

CAMPO MINADO – TP2

Este projeto é um novo trabalho prático da disciplina Algoritmos e Estruturas de Dados, ministrada pelo professor Juan Gabriel Colonna – ICOMP/UFAM. O objetivo era empregar os conceitos de TAD, recursão e arquivos de texto no primeiro trabalho prático de Campo Minado.

O trabalho foi implementado e tem como base o trabalho TP1 do Campo Minado.

Como executar

Basta compilar o arquivo com “gcc campo_minado_davicarlos.c matriz.c log.c” e executar “./a.out campo_minado_davicarlos.c”

Caso queira usar entrada por arquivo deve-se, além de compilar “gcc campo_minado_davicarlos.c matriz.c log.c”, o arquivo de texto deve estar no mesmo diretório e a primeira linha deve ser F/f, M/m ou D/d para definir a dificuldade, as próximas linhas devem ser as coordenadas separadas por vírgula. Por exemplo:

[F]

[1,2]

[1,3]

[...]

E então o comando deve ser: “./a.out. campo_minado_davicarlos.c < nomearquivo.txt”

Em relação à primeira versão do projeto (TP1), o TP2 apresenta as seguintes melhorias:

1. Organização do Código com TAD:

- As funcionalidades de manipulação da matriz do tabuleiro foram encapsuladas nos arquivos matriz.c e matriz.h.
- O gerenciamento do sistema de log foi implementado nos arquivos log.c e log.h, seguindo os princípios de TAD.

2. Implementação do Sistema de Log:

- Agora, o histórico completo do jogo é registrado em um arquivo log.txt.
- O log armazena:
 - Data e hora do início da partida.
 - Cada estado do tabuleiro impresso na tela.
 - As coordenadas jogadas pelo usuário.

3. Algoritmo Recursivo Flood Fill:

- Foi implementado um algoritmo recursivo flood fill para revelar automaticamente as células adjacentes com valor 0, quando o jogador acerta uma célula vazia.

4. Ajuste na Quantidade de Minas:

- A quantidade de minas em cada dificuldade foi redefinida para corresponder a 15% do total de células do tabuleiro:
 - Fácil (10x10): 15 minas.
 - Médio (20x20): 60 minas.

- Difícil (30x30): 135 minas.

5. **Documentação Completa e Exemplos de Uso:**

- Cada arquivo .h foi documentado detalhadamente, incluindo exemplos de uso para as funções.
- Além da entrada manual, o jogo agora aceita um arquivo de entrada (txt) para automatizar a escolha da dificuldade e das jogadas. Isso permite testes mais rápidos e consistentes.

A implementação de conceitos avançados, como TAD e recursão, tornou o projeto mais modular, eficiente e de fácil manutenção.