Relatório do Efolio B: Laboratório de Programação (21178)

Gabriel Alexandre Pereira, nº-2301429

Introdução

No Efolio B, foi-nos pedido a construção modular de um programa em C, que se baseava na criação e manipulação de retângulos, uma extensão da atividade formativa 3. Nesta atividade 3, implementamos funcionalidades para criação de retangulos, movimentação e aplicação de "gravidade"(que faz com que os retângulos caiam até encontrarem o chão ou outros retângulos) nos mesmos. No Efolio B, foi-nos pedido para adicionarmos a opção de fundir os retângulos, com alertas ao utilizador se tal é possível, e também imprimir “o” no centro dos retângulos. A estrutura modular do programa foi projetada para facilitar a manutenção e permitir a reutilização ou melhorias futuras.

Estrutura Modular do Programa

O programa é composto por quatro módulos principais:

1. Função Principal (main.c e main.h)
2. Gestão de Retângulos (retangulos.c e retangulos.h)
3. Visualização do Mundo dos Retângulos (visualizacao.c e visualizacao.h)
4. Interpretação de Comandos (comandos.c e comandos.h)

1. Função Principal (main.c e main.h)

Este módulo contém o ciclo principal (main) que lê e interpreta os comandos do utilizador. Ele exibe um menu de comandos disponíveis, solicita ao utilizador que digite um comando, e utiliza fgets para ler a entrada. O comando é então processado pela função interpretarComando localizada em comandos.c. O programa continua executando até que o comando "sair" seja dado.

Funções do main.c:

-Leitura de Comandos: Recebe entrada do utilizador.

-Interpretação de Comandos: Chama interpretarComando para processar a entrada.

-Exibição de Mensagens: Exibe mensagens de sucesso ou erros específicos.

-Ciclo Principal: Continua executando até a saída.

2.Gestão de Retângulos (retangulos.c e retangulos.h)

Este módulo gere os retângulos. Define uma estrutura *Retangulo* e utiliza um array estático para armazenar os retângulos existentes. Controla a criação, movimentação, fusão e a aplicação de gravidade aos retângulos.

Funções:

-*criarRetangulo*: Cria um retângulo se as coordenadas e dimensões forem válidas e não houver sobreposição, aplicando gravidade caso o retângulo tenha sido criado no "ar".

- Verifica se o retângulo está dentro dos limites permitidos;
- Verifica se há sobreposição com retângulos existentes;
- Se não houver sobreposição, adiciona o retângulo ao array e chama alertaSobreposicao e aplicarGravidade;
- Retorna um código de sucesso ou erro.

-*moverDireita* e *moverEsquerda*: Movem retângulos para a direita ou esquerda, aplicando gravidade após o movimento.

- Move um retângulo que está na posição (x, y) para a direita/esquerda pelo número especificado de posições;
- Verifica se o movimento está dentro dos limites permitidos;
- Aplica a gravidade após o movimento;
- Retorna um código de sucesso ou erro.

-*aplicarGravidade*: Move os retângulos para baixo até encontrar o limite inferior ou outro retângulo.

- Aplica a gravidade a todos os retângulos, movendo-os para baixo até que não haja mais espaço livre abaixo deles;
- Verifica se há espaço abaixo de cada retângulo e move-o para baixo se possível.

-*mergeRetangulos*: Une dois retângulos adjacentes, ajustando as dimensões do retângulo resultante.

- Une dois retângulos se possível, combinando suas dimensões;
- Remove o segundo retângulo do array após a união;
- Aplica a gravidade após a união.

-*alertaSobreposicao*: Verifica e alerta sobre sobreposições.

- Verifica todas as combinações possíveis de pares de retângulos para detetar sobreposições;
- Imprime uma mensagem de alerta se dois retângulos puderem ser unidos.

-*mergePossivel*: Verifica se dois retângulos podem ser unidos.

- Verifica se dois retângulos podem ser unidos, ou seja, se eles estão alinhados e podem formar um retângulo maior;
- Retorna 1 se a união for possível, caso contrário, retorna 0.

-*encontrarRetangulo*: Localiza um retângulo com base em coordenadas fornecidas.

- Encontra o índice de um retângulo que contém o ponto (x, y);
- Retorna o índice do retângulo encontrado ou -1 se nenhum retângulo contiver o ponto.

3.Visualização do Mundo dos Retângulos (visualizacao.c e visualizacao.h)

Este módulo trata da visualização do estado atual dos retângulos, desenhando-os em uma representação textual da tela. Foi adaptada da AF3, fazendo imprimir "x" nas bordas e "o" no centro.

A função *desenharMundo*, desenha a tela inicializando uma matriz 2D com espaços em branco e preenchendo-a com os retângulos. Imprime a matriz linha por linha, incluindo a numeração dos eixos X e Y.

- Inicializa cada posição da matriz tela com um espaço em branco;
- Define o último caractere de cada linha como nulo \0 para que a linha seja tratada como uma string;
- Itera sobre todos os retângulos e desenha cada um na matriz tela;
- Para cada retângulo, itera sobre suas posições e desenha "x" nas bordas e "o" no interior;
- Imprime as linhas da tela de cima para baixo, precedidas pela coordenada Y correspondente;
- Imprime os números do eixo X abaixo da tela, alinhando-os com as colunas correspondentes;
- Os números no eixo X são impressos a cada 10 colunas, e espaços são adicionados entre eles para manter o alinhamento.

4.Interpretação de Comandos (comandos.c e comandos.h)

Este módulo interpreta os comandos inseridos pelo utilizador e chama as funções apropriadas nos módulos de gestão de retângulos e visualização.

Temos apenas a função interpretarComando, que analisa a string do comando, identifica a operação a ser executada e chama as funções específicas para criar, mover, ou unir retângulos, e atualiza (ou não) a visualização consoante a escolha do utilizador.

Arquivos de Cabeçalho (.h)

Os arquivos .h correspondentes fornecem declarações de funções e definições de tipos compartilhados entre os módulos, permitindo que cada módulo seja desenvolvido e testado independentemente.

Conclusão

O programa desenvolvido no Efolio B é um sistema interativo que permite ao utilizador criar e manipular retângulos em uma área de desenho representada por uma grade. A organização modular do código facilita a manutenção e expansão do programa. Cada módulo tem responsabilidades bem definidas, garantindo um código mais limpo e estruturado.

A separação de funcionalidades em diferentes módulos não só facilita o desenvolvimento e a depuração, mas também permite que futuras melhorias ou novas funcionalidades sejam adicionadas de maneira eficiente e organizada. Utilizei apenas duas variáveis globais, pois visto o objetivo do programa é principalmente retângulos, achei que faria sentido. Também adicionei vários #define, para o caso de se adicionar mais funções ao programa, seja mais fácil perceber e analisar os returns, futuramente.

Anexos

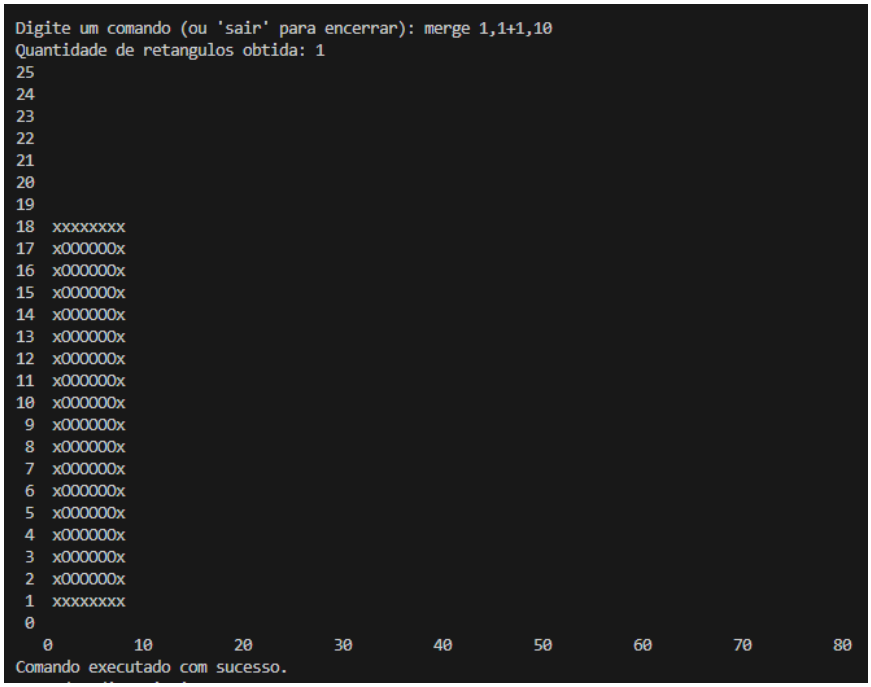
Teste 1

```
Digite um comando (ou 'sair' para encerrar): create 1,1+8,9
Retangulo criado com sucesso. Coordenadas: (1, 1), Largura: 8, Altura: 9
Quantidade de retangulos obtida: 1
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
 9  xxxxxxxx
 8  x000000x
 7  x000000x
 6  x000000x
 5  x000000x
 4  x000000x
 3  x000000x
 2  x000000x
 1  xxxxxxxx
 0
   0        10        20        30        40        50        60        70        80
Comando executado com sucesso.
```

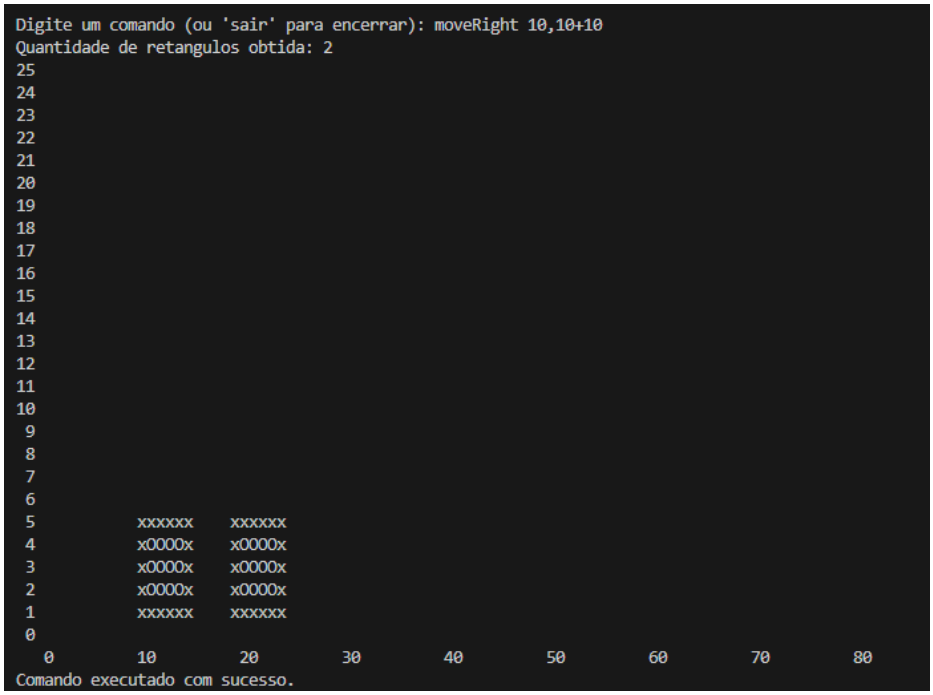
Teste 2(erro)

```
Digite um comando (ou 'sair' para encerrar): create 1,5+10,20
Comparando novo retangulo (1, 5, 10, 20) com retangulo existente 0 (1, 1, 8, 9)
Erro: Sobreposicao detectada com o retangulo 0.
Quantidade de retangulos obtida: 2
25
24
23
22
21
20
19
18  xxxxxxxx
17  x000000x
16  x000000x
15  x000000x
14  x000000x
13  x000000x
12  x000000x
11  x000000x
10  xxxxxxxx
 9  xxxxxxxx
 8  x000000x
 7  x000000x
 6  x000000x
 5  x000000x
 4  x000000x
 3  x000000x
 2  x000000x
 1  xxxxxxxx
 0
   0        10        20        30        40        50        60        70        80
Comando executado com sucesso.
```

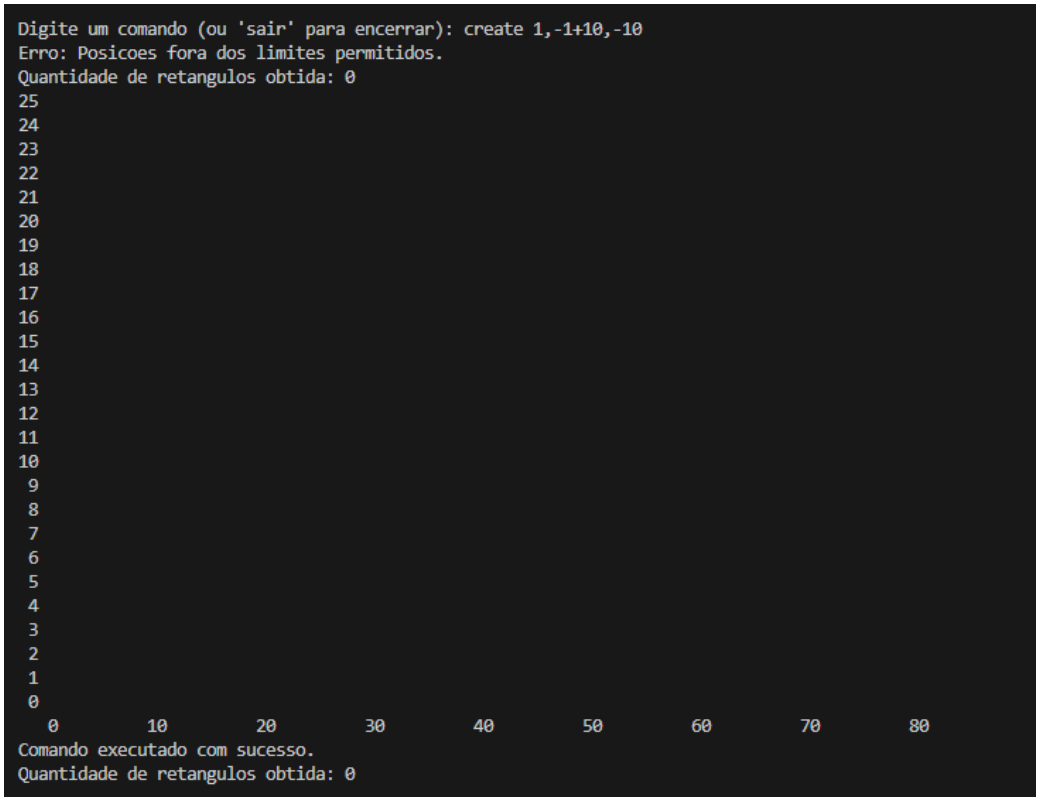
Teste 3



Teste 4



Teste5(erro)


Digite um comando (ou 'sair' para encerrar): create 1,-1+10,-10
Erro: Posicoes fora dos limites permitidos.
Quantidade de retangulos obtida: 0
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
0 10 20 30 40 50 60 70 80
Comando executado com sucesso.
Quantidade de retangulos obtida: 0