INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DO ESPÍRITO SANTO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Davidson Santos, Thiago Borges

Trabalho Prático 1 - Labirinto

Serra/ES

2024

Objetivo

O objetivo deste trabalho prático é implementar um programa em Java que resolva labirintos representados por matrizes de 0 e 1. No contexto deste labirinto, os números 0 representam paredes, enquanto os números 1 representam caminhos. Cada labirinto possui uma única entrada e uma única saída, garantindo que sempre haja um caminho que conecta esses dois pontos. No entanto, o labirinto pode conter becos sem saída.

Descrição da Atividade

1. **Entrada:**
   O programa deve ser capaz de ler instâncias de labirintos a partir de arquivos no formato CSV. Cada arquivo deve conter uma matriz, onde cada linha representa uma linha do labirinto e cada elemento é separado por vírgulas.
2. **Processamento:**
   O algoritmo deve encontrar o caminho da entrada até a saída do labirinto. Utilize técnicas de busca (como busca em profundidade ou em largura) para explorar os caminhos disponíveis, considerando que o labirinto pode conter becos sem saída.
3. **Saída:**
   Após encontrar o caminho, o programa deve ilustrar visualmente o movimento do elemento que percorre o labirinto. A visualização deve destacar o caminho encontrado, permitindo a fácil identificação da entrada, saída e das paredes.
4. **Requisitos Adicionais:**
   - O código deve ser bem estruturado e comentado, seguindo as boas práticas de programação.
   - O programa deve ser capaz de lidar com labirintos de diferentes tamanhos e formas, garantindo a robustez da solução.

Solução

Funcionalidades

1. **Carregamento de Labirintos:** Labirintos são carregados a partir de arquivos CSV, onde os números "1" representam caminhos possíveis e "0" representam paredes.
2. **Algoritmo de Backtracking:** Utiliza a técnica de backtracking para encontrar a saída do labirinto a partir de um ponto de entrada.
3. **Visualização Gráfica:** O labirinto e o percurso de solução são visualizados utilizando Swing, com animação do movimento através de um temporizador (Timer).
4. **Movimento Animado**: Cada passo da solução é desenhado em intervalos de 500ms, mostrando o avanço pelo labirinto até a saída.

Conceitos Utilizados

1. **Backtracking:** Algoritmo de busca utilizado para encontrar soluções em problemas que requerem exploração de várias alternativas, voltando atrás sempre que necessário (retrocedendo quando uma rota se mostra inviável).
2. **TAD Stack (Pilha):** Utilizada para manter o histórico de posições durante a exploração do labirinto.

Tecnologias Utilizadas

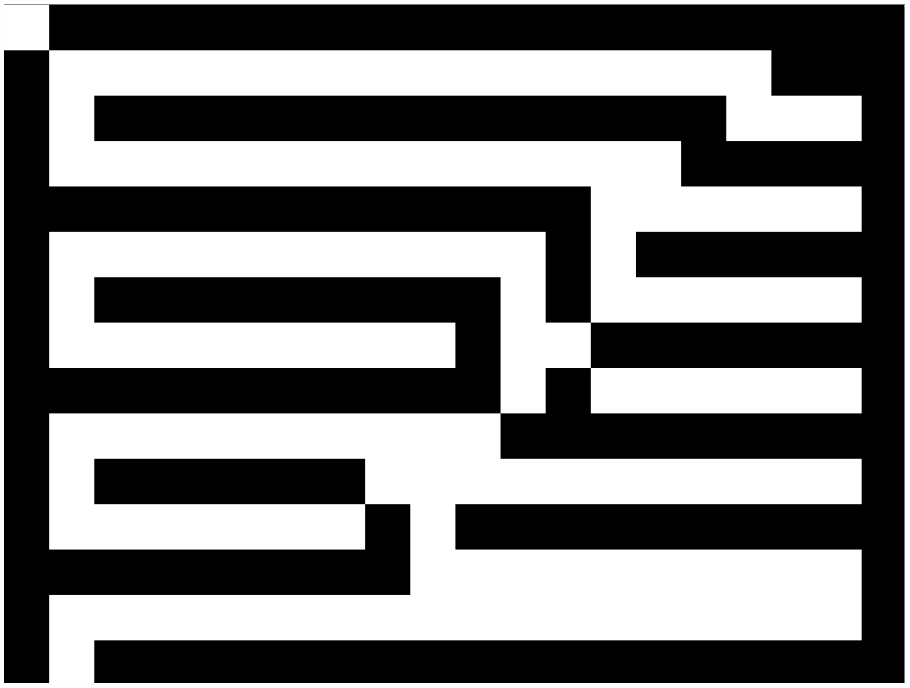
1. **Java:** Linguagem de programação utilizada para implementar a lógica do labirinto e a interface gráfica.
2. **Swing:** Biblioteca Java para construção da interface gráfica.

Estrutura do Projeto

1. **Maze**: Classe responsável por carregar, representar e processar o labirinto. Contém o algoritmo de backtracking para resolver o labirinto.
2. **Node**: Classe que representa os nós do labirinto (coordenadas x e y), utilizados durante o percurso.
3. **MazePanel**: Classe que desenha o labirinto e anima o percurso da solução.
4. **Main**: Classe principal que inicializa a interface gráfica e exibe o labirinto.

Execução

Entrada: Labirinto 15x15



Saída: Caminho Percorrido, Caminhos Falhos