

Les interfaces graphiques

- GUI : Graphic User Interface. Agencement d'un ensemble de composants généralement prédéfinis dans des bibliothèques standards.
- Programmation évènementielle.
- Gestion de la connection au système de fenêtrage : Gnome, KDE

Les différentes bibliothèques graphiques

- En C ou C++ : tk, Gtk, tkinter, OpenGL, qt, wx
- Exemple en qt :

```
#include <qapplication.h>

#include <qpushbutton.h>

int main(int argc, int **argv)

{

    QApplication a(argc, argv);

    QPushButton hello("Hello World !", 0);

    hello.resize(100,30);

    a.setMainWidget(&hello);

    hello.show();

    return a.exec();

}
```

- Ces bibliothèques sont interfacées avec un grand nombre d'autres langages : Python, Java, lisp
- Java propose également ses propres bibliothèques d'interfaces graphiques : awt, Swing (JFC), J3D

Éléments de base des interfaces graphiques

Les composants graphiques sont réunis sous le terme de *widgets*.

- La fenêtre principale et la connexion à l'environnement.
- Un container pour agencer les composants.
- Une barre de menu - composée de boutons.
- Un espace d'affichage : texte, dessin...
- Des fenêtres "*filles*" détachées.

Organisation

- La fenêtre principale
 - ▶ Le `display` - la fenêtre *racine*.
 - ▶ Le contexte graphique et la table des couleurs.
- L'agencement des composants
 - ▶ Emboitement de fenêtres : le type `Frame`, `Panel`, `XtMainWindow` `Canvas` ...
 - ▶ Gestion du placement : attachements aux bords de la fenêtre parente, positionnement relatif aux fenêtres *soeurs*.

Les bibliothèques JAVA

- AWT : Abstract Windowing Toolkit, package `java.awt`
historiquement les premières bibliothèques graphiques.
- JFC/Swing : nouveau paradigme java, package `javax.swing`
- java3D : avec le package `javax.media.j3d`, bibliothèques supplémentaires au jdk de base.

Introduction à AWT

- `java.awt.Graphics` : dessin de formes, lignes, images, sélection de fontes, couleurs
- `java.awt.Components` : boutons, zones de texte, menus, scrollbars
- Gestion des containers : `java.awt.*Layout` - si vous désirez organiser les différents composants.
- Gestion des évènements : `java.awt.event`
- Manipulation d'images : `java.awt.image`

Les composants

- Etendre Frame pour la fenêtre principale
- Utiliser l'interface ActionListener pour la gestion des évènements.

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

public class ButtonTest extends Frame implements ActionListener {

    Button OkButton;
    Label TheLabel;
    public ButtonTest () {
        setLayout(null);
        setBackground(Color.black);
        setForeground(Color.cyan);
        theLabel = new Label("Bonjour");
        theLabel.setSize(100,60);
        theLabel.setLocation(30,50);
        add(TheLabel);
        okButton = new Button("Ok");
        okButton.setSize(60,40);
        okButton.setLocation(50,120);
        okButton.addActionListener(this);
        add(OkButton);
    }
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        dispose();
        System.exit (0);
    }
}
```

Méthode main

```
public static void main(String[] args) {
    Frame f = new ButtonTest();
    f.setTitle("Button Test");
    f.setSize(160,200);
    f.setVisible(true);
    f.addWindowListener (new WindowAdapter () {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });
}
```


Web - Faire une applet

- Insérer une application java dans une page web.
- Nécessite que le navigateur ait le plug-in correspondant, sinon utiliser appletviewer.
- Ecrire une page HTML pour appeler le code.
- Ecrire le code *java* de l'applet et la compiler.

Exemple - Floride.java

```
import java.awt.Graphics;
import java.awt.Polygon;
public class Floride extends java.applet.Applet {
    public void paint(Graphics screen) {
        screen.drawString("Florida", 185, 75);
        screen.drawLine(185,80,222,80);
        screen.drawRect(2, 2, 345, 345);
        screen.drawRoundRect(182,61,43,24,10,8);
        int x[] = { 10, 234, 253, 261, 344, 336, 295, 259, 205, 211,
            195, 191, 120, 94, 81, 12, 10 };
        int y[] = { 12, 15, 25, 71, 209, 278, 310, 274, 188, 171, 174,
            118, 56, 68, 49, 37, 12 };
        int pts = x.length;
        Polygon poly = new Polygon(x, y, pts);
        screen.drawPolygon(poly);
        screen.fillOval(235,140,15,15);
        screen.fillOval(225,130,15,15);
        screen.fillOval(245,130,15,15);
        for (int ax = 50; ax < 150; ax += 10)
            for (int ay = 120; ay < 320 ; ay += 10)
                screen.drawArc(ax, ay, 10, 10, 0, -180);
    }
}
```

Exemple - Floride.html

```
<HTML>
<HEAD>
<TITLE> Exemple d'applet Java </TITLE>
</HEAD>

<BODY> Voici une applet Java : <br>
<APPLET CODE="Floride.class" WIDTH=400 HEIGHT=200>
</APPLET>
</BODY>

</HTML>
```

Programmation SWING

- La fenêtre la plus basique : `JWindow`, utilisée seulement pour les fenêtres de type *alerte*.
- La fenêtre principale : `JFrame` - possède par défaut un bouton de fermeture, un bouton de redimensionnement et un bouton pour iconifier.
- La fenêtre *popup* : `JDialog` - peut bloquer une autre fenêtre tant qu'elle est ouverte (*modale*).
- Gestion des évènements et *threads* - méthode `run` de l'interface `Runnable`.