# Programmation avancée – Java
# TD n°4 : Retour sur la librairie `swing`

Sebastien.Varrette@imag.fr

L'objectif de ce TD est de revenir sur la programmation d'interfaces graphiques en utilisant la librairie `swing`[1]. Un aperçu rapide des composants de cette librairie est disponible à l'url `http://java.sun.com/docs/books/tutorial/ui/features/`. On rappelle le contenu du programme affichant HelloWorld :

```
import java.awt.*;
import javax.swing.*;

/**
 * Application qui affiche "HelloWorld"
 */
public class HelloWorldApp {
    public static void main (String args[]) {
        JFrame app = new HelloWorldFrame("Premiere Application", "HelloWorld!");
        app.setSize(250,100);
        app.setVisible(true); // affichage effectif
    }
};

class HelloWorldFrame extends JFrame {
    HelloWorldPanel _contentPanel; // Le contenu de cette frame
    private static final long serialVersionUID = 1L; // pour éviter un warning

    public HelloWorldFrame(String title, String msg) {
        super(title); // appel du super-constructeur
        _contentPanel = new HelloWorldPanel(msg);   //créé le contenu
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);//ce qui arrive à la fermeture

        // On ajoute le contenu
        getContentPane().add(_contentPanel);
    }
};

class HelloWorldPanel extends JPanel {
    private JLabel _label;
    private static final long serialVersionUID = 1L;

    public HelloWorldPanel(String msg) {
        _label = new JLabel(msg, JLabel.CENTER);
        setLayout(new BorderLayout()); // Choix de l'organisation du contenu
        add(_label);
    }
};
```

[1] On pourra se référer au TP G de l'an passé.

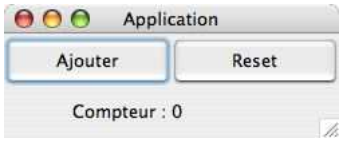
On rappelle également que les composants intégrés dans un conteneur suivent les règles d'un gestionnaire de mise en page (Layout Manager) qui définit la position de chaque composant inséré. On utilise la méthode `setLayout()` pour définir le gestionnaire que l'on souhaite utiliser. On utilise principalement :

- `FlowLayout` qui place les composants ligne par ligne de gauche à droite. Chaque ligne (centrée par défaut) est complétée progressivement jusqu'à être remplie, puis passe à la suivante. C'est la mise en page par défaut des applets ;
- `BorderLayout` qui découpe la surface en cinq zones : North, South, East, West, Center. On peut librement utiliser une ou plusieurs zones ;
- `BoxLayout` pour lequel les éléments sont alignés verticalement ou horizontalement ;
- `GridLayout` qui établit un réseau de cellules identiques qui forment une sorte de quadrillage invisible : les composants sont organisés en lignes et en colonnes. Les éléments insérés dans la grille ont tous la même taille ;
- `GridBagLayout` est le plus riche des gestionnaire : le conteneur est divisé en cellules égales mais un composant peut occuper plusieurs cellules de la grille et il est possible de faire une distribution dans des cellules distinctes. Un objet de la classe `GridBagConstraints` permet de donner les indications de positionnement et de dimension à l'objet `GridBagLayout`.

Enfin, on rappelle que les interfaces de gestion des événements sont résumées dans l'énoncé du précédent TP.

**Exercice 1**

On demande de réaliser une application qui ne contient qu'un bouton et un compteur, ce dernier affichant le nombre de fois où le bouton aura été cliqué. On pourra ajouter un button qui reinitialise le compteur. Un exemple du résultat obtenu est donné en figure 1(a).




(a) Application ButtonApp


(b) Application RochambeauxApp

**Exercice 2** *Pierre-papier-ciseaux*

Simuler le jeu de Rochambeaux : le joueur et l'ordinateur choisissent "Pierre", "Papier" ou "Ciseaux". Les règles sont les suivantes :

- "Papier" gagne sur "Pierre" ;
- "Pierre" gagne sur "Ciseaux" ;
- "Ciseaux" gagne sur "Papier".

On affichera le score (coloré). Un exemple d'exécution est donné dans la figure 1(b).