

PROJET DE PROGRAMMATION ORIENTÉE-OBJET

Année académique 2013-2014

Étape 2

Tetris

Lors de l'étape précédente, il vous a été demandé de créer le squelette de l'interface graphique ainsi qu'une première version du gestionnaire du jeu. Le but de cette nouvelle phase est d'étendre votre programme afin de permettre à un utilisateur de jouer une partie complète de Tetris.

Pour ce faire, il vous est demandé d'étendre le gestionnaire de jeu afin qu'il soit capable d'initialiser et de contrôler une partie. Le contrôle de la pièce courante s'effectue comme suit : les flèches gauche et droite permettent le déplacement latéral, la flèche vers le haut effectue une rotation (en respectant les rotations données dans l'annexe), la flèche vers le bas effectue une descente accélérée (*softdrop*), et enfin, la touche « **Enter** » fait arriver immédiatement la pièce à la position la plus basse possible sous la position latérale courante (*harddrop*). Après qu'une pièce ait terminé sa descente, le gestionnaire de jeu efface les lignes pleines et fait descendre les lignes non remplies situées au-dessus des lignes effacées. La touche du clavier « **p** » permet de suspendre temporairement le jeu et de reprendre le jeu suspendu. Lors d'une pause du jeu, la zone de jeu et l'aperçu de la pièce suivante doivent être cachés.

Le gestionnaire doit aussi comptabiliser le score obtenu et gérer les niveaux de jeu. Le score ajouté par ligne effacée dépend du nombre de lignes effacées simultanément et du niveau courant dans le jeu. À chaque augmentation de niveau de jeu, le rythme du jeu doit être accéléré. Une partie se termine s'il n'y a plus de place pour introduire la nouvelle pièce courante dans l'aire de jeu (suivant les règles précises données en annexe) ou si, au moment où la pièce finit sa course, elle ne se trouve pas entièrement dans la zone de jeu.

Votre programme doit générer les nouvelles pièces de façon aléatoire. Pour cela, on vous demande de développer votre propre générateur de nombres (pseudo-)aléatoires, **en interdisant** les classes fournies par Java, comme par exemple `java.util.Random`.

Informations et aide

Pour mener à bien votre projet, vous pouvez utiliser toutes les classes fournies par

les bibliothèques *Swing* et/ou *AWT* du langage Java en excluant les classes précédemment interdites.

De bonnes source d'informations vous sont données par *Oracle* sur les pages suivantes :

- <http://docs.oracle.com/javase/7/docs/api/index.html>
- <http://www.oracle.com/technetwork/java/index.html>
- <http://docs.oracle.com/javase/tutorial/>
- <http://www.oracle.com/technetwork/java/javase/overview/index.html>

Nous vous conseillons aussi de faire une relecture attentive du cours de programmation orientée-objet, principalement des exercices et des programmes vus au cours.

Consignes

On demande de rendre le travail sous la forme d'un code source en Java (correctement commenté) implémentant votre programme. De plus, vous devez fournir un rapport qui contiendra, au minimum, les informations suivantes :

- Une description précise des fonctionnalités implémentées et du programme (par exemple au moyen d'un ou plusieurs diagrammes).
- Une justification quant aux choix des structures de données utilisées.
- Une description des algorithmes complexes.
- La procédure complète pour compiler et exécuter le programme (n'hésitez pas à utiliser des utilitaires simplifiant la compilation tels que *make* ou *Ant*).
- Si votre rapport est une version complétée du précédent, une liste des modifications effectuées dans celui-ci.

Votre travail doit être rendu pour le lundi 28 octobre à 23h59 au plus tard à l'adresse suivante : proj-prog@montefiore.ulg.ac.be. Les travaux rendus en retard, les travaux plagés, les travaux non (ou mal) commentés et ceux ne respectant pas l'énoncé ne seront pas corrigés. Au moindre doute, contactez-nous!

Afin de pouvoir tester les programmes rendus, votre programme doit pouvoir être exécuté dans l'environnement des laboratoires de l'Institut Montefiore. Une démonstration dans cet environnement pourra d'ailleurs vous être demandée. Précisez sur quelle(s) machine(s) du laboratoire (ms8XX) votre programme a été testé.

Votre travail peut être rendu sous la forme d'une archive compressée `.tar.gz` ou `.zip` qui seront les seuls formats acceptés.

Bon travail!