

PROJET DE PROGRAMMATION ORIENTÉE-OBJET I

Année académique 2013-2014

Étape 5

Tetris

Le but de cette étape est de réaliser une version distribuée du programme, permettant des modes de jeux multiples à deux joueurs. Dans cette version, les joueurs sont sur des machines différentes qui interagissent avec un serveur. Chaque joueur utilise un module **client** qui communique par le réseau avec un module **serveur** qui gère la logique du jeu. Le module client se charge de l’affichage et de l’interaction avec le joueur.

Détails

Votre but est de transformer votre jeu Tetris en une application client-serveur. Pour ce faire, vous allez décomposer votre programme en deux modules indépendants qui peuvent être exécutés dans des processus séparés communiquant par l’intermédiaire d’un socket TCP.

Module serveur : le serveur doit être lancé en spécifiant un mode de jeu et un numéro de port. Une fois qu’il est démarré, il attend les demandes de connexions des clients sur le port donné. Dès que deux clients se sont connectés, le serveur leur envoie un message indiquant que la partie va démarrer. En cours de partie, le serveur envoie continuellement aux clients les informations relatives à la partie en cours. Précisons que le serveur ne doit pas obligatoirement posséder une interface graphique mais que, en revanche, il doit afficher dans une console et archiver dans un fichier `messages.log` l’ensemble des échanges effectués avec les clients.

Module client : un processus exécutant le module client doit rentrer en communication avec le serveur afin de pouvoir participer au jeu. Une fois que la partie est commencée, le client communique avec le serveur pour échanger les informations sur la partie en cours, et notamment, pour lui envoyer les événements déclenchés par l’utilisateur. Notons que le module client doit également offrir l’ensemble des fonctionnalités des étapes précédentes (autrement dit, il doit être toujours possible de lancer une partie locale par son intermédiaire).

Il est évidemment nécessaire de définir un protocole applicatif détaillant les messages pouvant être échangés entre client et serveur et leur signification. Il vous est demandé de décrire ce protocole dans votre rapport et d’expliciter les choix effectués dans l’élaboration de ce dernier.

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.

De bonnes sources 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

Le travail doit être rendu 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 10 mars 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 !