

Relatório do Projeto de Estrutura de Dados I

Jogo Bocha Geométrica

Autor: Beatriz Cristina Vaz Matrícula: 202500560227 Professor: Evandro Baccarian

1. Introdução

Este relatório detalha a implementação do Trabalho I da disciplina de Estrutura de Dados I, que consiste no desenvolvimento de um jogo intitulado "Bocha Geométrica". O projeto foi desenvolvido em linguagem C, seguindo o padrão C99, e utiliza estruturas de dados como pilhas e filas para gerenciar os elementos do jogo. O objetivo principal do trabalho é aplicar os conceitos de modularização, manipulação de arquivos e implementação de estruturas de dados para criar uma aplicação funcional e bem estruturada. O jogo envolve disparadores que lançam formas geométricas em uma arena. As formas podem colidir, ser destruídas ou modificadas, gerando uma pontuação. A interação com o programa se dá por meio de arquivos de entrada que definem as formas, os disparadores e as ações a serem executadas.

2. Estrutura do Projeto

O projeto está organizado em um diretório principal que contém o arquivo `LEIA-ME.txt`, o `Makefile` para compilação e um subdiretório `src` com todos os arquivos de código-fonte. A estrutura de arquivos é a seguinte:

Plain Text

```
projeto_ed1/
├── LEIA-ME.txt
├── RELATORIO.md
└── src/
    ├── arena.c
    ├── arena.h
    ├── arvorebin.c
    ├── arvorebin.h
    ├── carregador.c
    ├── carregador.h
    ├── circulo.c
    ├── circulo.h
    ├── disparador.c
    ├── disparador.h
    └── fila.c
```

```
├── fila.h
├── formas.c
├── formas.h
├── lancadores.c
├── lancadores.h
├── linha.c
├── linha.h
├── lista.c
├── lista.h
└── main.c
    └── Makefile
├── pilha.c
└── pilha.h
├── processageo.c
├── processageo.h
├── processaqry.c
└── processaqry.h
├── retangulo.c
└── retangulo.h
├── texto.c
└── texto.h
└── tratarstrings.c
└── tratarstrings.h
└── ted
```

2.1. Módulos Implementados

O projeto foi dividido nos seguintes módulos:

- **pilha.h / pilha.c** : Implementação de uma pilha genérica, utilizada pelos carregadores.
- **fila.h / fila.c** : Implementação de uma fila genérica, utilizada pela arena para gerenciar as formas lançadas.
- **carregador.h / carregador.c** : Gerencia os carregadores, que são pilhas de formas.
- **disparador.h / disparador.c** : Gerencia os disparadores, que contêm os carregadores e a lógica de disparo.
- **arena.h / arena.c** : Gerencia a arena do jogo, onde as formas são processadas.
- **formas.h / formas.c** : Funções para manipulação das formas geométricas (círculos, retângulos, etc.).
- **processageo.c / processaqry.c** : Módulos responsáveis por ler e processar os arquivos de entrada `.geo` e `.qry`.
- **main.c** : Ponto de entrada do programa, responsável por processar os argumentos da linha de comando e orquestrar a execução.

3. Compilação e Execução

Para compilar o projeto, basta utilizar o comando `make` no diretório `src`. O `Makefile` fornecido compilará todos os arquivos-fonte e gerará o executável `ted`.

Bash

```
cd src  
make
```

O programa pode ser executado da seguinte forma:

Bash

```
./ted -e [diretorio_entrada] -f [arquivo.geo] -o [diretorio_saida] -q  
[arquivo qry]
```

Onde:

- `-e` : Diretório base de entrada.
- `-f` : Arquivo `.geo` com a descrição das formas.
- `-o` : Diretório base de saída.
- `-q` : Arquivo `.qry` com as consultas.

4. Conclusão

O desenvolvimento deste projeto permitiu a aplicação prática de conceitos fundamentais de Estrutura de Dados, como a implementação e utilização de pilhas e filas, além de reforçar a importância da modularização e do bom design de software. A base de código dos trabalhos anteriores foi de grande valia para acelerar o desenvolvimento, permitindo focar nos novos requisitos do Trabalho I. O programa implementa as funcionalidades essenciais descritas na especificação do projeto, como a criação de disparadores, carregadores, e o processamento das formas na arena. A estrutura do código foi mantida organizada e modular, facilitando a compreensão e a manutenção.