

## Projet : réalisation d'un jeu simple

Ce projet pourra, si vous le souhaitez, être réalisé à plusieurs (max. 3). La notation tiendra compte du nombre de participants et du niveau de difficulté du projet rendu. Une partie des séances de TP seront consacrées à la progression du projet. Au cours de ces séances, les enseignants suivront l'évolution de chacun des projets.

### 1 Description du projet

Le but est d'implémenter un jeu simple utilisant l'API graphique du Java (swing). Au cours des TP précédents, nous avons vu l'ensemble des éléments techniques, nécessaires à son implémentation. D'autres ressources et des exemples seront mis en ligne pour vous aider. Il est important que vous réfléchissiez bien aux structures de données que vous allez employer. Ensuite, vous pouvez procéder par étapes. Il est probable qu'au cours du développement vous vous rendiez compte que les structures de données ne sont pas adaptées. C'est normal, car il est difficile de tout prévoir dès le départ. Prenez alors le temps de reconcevoir votre programme. N'hésitez pas à demander de l'aide aux enseignants. Il est fortement recommandé d'avancer par petits pas. Si vous essayez de tout écrire d'un coup, vous avez très peu de chances d'y arriver. Voici une liste de jeux qui pourront être choisis. Pas plus de deux groupes ne devront travailler sur le même jeu.

- Pong
- Casse-briques
- Puissance 4
- Jeu d'échec ou de dames (plateau et déplacement des pièces)
- Jeu de go (plateau, placement et capture)
- Space invaders
- Pacman
- Tetris
- Jeu du serpent
- Démineur

Si vous avez d'autres propositions, n'hésitez pas à en parler à votre enseignant de TP.

### 2 Évaluation

Le projet comportera trois éléments d'évaluation :

- une note de suivi, déterminé lors des séances de TP. Votre enseignant de TP préviendra chaque groupe quand il souhaitera que vous lui présentiez l'avancement de votre projet ;
- le code source commenté du programme ;
- un court rapport.

#### 2.1 Critères d'évaluation du programme

Les critères suivants seront notés, par ordre d'importance décroissante :

- la qualité de la conception orientée objet ;
- la qualité des commentaires. Les commentaires doivent expliquer le sens de chaque bloc de code, ils sont adressés à un lecteur qui essaie de comprendre votre programme. Les commentaires ne doivent pas répéter ce qu'il y a dans le code. Les commentaires ne doivent pas expliquer les éléments syntaxiques, mais plutôt la sémantique.
- la lisibilité du code : choix des noms de classes, des membres, des variables et des fichiers. Votre programme doit être proprement divisé en plusieurs fichiers. Faites attention à l'indentation.
- le résultat et les fonctionnalités du programme.

## AVERTISSEMENT

La qualité prime sur la quantité. Un programme court mais bien organisé et bien commenté sera mieux noté qu'un gros programme mal conçu et incompréhensible.

## 2.2 Comment rendre les fichiers

Vous devez rendre le code source de votre programme ainsi que tous les fichiers nécessaires pour le compiler (fichiers `html` et éventuellement les images) et l'exécuter. Ces fichiers seront tous rendus sous la forme d'une archive `.tar.gz` unique qui devra être construite de la manière suivante : si les personnes participant au projet s'appellent Durand et Dupond, créez un répertoire appelé `Dupond-Durand`. Mettez tous vos fichiers dans ces répertoires. Placez-vous dans le répertoire parent de `Dupond-Durand` et tapez : **`tar zcf Durand-Dupond.tar.gz Durand-Dupond`**

Ceci va créer l'archive `Durand-Dupond.tar.gz`. Envoyez ce fichier par mail à l'enseignant de votre groupe de TP.

## 2.3 Rapport

De quelques pages, il devra expliquer la structure du programme et les grandes lignes de l'implémentation. La compréhension du code doit être facile après la lecture du rapport. Le rapport devra être dactylographié : un rapport écrit à la main ne sera pas accepté.