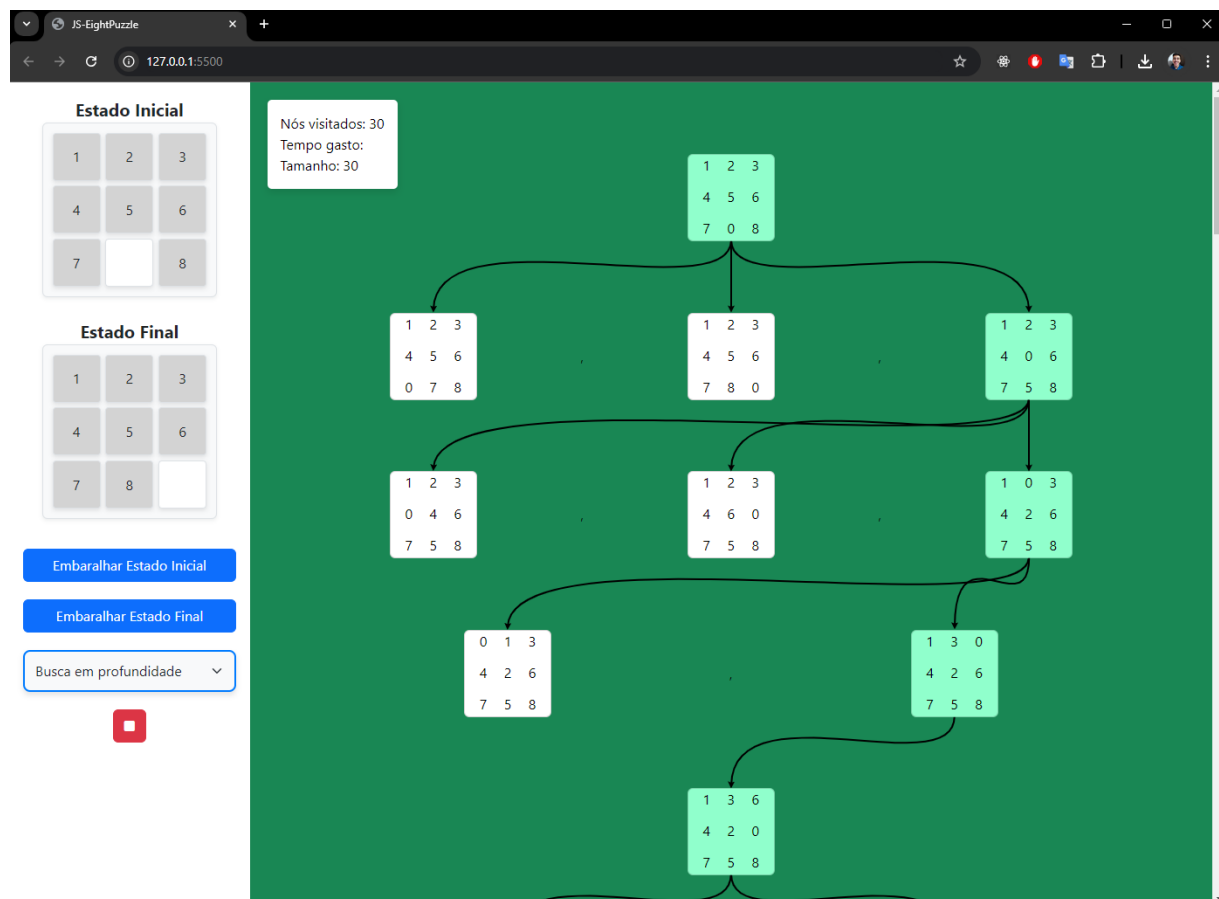
Trabalho 1º bimestre - 8puzzle

Presidente Prudente – SP

O que é o 8-puzzle

O 8-puzzle é um clássico quebra-cabeça, composto por um tabuleiro 3x3 com 8 peças numeradas de 1 a 8, além de um espaço vazio (representado pelo número 0). O jogo permite que as peças sejam movidas uma de cada vez para esse espaço vazio, com o objetivo de reorganizá-las até atingir uma configuração específica. Normalmente, o jogo começa com as peças em uma disposição desordenada, e o desafio é movê-las para chegar a uma sequência ordenada. Esse quebra-cabeça é amplamente utilizado como exemplo de problema de busca, sendo uma ferramenta comum para testar algoritmos de inteligência artificial, especialmente aqueles focados em busca heurística.

Apresentação da interface gráfica (tela)



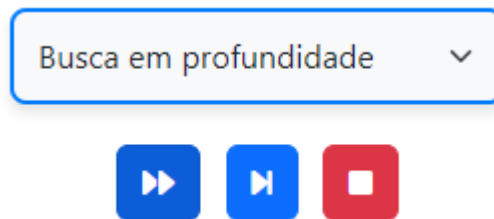
Painel Esquerdo

No painel à esquerda se encontram as opções de configuração da situação problema, sendo elas: Definição de estados (Inicial e Final) e seus respectivos botões para randomizar os estados, e a seleção do algoritmo de busca (seleção de heurística). Também é possível realizar o embaralhamento dos estados arrastando o bloco branco para cima, baixo, esquerda ou direita (1 casa por vez)

Seleção de algoritmo de busca

No campo de escolha de algoritmo, estão disponíveis as opções: Busca em Profundidade e A* (A-Estrela). Cada opção exige um comportamento de busca de solução diferente, interferindo diretamente nos valores dos resultados de quantidade de passos (nós) visitados para obter a solução, o tempo gasto, tamanho do caminho da solução encontrada.

Com todas as opções de configuração necessárias selecionadas, o botão 'resolver' irá dar início ao processamento da solução, e avançar passo a passo, ou ir diretamente ao resultado final



Painel Direito

No painel à direita se encontra a visualização dos valores gerados pela busca da situação problema, sendo eles: o relatório contendo a quantidade de passos (nós) visitados para obter a solução, o tempo gasto e tamanho do caminho da solução encontrada, a exibição do puzzle de estado (nó atual) e a exibição/construção da árvore de nós.

Lógica de processamento das buscas

Busca em profundidade

O algoritmo de busca em profundidade (DFS - Depth-First Search) é uma técnica de busca não informada frequentemente utilizada para explorar grafos. O DFS dá preferência em explorar um caminho no grafo até o final antes de retornar e explorar outros ramos. É importante lembrar que, ao contrário da busca em largura, o DFS não garante encontrar o caminho mais curto. Dependendo da estrutura do grafo, ele pode acabar preso em ciclos infinitos ou explorar profundamente antes de achar a solução, o que pode torná-lo menos eficiente em alguns casos. No entanto, o DFS é bastante aplicado em situações como a resolução de labirintos ou na travessia de árvores em algoritmos como a busca em profundidade limitada e a busca em profundidade iterativa.

4.3. Busca A* (A-Estrela)

O algoritmo A* utiliza uma função de avaliação baseada no cálculo da distância de Manhattan para estimar o custo do melhor caminho entre o nó inicial e o objetivo. A distância de Manhattan mede a soma das diferenças absolutas entre as coordenadas dos nós, o que é especialmente útil em problemas de movimentação em grades, como o 8-puzzle. Essa função heurística é admissível e consistente, garantindo que o

algoritmo A* seja ótimo e encontre a solução de menor custo, desde que o espaço de estados permita.

A expansão dos nós acontece com base no menor custo total, que é a soma do custo do caminho já percorrido até aquele nó e a estimativa heurística do custo restante até o objetivo. O algoritmo é considerado ótimo quando a heurística utilizada é admissível (nunca superestima o custo real para alcançar o objetivo) e consistente (a estimativa de custo de um nó é sempre menor ou igual ao custo de chegar a um nó sucessor somado à estimativa do sucessor).