

# Patrons de conception – TP 1 – Le jeu de *Hex*

L'objectif des TPs à venir est de mettre en œuvre différentes solutions inspirées des patrons de conception du *Gang of Four*, afin de développer une application Java permettant de jouer au jeu de Hex (cf. page Wikipedia). Dans le cadre de ces TPs, vous trouverez sur Moodle des éléments de code Java (archive `hex.jar`) ; il vous faudra développer plusieurs classes afin de les faire fonctionner ensemble et d'obtenir une application complète.

## 1. Création du projet

1. Créez un nouveau projet Java.
2. Ajouter l'archive `hex.jar` au *build path* de votre projet.
3. Téléchargez la documentation de l'archive sur Moodle et consultez-la.

## 2. Interface utilisateur

Pour le jeu Hex, les interactions avec l'utilisateur se résument à afficher le plateau de jeu, quelques informations supplémentaires et permettre le placement de pions.

1. Créez le paquetage `ihm` et codez-y une interface `IHM` offrant les méthodes suivantes :

```
public void actualiserPlateau(PlateauHex plateau);
public void afficherInfos(PartieHex partie);
public Coords demanderPlacement(PlateauHex plateau);
```
2. Tous les services d'une IHM en mode "ligne de commande" sont déjà codés dans le paquetage `hex.affich`, mais sont fournis par différentes classes.
  - Rappelez ce que sont les patrons de conception *Façade* et *Adapteur*.
  - Dans le paquetage `ihm`, créez une classe `IHMConsole` réalisant l'interface `IHM`.
  - Implémentez la classe `IHMConsole` en vous reposant sur les services fournis dans le paquetage `hex.affich`.

## 3. Gestionnaire de partie

L'application que nous développons n'est pas prévue pour gérer plusieurs parties simultanées. Afin de garantir l'unicité du gestionnaire de partie, vous allez mettre en œuvre le patron de conception *Singleton*.

1. Rappelez ce qu'est le patron de conception *Singleton*.
2. Créez le paquetage `ctrl` et codez-y l'interface `GestionnaireDePartie` fournissant les méthodes suivantes :

```
public void nouvellePartie(int taillePlateau);
public void jouer();
```
3. Dans le paquetage `ctrl`, proposez une classe `Gestionnaire` implémentant cette interface et mettant en œuvre le patron de conception *Singleton*.
  - L'instanciation du gestionnaire doit être assurée par la classe `Gestionnaire` elle-même (le constructeur est donc privé).
  - La méthode `nouvellePartie` assure l'initialisation d'une nouvelle partie qui sera stockée comme un attribut privé du gestionnaire.
  - La méthode `jouer` repose sur une `IHM` et assure la boucle de jeu selon le principe suivant :

```
tant que la partie n'est pas finie
    afficher le plateau
    afficher les informations du tour
    demander au joueur courant son prochain coup
    mettre à jour la partie en cours
fin tant que
afficher le plateau
afficher les informations
```

## 4. Créer le lanceur

Si vous ne l'avez pas encore fait, il est grand temps de créer une classe principale (`main`) permettant de lancer effectivement une partie.

1. Dans le paquetage `main`, créez une classe principale `Lanceur`.
2. Instanciez-y un gestionnaire de partie, créez une nouvelle partie puis appelez la méthode `jouer`.
3. Jouez contre vous-même ou avec votre voisin.e (si son programme est terminé, bien sûr).