

## Projet de programmation objet n°1

(4 séances encadrées + travail personnel)

### 1 Objectif

L’objectif est de programmer un ensemble de classes en Java pour réaliser partiellement un jeu de Tetris en s’intéressant tout particulièrement au mouvement des pièces sur le plateau de jeu. Il n’est pas demandé de réaliser une interface graphique ni même de gérer une interaction au clavier pour jouer effectivement une partie. Par contre, il faut être capable de gérer au moins les fonctionnalités suivantes :

1. Créer un plateau de jeu ;
2. Créer les différentes pièces du jeu et les placer aléatoirement en haut du plateau ;
3. Déplacer les pièces vers la gauche, vers la droite ou vers le bas ;
4. Détecter si une ligne est complète sur un plateau afin de la supprimer ;
5. Détecter si une partie est finie ;
6. Simuler une partie en tirant aléatoirement les pièces et leurs déplacements ;
7. Afficher le contenu d’un plateau dans la console en mode texte.

### 2 Programmation

Le but est de proposer une solution dans le cadre de la programmation objet. Il est impératif de structurer votre programme selon un ensemble de classes bien choisies. Dans ce but, les classes suivantes sont imposées (et vous pourrez en ajouter) :

1. Une pièce est une instance de la classe **Piece** définie par un ensemble de cellules.
2. Une cellule est une instance de la classe **Cellule** définie par des coordonnées entières et un caractère (jouant le rôle de la couleur en mode graphique pour identifier les pièces).
3. Un plateau est une instance de la classe **Plateau** représentant une grille pouvant accueillir des pièces, chaque emplacement de la grille étant libre ou occupé par une cellule.
4. Une fabrique est une instance de la classe **Fabrique** qui crée chacun des types de pièces du jeu de Tetris.

De plus, les méthodes suivantes sont imposées (et vous pourrez en ajouter).

Dans la classe **Piece** des méthodes de déplacement, chacune créant une nouvelle pièce correspondant à un mouvement de **this** (sans modifier **this**) :

```
Piece versLeBas();    // déplacement de this vers le bas
Piece versLaGauche(); // déplacement de this vers la gauche
Piece versLaDroite(); // déplacement de this vers la droite
```

Dans la classe `Plateau` des méthodes de gestion des pièces :

```
boolean accepter(Piece p); // teste si this peut recevoir p
void retirer(Piece p);      // retire p de this
void ajouter(Piece p);      // ajoute p dans this
```

Ainsi, un mouvement d'une pièce sur un plateau consiste à créer une nouvelle pièce correspondant à ce mouvement (via les méthodes de la classe `Piece`), à tester si le plateau peut accepter la nouvelle pièce et si c'est le cas, à retirer l'ancienne pièce puis ajouter la nouvelle.

Dans la classe `Fabrique` les méthodes de génération des pièces, par exemple :

```
Piece creerCarre();           // retourne un bloc de 2x2 cellules
Piece creerBatonVertical();    // retourne un baton vertical de 4 cellules
Piece creerBatonHorizontal(); // retourne un baton horizontal de 4 cellules
```

Avant de commencer à programmer, il faut bien sûr réfléchir à la structure générale, quelques pistes étant proposées ci-après :

1. Se documenter sur le jeu de Tetris, les formes de pièces, etc.
2. Comment représenter une pièce ? Comment repérer une pièce sur un plateau ? Comment tester si un emplacement est libre sur un plateau ?
3. Comment générer une pièce dans la fabrique, avec quelles coordonnées pour ses cellules ?
4. Comment placer sur le plateau une pièce créée par la fabrique (considérer aussi l'aspect aléatoire sachant que la pièce peut être disposée n'importe où en haut du plateau) ?

### 3 Questions optionnelles

Ces questions peuvent être traitées si le temps le permet.

1. Réfléchir à la rotation des pièces. Votre solution permet-elle de gérer facilement les rotations sans trop de modifications ? Le but est ici d'avoir un regard critique.
2. Réaliser une interface graphique.

### 4 Réalisation et rendu du projet

Ce projet est encadré pendant les séances de TP n°3, 4, 5 et 6. Il est évalué sur la base des programmes sources réalisés en Java et d'un rapport au format pdf de 10 pages maximum constitué des parties suivantes : une introduction (avec la description du travail demandé), la description synthétique des classes (pour chaque classe : objectif, interface, structure de données et autres points importants) et des relations entre ces classes, les jeux de tests accompagnés d'une discussion sur les résultats obtenus et une conclusion.

Chaque binôme doit poster un fichier archive (avec le rapport et les programmes sources sans le bytecode) sur madoc en respectant les règles suivantes :

- Rendu au plus tard une semaine après la séance n°4 (à fixer dans chaque groupe).
- Format du fichier archive `zip` ou `tar.gz`.
- Dans un binôme, **une** personne poste le fichier dans le devoir lié à son groupe de TP.