

TP (binômes volontaires) Jeu de plateau (suite Morpion)

Algorithmique – Programmation FIP (ING39)

V. Aponte, P. Courtieu

Contraintes obligatoires et style objet

Vous devez obligatoirement démarrer à partir du corrigé du morpion (`git pull`) et suivre les directives ci-dessous.

Directive 1 : le projet de départ

Avant de commencer comprenez le code de la méthode `testVictoire` dans le corrigé. Notamment voyez comme l'utilisation des fonctions de voisinage rendent le code beaucoup plus facile à écrire. Lisez également les suggestions dans les commentaires à propos du puissance 4.

Directive 2 : calcul de coordonnées encapsulés

TOUS les calculs de coordonnées doivent être encapsulés dans la classe `Coordonnees`. Vous pouvez ajouter des méthodes à cette classe. Par exemple des méthodes du genre ci-dessous pourront être utiles (ça dépend du jeu que vous implémentez) :

```
Coordonnees[] allH(int n) // p.allH(n) retourne un tableau contenant les n coordonnees
                        // situées au-dessus de coordonnées (this).
Coordonnees[] allTo(Coordonnees c) // p.allTo(n) retourne un tableau contenant les
                                   // coordonnees situées entre this et c.
etc
```

De cette manière vous éviterez la duplication (et surtout la duplication des erreurs) de code concernant ces calculs.

Raffinement optionnel : vous pouvez même définir une nouvelle classe `CoordGrille` équivalente à `Coordonnees` mais prenant en compte les dimensions de la grille. En effet la classe `Coordonnees` correspond en fait à une grille infinie (il y a toujours une coordonnées voisine dans toutes les directions). Par exemple si `c` est une `CoordGrille` situé sur la première ligne de la grille alors `c.h()` lève une exception (il n'y a pas de voisine au-dessus). Notez que pour rester « propre » il faudrait remplacer `Coordonnees` par `CoordGrille` un peu partout dans le projet, y compris dans `Plateau` et `Morpion`. Cet option technique n'est qu'optionnelle.

Exercice 1 : puissance 4 dans le même projet que le (corrigé du) morpion

Le but de ce TP est d'utiliser et enrichir un ensemble de classes. Vous développerez le nouveau jeu de plateau « puissance 4 » **dans le même projet java** que le morpion (prenez le corrigé obligatoirement, voir plus bas) dans le package `jeu.puissance4`.

Dans ce jeu le joueur désigne toujours une case à chaque tour mais c'est seulement la colonne qui est prise en compte. Vous n'avez pas à changer la classe `IOGraphiquePlateau`.

Exercice 2 : un autre jeu toujours dans le même projet

Une fois le puissance 4 terminé (y compris les tests de victoire), vous pouvez démarrer un autre jeu. Exemples par ordre de difficulté approximative :

- petits chevaux (2 couleurs de pion, 4 pions par joueurs),
- dames (sans prise maximale)
- échecs
- dames avec la règle de prise maximale.

Vous pouvez proposer un autre jeu après validation auprès des enseignants.

Dans ces jeux le joueur désigne parfois deux cases (départ et arrivée). Deux appels consécutifs à `getCoup` étant suffisants Vous n'avez *toujours pas* à modifier la classe `IOGraphiquePlateau`.

Pièces mobiles et prises

Si votre troisième jeu (après le morpion et le puissance 4) implique que les pièces doivent se déplacer, vous devrez implanter les mouvements et leur validation *dans la classe de chaque pièce* (par exemple aux échecs les pièces de la classe `PionNoir` ne se déplacent pas de la même façon que ceux de la classe `Fou`), et ceci devra être déclaré dans une interface héritant de `Piece`, par exemple `PieceMobile`. De même si le jeu implique des « prises » (une pièce qui élimine une autre pièce du plateau) il faudra enrichir l'interface en conséquence. Par exemple il faudra que les mouvements et les prises soient calculables par des méthodes du genre :

```
boolean peutAllerA(Coordonnees c,Plateau p);  
... peutAllerAAvecPrise(Coordonnees c,Plateau p);  
... effectuerMouvement(...) // déterminer le type de retour et des arguments  
... effectuerPrise(...) // idem
```

Notez que les méthodes prennent le plateau en paramètre, en effet le déplacement et la prise dépendent du contenu du plateau.