

# CHOIX DES DESIGN PATTERN

## 1. MVC

Nous allons utiliser le Design Pattern MVC pour avoir un code structuré.

Nous aurons un package Controller qui se chargera de mener les actions en modifiant les valeurs du modèle.

Nous aurons un package Model qui se constituera d'une arborescence de classe pour pouvoir accéder aux données de tous les composants de notre application.

Nous aurons un package View qui sera notre affichage, et qu'il s'adaptera en fonction du modèle.

## 2. Singleton

Le Design Pattern Singleton sera là pour nous permettre d'instancier qu'une seule fois certaines classes et ainsi empêcher les doublons et le mauvais fonctionnement de certaines classes.

## 3. Façade

Le Design Pattern façade nous aidera à avoir un code encore mieux structuré nettement avec des classes façade pour avoir une vue d'ensemble de toutes méthodes, attributs de nos packages.

## 4. Observer

Le Design Pattern observer nous permettra de faire interagir le package Model avec celui de la View. On pourra à chaque changement d'état d'un attribut dans le modèle, notifier la vue de ce changement et ainsi la vue pourra afficher les bonnes données aux bons moments.

## 5. Strategy

Le Design Pattern strategy nous permettra de sélectionner le comportement du client à n'importe quel temps. Nous allons devoir créer une classe de client mais avec plus de comportements qui seront capables d'être switchés.

## 6. Builder

Le Design Pattern builder nous permettra de créer les objets et comment elles seront utilisées dans l'application. En fait, ce DP nous permet de résoudre le problème de comment contrôler les objets créés.