

Programação Orientada por Objectos (POO)

Semestre de verão de 2009-2010

Trabalho Final

LEETC – LEIC – LERCM

OBJECTIVOS

Consolidação dos temas tratados na unidade curricular. Pretende-se implementar o jogo *ooSameGame*, em Java, uma **versão adaptada** do jogo *SameGame* (descrito em <http://en.wikipedia.org/wiki/SameGame>).

INTRODUÇÃO

ooSameGame joga-se numa área quadrada (tabuleiro) preenchida por vários blocos de diferentes tipos colocados aleatoriamente. O jogador pode seleccionar e remover conjuntos de blocos adjacentes de acordo com o tipo de bloco (e.g. com a mesma cor). Cada conjunto de blocos tem de ter pelo menos dois blocos seleccionados. Em geral, os blocos que não são suportados por outros blocos (isto é, que não têm pelo menos um bloco por baixo) caem (efeito gravitacional) e as colunas vazias são removidas, existindo um deslocamento das colunas adjacentes para a esquerda. Este estilo de jogo designa-se por deslocamento gravitacional. Poderão existir outros estilos de jogo, incluindo outros tipos de deslocamento e adição de novas colunas (continuidade). O objectivo do jogo é remover o maior número de blocos maximizando o número de pontos. Os pontos são atribuídos em função do número de blocos removidos. O jogo termina quando não é possível remover mais blocos.

REQUISITOS

Estilos de Jogo

A aplicação a implementar inclui a interface gráfica com o utilizador e deverá suportar diferentes **estilos de jogo**. O jogador deverá poder jogar com *deslocamento gravitacional* ou *deslocamento gravitacional com efeito atractor*, em que no segundo caso, após cada remoção, além do efeito gravitacional, existem blocos que são deslocados para o lado esquerdo de modo a não existirem espaços vazios entre dois blocos na mesma linha. Deverá também ser possível outros estilos de jogo, resultantes da combinação dos anteriores com o *efeito da continuidade*, em que uma nova coluna de dimensão aleatória é adicionada à direita por cada coluna eliminada do tabuleiro.

Existem portanto quatro estilos de jogo:

1. deslocamento gravitacional (*gravitational*);
2. deslocamento gravitacional com efeito atractor (*shifter*);
3. deslocamento gravitacional com **continuidade** (*mega gravitational*);
4. deslocamento gravitacional com efeito atractor e **continuidade** (*mega shifter*).

Tipo de blocos

Os blocos poderão ser de um dos seguintes tipos ou cores:

- **Vermelho, Verde e Azul** – bloco que para ser removido precisa de pelo menos um bloco adjacente (na horizontal e na vertical) a ele do mesmo tipo;
- **Amarelo e Magenta** – bloco que para ser removido precisa de pelo menos um bloco adjacente a ele, incluindo diagonais, que seja do mesmo tipo;
- **Preto** – bloco que permite eliminar uma linha independentemente da cor dos blocos nessa linha;
- **Branco** – bloco que permite eliminar uma coluna independentemente da cor dos blocos nessa coluna.

Pontuação

A **pontuação do jogo** é calculada em função do tamanho dos grupos de blocos removidos durante o jogo. Pela eliminação de um grupo de blocos é atribuída uma pontuação igual a $(n-1)^2$, sendo n o número de blocos nesse grupo. A interface gráfica deve apresentar a pontuação do jogador durante o decorrer do jogo.

Deverá também ser mantido **registo das pontuações máximas**, por estilo de jogo, com o nome do jogador e respectiva pontuação. Cada registo deverá ter no máximo dez entradas.

Memorização do estado do jogo

A aplicação deverá permitir a **memorização do estado de um jogo**, de modo a que o utilizador possa recommençar a jogar a partir do estado guardado. Salienta-se que esta funcionalidade é semelhante ao “*auto-save*” disponível noutras aplicações.

Rotação de tabuleiro (extra)

Adicionalmente aos requisitos anteriormente descritos, valoriza-se a implementação da opção de **rotação do tabuleiro**. Nesta opção, deverá ser possível rodar o tabuleiro para a esquerda ou para a direita, sendo as peças deslocadas de acordo com o estilo de jogo actual.

SUGESTÕES

Na concepção da solução, elabore um esboço do diagrama UML estático de classes que representa os vários elementos que compõem o jogo e as relações entre eles. Antes de proceder à implementação, discuta-o com o docente.

Tenha em consideração a utilidade de separar (em hierarquias de classes distintas) a implementação da lógica do jogo da interface gráfica.

Inspire-se na versão original e, sobretudo, seja criativo.

ENTREGA DO TRABALHO

A data de entrega de uma versão funcional do trabalho é **16 de Julho**. A data de entrega da versão final e do relatório será combinada com o docente de cada turma. O relatório do trabalho deve evidenciar quais dos temas estudados ao longo do semestre foram aplicados e quais as vantagens resultantes da sua aplicação na implementação do jogo. Em particular, os seguintes temas: algoritmos recursivos; herança e polimorfismo; tratamento de excepções; *streams*; estruturas de dados; e programação *event driven*.

Bom trabalho,
Os docentes de POO