Chapitre 4

Événements

Traitement des événements

L'utilisateur va intervenir sur le programme via le clavier ou la souris. Le programme devra associer des traitements aux actions possibles de l'utilisateur.

On distingue deux types d'événements:

- événements de bas niveau: appuyer ou relâcher un bouton de souris.

- événements logiques: clic sur une souris.

Si vous appuyez sur la lettre A, vous produisez les événements suivants:

- 4 événements de bas niveau:
Appuie sur la touche shift
Appuie sur la touche A
Relâchement de la touche A
Relâchement de la touche shift

- 1 événement logique:
Frappe du caractère A

Les événements sont représentés par des instances de sous-classes de java.util.EventObject

Événements de bas niveau : KeyEvent (action sur une touche), MouseEvent (action sur la souris)

Événements de haut niveau : FocusEvent (une fenêtre qui prend le focus ou la main), WindowEvent (fenêtre fermée, ouverte, icônifiée), ActionEvent (une action est déclenchée), ItemEvent (choisir un Item dans une liste), ComponentEvent (un composant caché, montré, déplacé, retaillé)

Dans le paragraphe qui suit, nous allons nous intéresser plus particulièrement à ActionEvent. Puis par la suite, nous introduirons au fur et à mesure les autres événements.

ActionEvent

Chacun des composants graphiques a ses écouteurs (listeners) qui s'enregistrent (ou se désenregistrent) auprès de lui comme écouteur d'un certain type d'événement (par exemple, clic de souris)

Un objet écouteur intéressé par les événements de type "action" (classe ActionEvent) doit appartenir à une classe qui implémente l'interface java.awt.event.ActionListener

Définition de ActionListener :

```
public interface ActionListener extends EventListener {
     void actionPerformed(ActionEvent e);
}
```

La méthode actionPerformed représente le message qui sera envoyé à l'écouteur.

On inscrit un tel écouteur auprès d'un composant nommé bouton et cela comme suit : **bouton.addActionListener (ecouteur);**

Quand un événement, un ActionEvent, est engendré par une action de l'utilisateur sur le bouton qui envoie le message actionPerformed() à l'écouteur. Le bouton lui passe l'événement déclencheur : ecouteur.actionPerformed(unActionEvent);

(Pour plus de détails voir les exemples Compteur.java et OvaleScrollbar.java)

Pour la souris, elle peut avoir 7 actions possibles :

- MouseListener

```
public void mousePressed(MouseEvent e) ...
public void mouseClicked(MouseEvent e) ...
public void mouseReleased(MouseEvent e) ...
public void mouseEntred(MouseEvent e) ...
public void mouseExited(MouseEvent e) ...
```


- MouseMotionListener

Bouton de la souris est pressé souris déplacée

```
public void mouseDragged(MouseEvent e) ...
```

Souris déplacée et le curseur se trouve dans un composant

```
public void mouseMoved(MouseEvent e) ...
```

Implanter « MouseListener » signifie redéfinir toutes les méthodes! Une solution consiste à utiliser « MouseAdapter » comme suit :

```
class MouseAdapter implements MouseListener {
     public void mousePressed(MouseEvent e) {}
     public void mouseClicked(MouseEvent e) {}
     public void mouseReleased(MouseEvent e) {}
     public void mouseEntred(MouseEvent e) {}
     public void mouseExited(MouseEvent e) {}
     public void mouseDragged(MouseEvent e) {}
     public void mouseMoved(MouseEvent e) {}
}

class EcouteMaSouris extends MouseAdapter {
     public void mouseClicked(MouseEvent e) {
          // bla bla bla bla
     }
}
```

```
class MaFenetre extends Applet {
     addMouseListener(new EcouteMaSouris());
}
```

```
class MaFenetre extends Applet {
     addMouseListener( new MouseAdapter {
               public void mouseClicked(MouseEvent e) {
                    // bla bla bla bla
               }
          }
     );
}
```

Pour retrouver lequel des boutons qui a été cliqué, il faudra utiliser l'une des méthodes :

```
InputEvent.BUTTON1_MASK (gauche)
InputEvent.BUTTON2_MASK (central)
InputEvent.BUTTON3_MASK (droite)
```

MouseEvent → getModifiers

```
if (e.getModifiers() & InputEvent.BUTTON1_MASK) != 0) blablabla ...
```

(Pour plus de détails voir l'exemple Pong.java)