

Tema

Mundo dos retângulos

Objetivos

- Reutilização de código
- Melhoria e adaptação de código

Descrição da E-atividade

Escreva um programa para manipular um conjunto de retângulos num plano vertical, sem que haja interseções entre eles. O programa deve aceitar comandos para: criar retângulos com posição e dimensões indicadas; deslocar retângulos horizontalmente, para a esquerda e para a direita.

O plano que serve de cenário a este "mundo" baseia-se num sistema de coordenadas inteiras (x =abscissa, y =ordenada), em que $x, y \geq 1$, $x \leq 80$ e $y \leq 25$, e nele funciona a gravidade, isto é, todos os retângulos criados devem "cair" até ao chão (linha $y=1$), ou até encontrarem o topo de outro retângulo existente.

Os comandos a serem identificados pelo programa são:

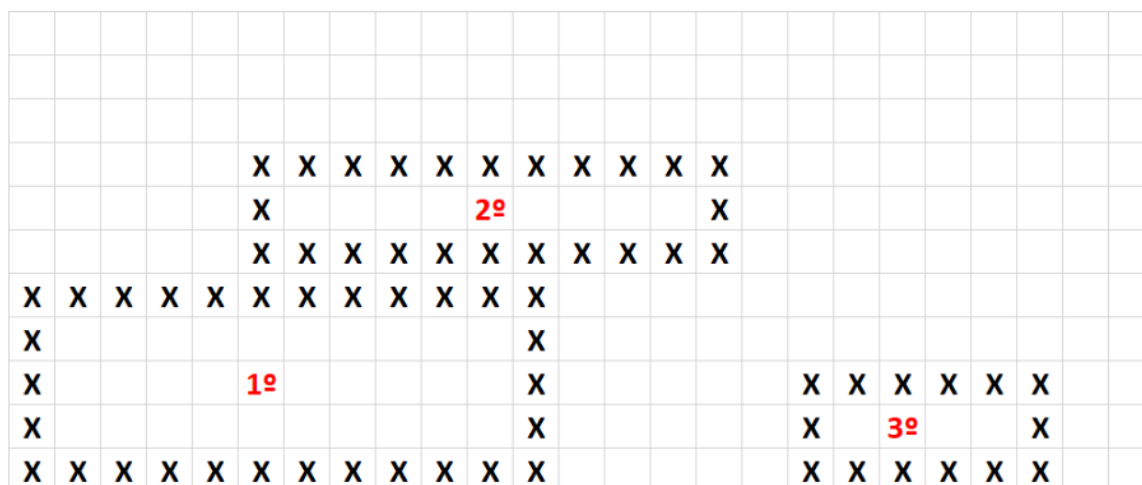
- create $x,y+l,h$ - cria um retângulo em que (x,y) são as coordenadas do canto inferior esquerdo e (l,h) o comprimento e altura, respetivamente.
- moveright $x,y+p$ - desloca o retângulo situado nas coordenadas (x,y) para a direita p posições
- moveleft $x,y+p$ - desloca o retângulo que contém o ponto (x,y) para a esquerda p posições

Qualquer destas operações, sempre que for possível (se não for, o programa deve emitir mensagem de erro adequada), implica ajustar todos os retângulos existentes tendo em conta o efeito da gravidade. O resultado dos comandos deverá poder ser visualizado graficamente no ecrã, usando o símbolo x para compor as linhas horizontais e verticais.

Exemplo E1: A seguinte sequência de comandos:

```
create 1,3+12,5
create 9,6+11,3
create 18,10+6,3
moveleft 12,7+3
```

deverá produzir o resultado:



(os números a vermelho não são para mostrar no ecrã, servem apenas para ilustrar a ordem da respetiva criação no exemplo dado). Na organização do seu programa, sugere-se que considere os seguintes módulos:

- análise do texto dos comandos;
- controlo das posições dos retângulos;
- visualização do resultado.

== METODOLOGIA DE TRABALHO ==

Defina uma estrutura de dados para representar os retângulos e planeie a organização do programa em módulos.

Use memória dinâmica e desenvolva uma função de testes, por exemplo com a sequência de comandos indicados em E1. Trate os erros do programa, por exemplo, valide e trate a introdução de dados no formato correto.

Defina a interface de cada módulo, desenvolva o código e teste o seu programa.

Nota: Não é necessário efetuar a animação dos retângulos a deslocarem-se, mostre apenas a posição final.

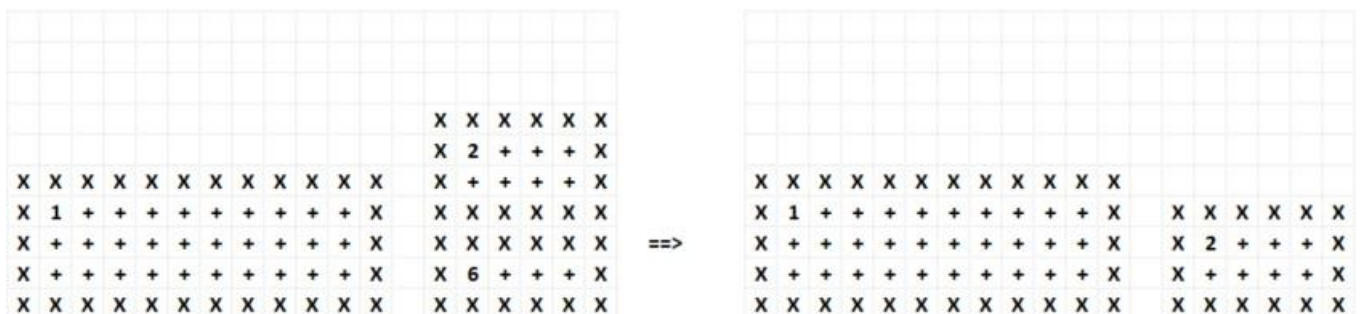
Trabalho a desenvolver

Considere o mundo dos retângulos desenvolvido.

Pretende-se que o interior dos retângulos seja preenchido com o símbolo “+” e que seja gerado, aleatoriamente, um número de identificação única do retângulo. A identificação do retângulo (de 0 a 9) deve ser apresentada numa qualquer posição do seu interior. O número máximo de retângulos possível dentro do mundo é 10.

Pretende-se ainda que seja possível apagar um retângulo. Deve usar o comando *delete x,y* que apaga o retângulo que contém o ponto (x, y) .

O comando delete 15,1 sobre o mundo representado à esquerda, deverá produzir o seguinte resultado no mundo representado à direita, após o efeito da gravidade:



Integre este comando *delete* no seu código.

Deteção de colisão lateral

Para além do comando *delete*, deverá também implementar um sistema de deteção de colisão lateral entre retângulos.

Uma colisão lateral ocorre quando, após a execução de um comando *moveleft* ou *moveright* e após a aplicação da gravidade), dois retângulos ficam lado a lado horizontalmente (ou seja, as faces verticais de dois retângulos ficam encostadas sem espaço entre elas em qualquer um dos pontos de limite lateral).

Sempre que um comando de movimentação (*moveleft*, *moveright*) for executado com sucesso (ocorreu um movimento de um retângulo e também após efeito da gravidade), o sistema deve emitir uma mensagem de alerta informando se ocorreu alguma colisão lateral, apresentando a identificação dos retângulos envolvidos na colisão. Por exemplo: “Colisão lateral detetada entre os retângulos 3 e 7.”

O programa deve permitir a introdução direta dos comandos no formato indicado (uma única linha de comandos) e não solicitar ao utilizador cada um dos parâmetros isoladamente.

Desta forma, sugere-se que desenvolva as funções:

- `alertaColisaoLateral()`: emite uma mensagem de alerta se verificar colisão lateral.
- `verificaColisaoLateral()`: avalia se dois retângulos estão em contacto lateral em qualquer um dos seus pontos laterais limite. Se dois retângulos ficaram “colados lateralmente” – total ou parcialmente - após a criação ou movimentação.

Acrescente à funcionalidade de testes realizada na AF3, o comando delete. Por exemplo:

Conjuntos de comandos	Resultado final após conjunto de comandos
create 1,1+15,5 create 5,6+12,3 create 22,10+6,9 create 21,12+6,7	
moveleft 23,11+2 moveright 10,8+2	
delete 23,2	

Notas: As cores são apenas auxiliares de visualização deste exemplo, não têm de ser implementadas. Os pontos indicados pelos comandos estão evidenciados com uma tonalidade diferente. A indicação de colisão não é necessária mostrar. Não é necessário mostrar as linhas de grelha nem a numeração das linhas. Após cada comando executado deve ser mostrado o resultado, para simplificação, este exemplo apresenta o resultado final da execução de um conjunto de comandos.

Componentes da Entrega:

- Código dos programas: O código do programa desenvolvido em formato de ficheiro editável ".c" e ".h", por exemplo: modulo1.c modulo2.c ficheiro.h ficheiro.c main.c.
- Ficheiro Executável do Programa (em Windows): O ficheiro executável dos programas deve ser fornecido exclusivamente e especificamente para o sistema operativo Windows, de forma a garantir que o avaliador consiga executar o programa.
- Relatório: Deverá ser entregue um relatório contendo no máximo 10 páginas (excluindo a capa e outras páginas que não referentes ao conteúdo E1, E2, E3, E4 e E5), no formato docx ou pdf.

Relatório

O relatório deverá indicar claramente os seguintes elementos:

- E1: Explicação da organização modular do seu código;
- E2: Estruturas de dados usadas na implementação;
- E3: Descrição da funcionalidade global do programa;
- E4: Descrição de como modificou ou adaptou o código da Atividade Formativa 3 para acomodar as novas funcionalidades;
- E5: Objetivos e descrição dos testes de unidade e integração efetuados.

Normas:

- O **código tem de compilar** de modo a poder ser avaliado.
- O código deve criado no Visual Studio Code e compilar corretamente no GCC com a configuração básica. Utilize um comando semelhante a: gcc modulo1.c modulo2.c main.c -o executavel.
- O código submetido deverá respeitar as normas ANSI C.
- Deverão ser utilizadas exclusivamente bibliotecas *standard*.
- Deve entregar dois ficheiros:
 - Um ficheiro word ou pdf com o relatório e também contém o conteúdo de todos os ficheiros de código desenvolvidos;
 - Um ficheiro zip com todos os ficheiros de código desenvolvidos e executável para sistema Windows.
- Não serão avaliados trabalhos em que o código contido no ficheiro do relatório não esteja em formato de texto, ou seja, o texto do relatório deve ser editável, não devendo estar convertido, por exemplo, em imagem.
- Os dois componentes (relatório + código e executável para sistema windows) não devem ser comprimidos em ficheiro zip, mas sim entregues isoladamente (ex: relatorio.pdf, codigo_e_exe.zip).

Instruções Adicionais:

- Certifique-se de que o ficheiro executável é compatível com o sistema operativo Windows.
- Verifique se todos os caminhos de inclusão de bibliotecas e ficheiros estão corretamente definidos para garantir uma compilação sem problemas.

Critérios de avaliação e cotação

- Modularidade e organização do código (Programa: 0,7 valores. Relatório: 0,2 valores)
- Clareza e legibilidade do código (Programa: 0,5 valores)
- Implementação de testes de unidade e de integração (0,3 valores)
- Funcionamento conforme requisitos (total de 1,6 valores):
 - Uso adequado de memória dinâmica (0,2 valores)
 - Implementação dos requisitos da AF3 (0,6 valores)
 - Visualização do mundo dos retângulos
 - Comando *create*
 - Comandos *moveleft* e *moveright*
 - Efeito da gravidade
 - Impossibilidade de interseção de retângulos
 - Comando delete, geração de identificação e limite de retângulos (0,3 valores)
 - Detecção de colisão lateral (0,3 valores)
 - Mensagens e alertas ao utilizador (0,2 valores)
- Grau de reutilização e facilidade das adaptações/modificações (Programa: 0,5 valores. Relatório: 0,2 valores)
- Ausência de ficheiro executável ou ficheiro não executável em sistema Windows (-0,5 valores)
- Existindo necessidade de editar as configurações do projeto para que eventualmente compile e possa ser avaliado o programa, o trabalho está sujeito a penalização de 0,5 valores.

Votos de bom trabalho!