

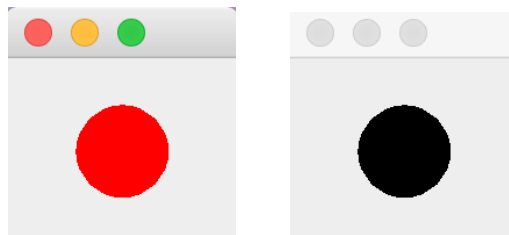
TP 10 : Les Interfaces Graphiques (II)

christina.boura@uvsq.fr, stephane.lopes@uvsq.fr

22 avril 2016

1 Fenêtre active

Le but de cet exercice est de programmer le comportement suivant lié à une fenêtre. Lorsque la fenêtre est active, un disque rouge doit s'afficher au milieu de celle-ci. De l'autre côté si la fenêtre est inactive le disque affiché doit se colorer en noir, comme illustré dans l'image suivante.



Pour réaliser votre application, vous pouvez écrire les trois classes suivantes :

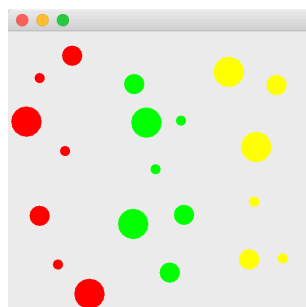
1. Une classe `Panneau` extends `JPanel` pour représenter le panneau sur lequel le disque s'affichera. Cette classe aura un seul attribut qui sera une variable booléenne indiquant si le disque est rouge ou noir. Une seule méthode y sera présente,

```
public void paintComponent(Graphics g)
```

dont le rôle sera de dessiner le disque.
2. Une classe `Fenetre` extends `JFrame` implements `WindowListener` pour représenter la fenêtre principale de l'application. Cette classe doit définir les méthodes décrites par l'interface `WindowListener`, notamment les méthodes `windowActivated` et `windowDeactivated`.
3. Une classe de test.

2 Dessiner des disques colorés avec un clic de la souris

Le but de cet exercice est de créer une fenêtre dans laquelle on dessinera des disques colorés à l'aide de la souris. Si le clic de la souris est sur la partie gauche de la fenêtre, le disque affiché doit être rouge, si on clique au milieu de la fenêtre le disque doit être vert et si on clique sur la partie droite le disque doit être jaune. Par ailleurs, la taille des disques doit varier selon le bouton de la souris appuyé. Si le bouton gauche est appuyé le disque doit être petit, si la molette est appuyée le disque doit être de taille moyenne et si le bouton droit est appuyé alors le disque doit être grand.



1. Vous pouvez commencer par créer une classe `Disque` pour représenter un disque coloré. Cette classe aura quatre attributs : les coordonnées `x` et `y` du cercle (point à haut à droite du carré entourant le cercle), son rayon et une couleur (classe `java.awt.Color`).
Créez également la méthode `void dessiner(Graphics g)` qui dessine un disque coloré.
2. Créez une classe `Panneau` `extends JPanel` pour représenter le conteneur principal de la fenêtre qui va accueillir les disques colorés. Ajoutez un écouteur `MouseListener` et implémentez la méthode `public void mousePressed(MouseEvent e)` qui doit décrire les actions à faire lorsque un bouton de la souris est appuyé. Pour déterminer lequel des trois boutons de la souris a été appuyé vous pouvez utiliser les méthodes de la classe `javax.swing.SwingUtilities`.
3. Créez une classe de test contenant la méthode `main` de l'application.

3 Changer la police d'écriture d'un texte

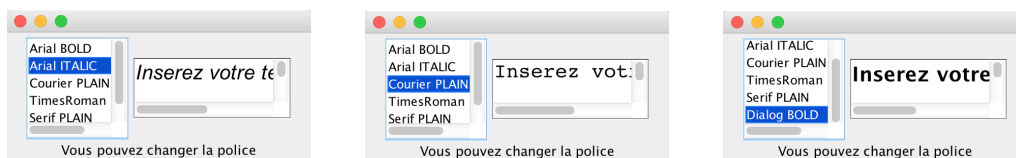
On veut dans cet exercice créer la fenêtre de l'image ci-dessous. Cette fenêtre comportera trois composants :

1. Une étiquette non-sélectionnable avec le texte " *Vous pouvez changer la police*" en bas de la fenêtre.
2. Une liste de choix de police munie d'ascenseurs à gauche de la fenêtre.
3. Une zone de texte à droite de la fenêtre munie également d'ascenseurs.

Chaque fois que l'utilisateur choisit une police dans la liste de gauche, le texte à droite doit s'afficher en utilisant la police correspondante.

La liste de gauche doit être composée des six choix suivants :

- Arial BOLD
- Arial ITALIC
- Courier PLAIN
- TimesRoman ITALIC
- Serif PLAIN
- DIALOG BOLD



Commencez par créer une classe `Police` qui aura deux attributs privés : le nom de la police (`Aria`, `Courier`, `etc.`) et le style de la police (`PLAIN`, `BOLD`, ou `ITALIC`). Le style est représenté par un entier : 0 pour `PLAIN`, 1 pour `BOLD` et 2 pour `ITALIC`. On suppose ici que la taille de la police est constante. Créez une méthode `public String toString()` qui affichera le nom de la police suivi du style, par exemple

TimesRoman PLAIN.

Créez également une méthode `getNom()` et une méthode `getStyle()` qui retourneront le nom et le style de la police respectivement.

Écrivez maintenant une classe

```
public class Fenetre extends JFrame implements ListSelectionListener
```

pour représenter la fenêtre principale. Cette classe aura trois attributs :

- une instance de `JList<Police>` pour représenter la liste des choix du composant à gauche. Chaque élément de cette liste est une instance de la classe `Police`.
- une instance de la classe `JTextArea` pour représenter le composant texte à droite de la fenêtre.
- une instance de la classe `Police` pour garder la police courante.

Pensez à ajouter un écouteur `ListSelectionListener` à la liste des choix et implémenter la méthode

```
public void valueChanged(ListSelectionEvent e)
```

pour indiquer ce qui doit se passer lorsque l'utilisateur sélectionne un élément de la liste.

Pour le reste, laissez vous inspirer des exemples traités pendant le cours.