



Chapitre 8

Clavier et souris

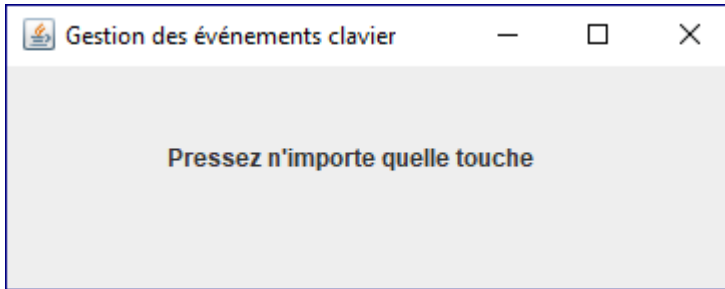
Gestion des événements liés à l'utilisation du clavier et de la souris

Clavier et souris

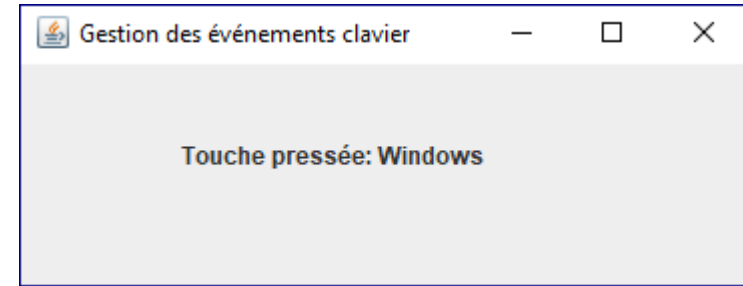
1. Gestion des événements clavier

Gestion événements clavier

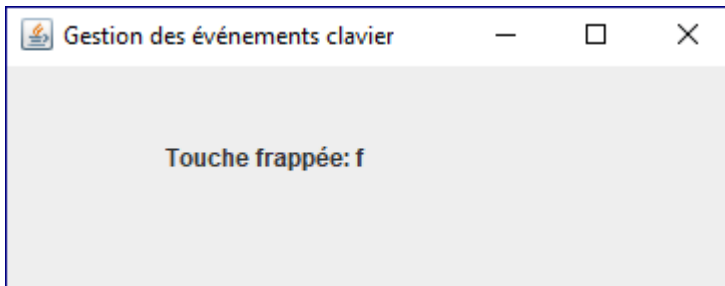
Au lancement de l'application



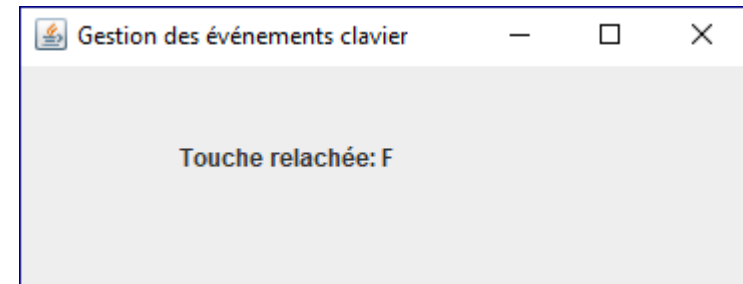
Si clic sur une touche action



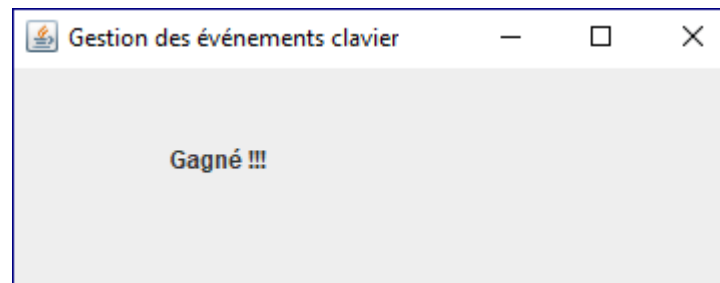
Si clic sur une touche autre que action



Si touche relâchée



Si clic sur la touche mystère à trouver (ici, la touche "a")



Gestion événements clavier

```
public class KeyboardListeningWindow extends JFrame {  
  
    private JLabel label;  
    ...  
  
    public KeyboardListeningWindow()  
    { ...  
  
        label = new JLabel("Pressez n'importe quelle touche");  
        ...  
  
        KeyboardListener listener = new KeyboardListener();  
  
        this.addKeyListener(listener);   —————>  On écoute la fenêtre !!!  
  
        ...  
    }  
}
```

Interface

```
private class KeyListener implements KeyListener {
```

```
    public void keyTyped (KeyEvent event) —————> Touche autre qu'une touche action
```

```
    { if (event.getKeyChar() == 'a') —————> Retourne la touche pressée
```

```
        label.setText ("Gagné !!!");
```

```
    else
```

```
        label.setText ("Touche frappée: "+event.getKeyChar());
```

```
    }
```

```
    public void keyPressed (KeyEvent event) —————> N'importe quelle touche pressée
```

```
    { label.setText ("Touche pressée: "+event.getKeyText(event.getKeyCode()));
```

```
    }
```

```
    public void keyReleased (KeyEvent event) —————> N'importe quelle touche relâchée
```

```
    { label.setText ("Touche relachée: "+event.getKeyText(event.getKeyCode()));
```

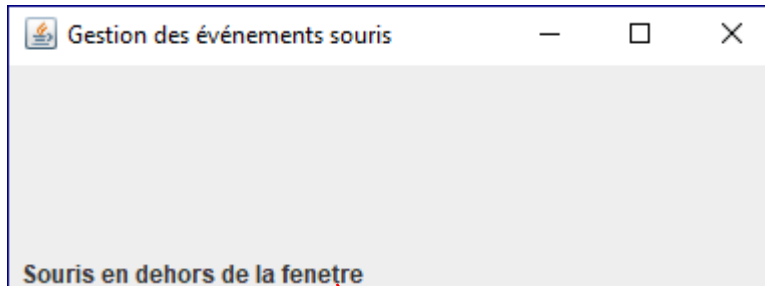
```
    }
```

```
}}
```

Clavier et souris

1. Gestion des événements clavier
2. Gestion des événements souris

Gestion événements clavier



Autres messages possibles dans la status bar :

Clic en [...,...] Exemple : Clic en [125,256]

Pression en [...,...]

Relâchement en [...,...]

Tirée en [...,...]

Déplacée en [...,...]

Gestion événements clavier

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


Gestion événements clavier

```
public class MouseListenerWindow extends JFrame {
```

```
    private JLabel statusBar;
```

```
    private Container frameContainer;
```

```
    public MouseListenerWindow()
```

```
    { ...
```

```
        statusBar = new JLabel();
```

```
        frameContainer = getContentPane();
```

```
        frameContainer.add(statusBar, BorderLayout.SOUTH);
```

```
        MouseEventListener listener = new MouseEventListener();
```

```
        frameContainer.addMouseListener(listener);
```

```
        frameContainer.addMouseMotionListener(listener);
```

```
        ...
```

```
    }
```

Deux types d'écouteurs d'évènements

⇒ deux interfaces :

MouseListener

MouseMotionListener

On écoute le container

private class **MouseEventListener** implements **MouseListener**, **MouseMotionListener**

```
{  public void mouseClicked( MouseEvent event)
    { statusBar.setText("Clic en ["+event.getX() + ","+event.getY() +"]"); }

  public void mousePressed( MouseEvent event)
    { statusBar.setText("Pression en ["+event.getX()+ ","+event.getY()+"]"); }

  public void mouseReleased( MouseEvent event)
    { statusBar.setText("Relachement en ["+event.getX()+ ","+event.getY()+"]"); }

  public void mouseExited( MouseEvent event)
    { statusBar.setText("Souris en dehors de la fenetre"); }

  public void mouseEntered( MouseEvent event)
    { statusBar.setText("Souris dans la fenetre"); }

  public void mouseDragged( MouseEvent event)
    { statusBar.setText("Tirée en ["+event.getX()+ ","+event.getY()+"]"); }

  public void mouseMoved( MouseEvent event)
    { statusBar.setText("Déplacée en ["+event.getX()+ ","+event.getY()+"]"); }

}}
```

Coordonnées x et y de la souris
au moment de l'évènement