

Programação Centrada em Objetos

Licenciatura em Tecnologias da Informação

Projeto - Fase 1 2022/2023

O objetivo final do projeto de PCO, feito em 3 fases, é pôr em prática os conhecimentos que vão sendo adquiridos nas aulas.

Nesta primeira fase do projeto vão exercitar as seguintes matérias lecionadas em PCO: declaração de variáveis, atribuição de valores a variáveis, expressões, abstração procedimental (definição de métodos *static* e sua invocação), comandos condicionais, ciclos e utilização dos tipos de dados não primitivos String e *array*.

Descrição do contexto onde se enquadra o objetivo deste trabalho

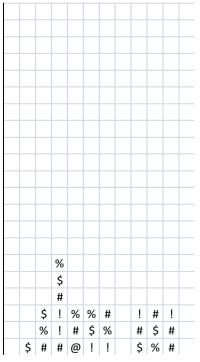
O *Columns* é um jogo de puzzle criado em 1989. Na sua versão original, o jogo usa uma área rectangular com 240 posições, distribuídas por 20 linhas com 12 colunas cada, na qual o jogador irá colocar peças. Cada peça consiste de 3 símbolos diferentes (cores diferentes no jogo original) que aparecem, uma de cada vez, no

topo da área de jogo e caem até chegarem ao fundo da área ou chegarem ao topo de outras peças anteriormente colocadas.

O jogador pode não só deslocar uma peça para a esquerda ou para a direita mas também alterar a ordem das cores da peça.

Quando a peça "aterra", se existirem 3 ou mais símbolos contíguos do mesmo tipo, quer na horizontal, na vertical ou na diagonal, então esses símbolos desaparecem. Peças que estejam colocadas em cima das que desaparecem, caem até atingirem o fundo da área de jogo ou até atingirem o topo de outras peças anteriormente colocadas. Se este reposicionamento colocar outras peças em posição de serem eliminadas, então o processo repete-se até não existirem mais peças que causem mais eliminações.





Nas várias fases deste projeto os alunos irão construir, a pouco e pouco, um jogo inspirado no Columns.

Nesta 1ª fase, a eliminação de sequências tem as seguintes restrições:

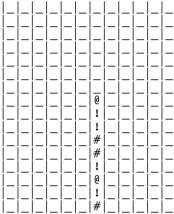
- A eliminação é apenas feita na vertical e na coluna na qual a peça foi inserida.
- A verificação de sequência a eliminar é feita do topo da coluna para o fundo da mesma.
- Caso após um acomodamento de símbolos na coluna, surjam mais sequências que devem ser eliminadas (tal como ilustrado num exemplo mais à frente), deve-se repetir todo este processo até não existirem mais sequências para eliminar nessa coluna.

Exemplos de colocação seguida de eliminação e acomodação

Usando uma representação textual simples, seguem-se exemplos que ajudam a perceber o mecanismo das ações a serem feitas para a colocação de uma peça na área de jogo.

Exemplo de uma colocação que não necessita de eliminação subsequente

Seja esta a grelha inicial: |-|-|-|-|-|-|-



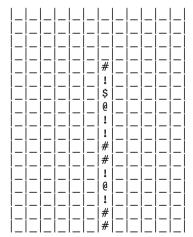
esta a peça a colocar:

! \$

e 7 a coluna onde colocar a peça.

A grelha resultante dessa colocação será a seguinte:

_|#İ



Como não há qualquer eliminação nem acomodação a fazer, neste caso o resultado do processo seria esta segunda grelha.

No caso de haver sequências de pelo menos 3 símbolos iguais contíguos na coluna em questão, esses símbolos deverão ser eliminados, seguindo-se a acomodação correspondente. Este processo deverá repetir-se enquanto houver eliminações a fazer ou até não haver mais símbolos nessa coluna.

Exemplo de uma colocação seguida de eliminação e acomodamento

Seja esta a grelha inicial:

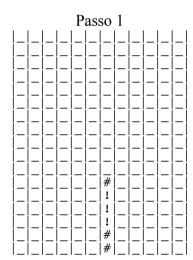
e esta a peça a colocar:

! !

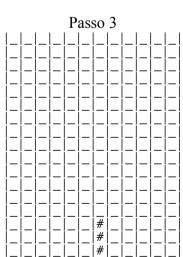
e 7 a coluna onde colocar a peça.

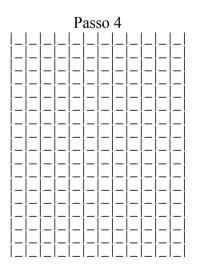
As seguintes transformações deverão ser aplicadas:

-! # #



Passo 2											
_	_	_	_	_	_	_	_	_	_	_	_
_	_	_	_	_	_	_	_	_	_	_	_
<u> </u> -	<u> </u>	<u> </u> _			_					_	-
-	-	—	-	-	_ _	-	-	-	-	-	-
-	—	-	_	-	-	-	-	-	-	-	-
\ <u>-</u>	-	¦-	¦-	¦-	<u> </u>	-	-	-	—	¦-	-
-	—	-	-	-	<u> </u> _	-	-	-	-	-	-
\ <u>-</u>	-	¦-	¦-	¦-	_	-	-	-	—	¦-	-
-	—	-	-	—	<u> </u>	-	-	-	-	—	-
\ <u>-</u>	¦-	¦-			-					¦-	-
\ <u>-</u>	-	¦-	¦-		_					¦-	-
-	-	-	-							-	-
-	-	-	-		<u> </u>					-	-
-	-	-	-		-			_	-	-	-
-	-	<u> </u>	-	-	-	#	-	-	-	-	-
-	-	—	-	—	-	#	-	-	-	—	-
1—	I —	I —	I —	I —	I —	"	-	_	—	I —	I — I





Neste caso o resultado do processo seria esta última grelha (passo 4).

O que se pretende de vós, afinal, nesta 1ª fase do projeto?

Para já, nesta primeira fase, é-lhes pedido para construirem vários métodos para que seja possível, dada uma coluna do jogo e uma peça, obter a coluna de jogo

resultante de fazer a colocação da peça e as eventuais eliminações e reposições de acordo com o descrito acima, mas tendo só em conta a direção vertical.

A coluna de jogo vai ser representada por um *array* unidimensional (vetor) de carateres; uma peça será representada por um *array* unidimensional de 3 carateres. **ATENÇÃO**: O último elemento do vetor coluna representa o elemento "de baixo" da coluna. O mesmo para a peça.

São-vos fornecidas as seguintes duas classes:

- TransformMethods, onde devem colocar os métodos que vão implementar; esta classe já contém um método que invoca os métodos em falta o resultado deste método applyTransforms é a coluna resultante de colocar uma dada peça numa dada coluna e aplicar as transformações necessárias;
- PCOFase1 que faz várias invocações dos vários métodos a implementar e que podem usar para testar os vossos métodos.

Como poderão ver no método applyTransforms incluído na classe fornecida TransformMethods, é necessário implementar os métodos seguintes:

- a) static char[] copyColumn(char[] col), que constrói e devolve uma cópia do vetor col;
- b) static void placePiece(char[] col, char[] piece) que modifica o vetor col "colocando-lhe por cima", se for possível, os elementos do vetor piece; se col não tiver 3 ou mais elementos vazios, nada será modificado;
- c) static boolean eliminate(char[] col) que modifica o vetor col, substituindo todos os elementos que formam sequências de 3 ou mais carateres iguais, pela constante *EMPTY*, e devolve true ou false conforme tenha feito alguma modificação ou não;
- d) static void accomodate (char[] col) que modifica o vetor col, acomodando todas as sequências contendo o caráter **EMPTY** de modo a que não figuem "buracos" vazios na coluna (rever passos 2 e 3 das figuras acima);

Outros métodos privados podem ser criados, de modo a controlar a complexidade das vossas tarefas. Para verem os resultados dos vossos métodos, poderão executar a classe PCOFase1, comentando as instruções que não querem que sejam executadas.

Avaliação

Para efeitos de avaliação desta fase do projeto, os métodos a) e b) valem 50% e os métodos c) e d) valem 25% cada.

Reparem que a classe PCOFasel está preparada para ser possível testar somente os métodos a) e b) ($Nivel\ 1$), testar esses dois e também o método c) ($Nivel\ 2$) e finalmente testar todos os métodos ($Nivel\ 3$).

Isto permite que os alunos que não consigam, por alguma razão, implementar todos os métodos, tenham hipótese de testar parte deles, desde que implementem pelo menos os métodos a) e b), ou os métodos a), b) e c). Basta para isso pôr em comentário, no método main da classe PCOFasel, as instruções correspondentes aos níveis que não desejam testar.

O que entregar?

Não há relatório a entregar porque o vosso *software* é a vossa documentação. Assim, <u>têm</u> que comentar condignamente todos os métodos da classe <u>TransformMethods</u>: incluir no início da classe um cabeçalho Javadoc com <u>Gauthor</u> (número do grupo e nome e número dos alunos que compõem o grupo); para cada método definido, incluir a documentação Javadoc apropriada.

Apresentem código que siga as normas de codificação em Java aprendidas nas aulas, bem alinhado e legível.

Para entregar: Um ficheiro *zip* com a classe que compõe a vossa solução e com os ficheiros Javadoc que geraram a partir dela (relembrem como fazer isto no Guião do Eclipse).

O nome do ficheiro zip que contém o vosso trabalho deverá ter o formato PCOxxx.zip (onde xxx é o número do vosso grupo).

Como entregar o trabalho?

Através do Moodle de PCO. Às 23h55 do dia acordado para a entrega, 26 de Outubro, os trabalhos entregues serão recolhidos.

Atenção que ao entregar o trabalho está a comprometer-se com o seguinte:

- O trabalho entregue é atribuível única e exclusivamente aos elementos que constituem o seu grupo;
- Qualquer indício de plágio será investigado e poderá levar ao não aproveitamento dos elementos do grupo e consequente processo disciplinar.

Outputs esperados:

Pode ver os outputs esperados da execução do método main da classe PCOFase1 no ficheiro outputs.txt fornecido.

Atenção que, por questões de não alongar muito o espaço ocupado para o output, as colunas são representadas na sua forma transposta, ou seja, "deitadas".