

**TD/TP 6****CONCEPTION ET  
REALISATION DU MODELE DE  
PIECES ET DE PLATEAU DE JEU**

**Cette semaine, on se concentre sur les parties essentielles du modèle de jeu. Nous intégrerons plus tard les aspects graphiques et événementiels.**

**1. Mise en place du pattern Observer**

Créez vos propres classes et interfaces de mise en place d'un pattern Observer, en vous inspirant de la page 10 du CM.

La classe AbstractModeleEcoutable pourra ainsi être la classe mère des classes de modèles que vous voulez rendre écoutables.

Testez vos classes en créant un exemple dans lequel une vue écoute un modèle, comme indiqué dans la page 11 du CM.

**2. Le cas de modèles complexes : premiers pas sur la partie événementielle du CC**

Dans des cas plus complexes comme par exemple dans votre devoir de CC, il est courant qu'un modèle soit lui-même composé de plusieurs sous-modèles. Dans votre cas, vous aurez par exemple un PlateauJeu qui comprend l'ensemble des éléments du jeu, comme les différents personnages, les mines déposées, etc.

Dans ce cas, si l'un des personnages se déplace, il faut que le plateau de jeu prévienne ses listeners qu'il a lui-même changé, afin notamment que sa ou ses vues se mettent à jour (l'idée étant que l'on peut considérer qu'un conteneur d'objets a changé à partir du moment où l'un de ses composants a changé).

Une façon élégante de gérer cela est de procéder ainsi :

- Chaque élément (Personnage, Bombe, etc.) est écoutable
- Un PlateauDeJeu écoute chacun des éléments qu'il contient
- Lorsqu'un élément a changé (par exemple la méthode déplacement d'un Personnage), le plateau est donc prévenu (qui est lui aussi écoutable) déclenche alors à son tour un événement
- On crée ainsi une chaîne événementielle qui ira notamment jusqu'à la vue qui se mettra alors à jour.

Mettez en place cette architecture avec les éléments de base de votre application (il vous faut pour commencer a minima une classe PlateauJeu et une classe Personnage). Créez

un classe `VuePlateauJeu` qui, pour le moment, se contente de faire un `println("le modèle a changé")` dans sa méthode événementielle. Créez une instance de `PlateauJeu`, ajoutez-y deux `Personnages`, créez une vue associé au plateau, déplacez l'un des deux `Personnages`, et vérifiez bien que le `println` s'exécute. Le cas échéant, vous avez réussi à mettre en place l'architecture événementielle de base de votre application.

### 3. Suite du travail sur le devoir de CC

La fin de ce TP sera consacrée à la suite de la mise en place du devoir de CC, avec dans un premier temps un travail sur la partie « modèle », avec les classes permettant de mettre en œuvre la partie métier du jeu : un plateau de jeu contenant des personnages, des bombes, des mines, et les différentes méthodes permettant de faire interagir tout ce petit monde selon les règles du jeu.

La partie événementielle de ce modèle s'appuiera sur la base que vous venez de constituer dans la question précédente.