

## P1 – MVC – Cartagena (1)

*Cette série de TD a pour but de vous faire mettre en œuvre les JavaBeans et le DP MVC dans un exemple moins trivial que le précédent et de vous faire apparaître les avantages du développement par composants.*

### LE MODÈLE

L'an passé en première année vous avez eu à rédiger le modèle du jeu de **Cartagène** que vous avez mis en œuvre au travers d'une interface 'console'.

L'énoncé de ce travail vous est fourni en annexe ainsi que le .jar d'un modèle opérationnel et sa Javadoc (vous pouvez utiliser celui qui vous est fourni mais il serait plus intéressant de récupérer celui que vous avez écrit l'an passé).

### UN COMPOSANT DE LA VUE

Vous pourriez concevoir votre vue comme vous l'entendez mais cet exercice présentant une certaine complexité, nous allons décrire quelques classes utiles.

Nous allons débiter par concevoir un composant simple: **CarteVue**

**CarteVue:** représentation graphique d'une carte

Nous allons concevoir ce composant graphique en tant que JavaBean héritant de JPanel.

Une CarteVue devra permettre de représenter visuellement une Carte grâce à une image du Symbole porté par la Carte<sup>1</sup>.

Cette CarteVue devra pouvoir être sélectionnée et désélectionnée au travers d'un clic gauche de souris et émettre un PropertyChangeEvent lors de ce clic si sa propriété 'sélectionnable'<sup>2</sup> est à vrai.

Nous devons aussi prévoir une méthode `setSelectionnable(boolean b)` et une méthode `isSelectionnable()`.



<sup>1</sup> Nous vous fournissons des images jpeg 80x80 des différents symboles (celles dont le nom se termine par GD). N'hésitez pas à prendre vos propres images.

<sup>2</sup> Le composant a donc bien deux attributs différents; sélectionné et sélectionnable.

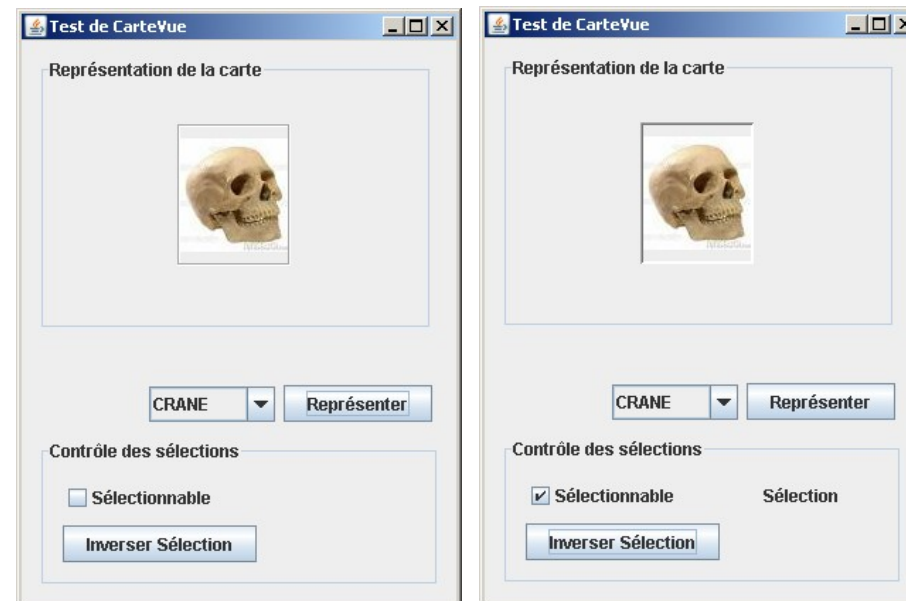
Pour assurer la correspondance entre **Symbole** et les images à afficher, nous utiliserons une classe utilitaire `ConversionVersGUI` offrant une méthode statique<sup>3</sup> `Image getImageDuSymbole(Symbole symbole)`.

Pour afficher l'image, utilisez la méthode `drawImage()` de `Graphics` dans la méthode `paintComponent()` du `JPanel`.

**Remarque:** vous aurez à placer le .jar fourni dans les bibliothèques de votre application.

### TEST DU COMPOSANT CARTEVUE

Pour permettre de tester la validité de votre développement, créez une JFrame analogue à celle présentée ci-dessous.



<sup>3</sup> Comme la méthode est statique, je ne pourrai pas faire appel à la méthode `getToolkit()` (comme expliqué dans le Guide Graphics) ... mais je peux utiliser une `ImageIcon` et lui demander de retourner son `Image`.