

# Bases de la Programmation Orientée Objet

## Le morpion (Le jeu. Pas la p'tite bête!)

Jérôme Buisine, Rémi Cozot, Rémi Synave et Bruno Warin

### Travail préliminaire

Créez un répertoire `Morpion_votre_nom` dans votre répertoire `java`. Vous y placerez tous les fichiers utiles pour la réalisation du TP. vous avez **6 séances** pour faire ce TP. C'est un projet qui est fortement guidé. Merci de bien suivre le sujet.

## 1 Contexte

Nous sommes au début du mois de Mars 2020. Le monde est en panique face au coronavirus. A la télé, vous voyez les gens se ruer dans les supermarchés pour faire des stocks de vivres alimentaires. Personnellement, vous n'êtes pas sensible à cet affolement mais vous décidez tout de même d'aller faire vos courses juste au cas où...

Vous vous rendez donc au Carrefour que vous avez l'habitude de fréquenter. A Calais, point de désordre ou de ruée dans les magasins. Le calme règne. Vous balancez dans votre caddie légumes et fruits frais jusqu'à cette vision d'horreur. Vous vous êtes lourdement trompé! Les rayons sont vides! Plus aucune bière. Vous sentez une bouffée d'inquiétude monter et vous préférez quitter immédiatement les lieux. Vous laissez tout en plan et partez au méga Auchan de Coquelles. Mais là aussi, les gens ont vraiment fait des réserves et le rayon est vide également!

Dépité, vous rentrez chez vous et retrouvez votre colocataire à qui vous racontez l'histoire. Tout sourire, il ouvre son frigo et vous êtes rassuré. Il a pris les devants depuis bien longtemps et vous avez assez de réserves pour tenir au moins jusqu'à la fin de l'année scolaire. Sur cette bonne nouvelle, vous décidez de fêter ça en buvant de la triple Karmeliet. Au bout de 3 litres chacun, vous sortez une feuille, un crayon et commencez à jouer au Morpion. Il est 3h57 lorsque :

Votre coloc : - "Je suis sûr que tu es plus bourré que moi!"

Vous : - "N'importe quoi! Je pourrais encore réciter l'alphabet sans me tromper!"

Votre coloc : - "Trop facile. Petit défi pour nous départager?"

Vous : - "OK!"

Votre coloc : - "On code le jeu du morpion en utilisant la bibliothèque MG2D. Le plus lent

paie le kebab à l'autre pendant une semaine midi et soir."  
Vous : - "OK mais on reboit une bière avant."

Voici comment vous vous êtes retrouvé à coder le jeu du morpion un dimanche soir (ou lundi matin...).

## 2 Le projet

Le projet consiste à développer un jeu du morpion graphique. Ce premier projet du module **M2103** sera fortement guidé. Toutes les classes, attributs et méthodes sont donnés. Tout ce que vous devrez écrire sera expliqué. Vous devez absolument suivre le sujet.

## 3 Les classes à développer

Le jeu du morpion se joue sur un plateau à deux dimensions. Dans ce projet, vous allez utiliser un tableau statique (tableau dont la taille ne varie pas) à deux dimensions d'objets. Chaque case du tableau va contenir un objet permettant de stocker le joueur ayant pris cette case.

Dans un premier temps, créez une classe **Main** contenant un simple main et qui vous servira à tester les classes et méthodes que vous allez écrire. Chaque fonctionnalité doit être testée avant de passer à la suivante !

### 3.1 La classe **Case**

Cette classe va permettre de stocker le numéro du joueur occupant la case. Cette classe ne contiendra donc qu'un attribut privé de type **int** que vous appellerez **joueur**. Cet attribut ne pourra contenir que :

- 0 si la case n'est pas occupée. Ce sera donc sa valeur par défaut. (valeur utilisée par le constructeur par défaut).
- 1 si la case est occupée par le premier joueur.
- 2 si la case est occupée par le second joueur.

Vous écrirez toutes les méthodes vues en cours et nécessaires au bon fonctionnement de la classe : constructeurs, accesseurs, mutateurs, equals, toString.

### 3.2 La classe **Plateau**

Cette classe représentera le plateau du jeu. Le plateau est un tableau de  $3 \times 3$  dans lequel chaque joueur se positionnera pour tenter de créer un alignement de 3.

Créez la classe **Plateau** qui contiendra :

- Un attribut privé de type tableau statique à 2 dimensions de **Case**.
- Un constructeur par défaut permettant d'instancier un tableau de  $3 \times 3$  et de créer tous les objets de type **Case**.

- Une méthode de signature `Case getX(int i, int j)` retournant l'objet de type `Case` se trouvant en  $(i, j)$ . Attention aux sorties de tableau. Si un problème est détecté sur les valeurs de  $i$  ou  $j$ , un message d'erreur s'affichera dans la console et le programme se terminera.
- Une méthode de signature `boolean plein()` retournant vrai si la grille est remplie, faux sinon.
- Une méthode de signature `int gagne()` permettant de savoir si un joueur a gagné ou non. La méthode va parcourir les lignes, les colonnes, les diagonales pour vérifier si la grille est gagnante. Le numéro du joueur vainqueur sera retourné sinon 0.  
**Conseil :** Vous pouvez décomposer cette méthode en plusieurs sous méthodes qui seront privées dans la classe (mais ce n'est pas obligatoire).
- Une méthode `toString` retournant une chaîne de caractères permettant un affichage lisible dans le terminal.

### 3.3 La classe Main

Dans la classe `Main`, dans le programme principal, faites en sorte de pouvoir jouer une partie dans le terminal.

1. Affichage de la grille
2. On demande au joueur d'entrer la coordonnées X
3. On demande au joueur d'entrer la coordonnées Y
4. On place le pion si possible sinon on redemande des coordonnées
5. On vérifie s'il y a un gagnant.
6. On recommence avec l'autre joueur tant qu'il n'y a pas de vainqueur et que la grille n'est pas pleine

A la fin de la partie, on affiche la grille finale et on indique qui a gagné ou s'il y a égalité.

### 3.4 La classe InterfaceGraphique

Développez la classe `InterfaceGraphique` permettant de jouer en mode graphique. Vous utiliserez la bibliothèque MG2D disponible sur Moodle. Le principe est le même que le programme principal que vous avez écrit précédemment sauf que vous devez afficher et mettre à jour l'affichage à chaque coup joué. De plus, on ne demande plus au joueur les coordonnées dans le terminal. Le joueur clique directement dans l'interface graphique pour positionner son pion. Un exemple d'interface graphique est proposé en figure 1

## 4 Pour aller plus loin

Si vous avez terminé rapidement, vous pouvez développer l'une des fonctionnalités suivantes :

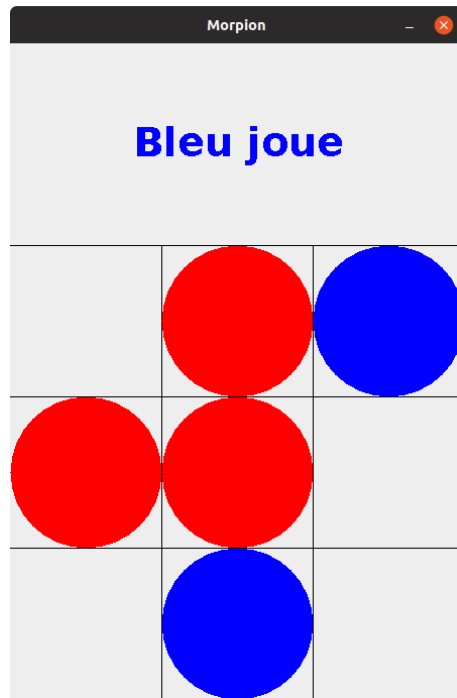


FIGURE 1 – Exemple d’interface graphique.

1. Faire un screenshot de la grille finale et l’enregistrer sur le disque.
2. Pouvoir paramétrer la taille du plateau.
3. Pouvoir paramétrer le nombre de pions à aligner pour gagner.
4. Afficher les pions gagnants dans le terminal (ou les mettre en surbrillance dans l’interface graphique).
5. Pouvoir paramétrer le nombre de joueur.
6. Programmer une IA.