

# Interface graphique en JAVA

## Développement d'interfaces graphiques en SWING :

### 1. Qu'est-ce que SWING ?

Swing est une bibliothèque graphique sur Java et fait partie du package JFC ( java foundation classes).

### 2. Pourquoi Swing ?

Elle nous offre la possibilité de créer des interfaces graphiques quel que soit le système d'exploitation et dispose de plusieurs choix d'apparence ( idéal pour pouvoir implanter notre propre image du jeu et les différents composants graphiques là où on souhaite) .

En plus, elle constitue l'une des principales évolutions apportées par Java 2 par rapport aux versions antérieures.

De nombreux exemples sont disponibles pour pouvoir maîtriser cette bibliothèque simple d'utilisation.

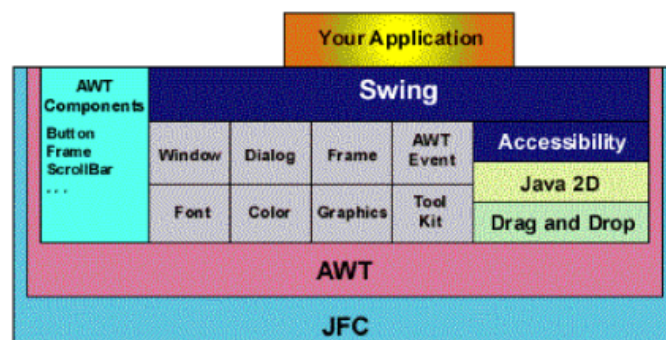
**Playlist Youtube pour le développement d'un jeu Mario sur Eclipse :**

[https://www.youtube.com/watch?v=MwzaL6WwgV8&list=PLDKRpuLSe0f\\_4YgSROSWf\\_3wsdXwqefw9&index=1](https://www.youtube.com/watch?v=MwzaL6WwgV8&list=PLDKRpuLSe0f_4YgSROSWf_3wsdXwqefw9&index=1)

## Toolkits graphiques Java

### Pour le Desktop

- **JavaFX**  
nouveau, inspiré du Web
- **Swing**
- **AWT Components**  
obsolète
- **SWT**  
Eclipse Foundation



### Swing

- est multi-plateformes
- repose sur **AWT Components** mais **pas la même chose !**
  - attention : **JButton** != **Button** !

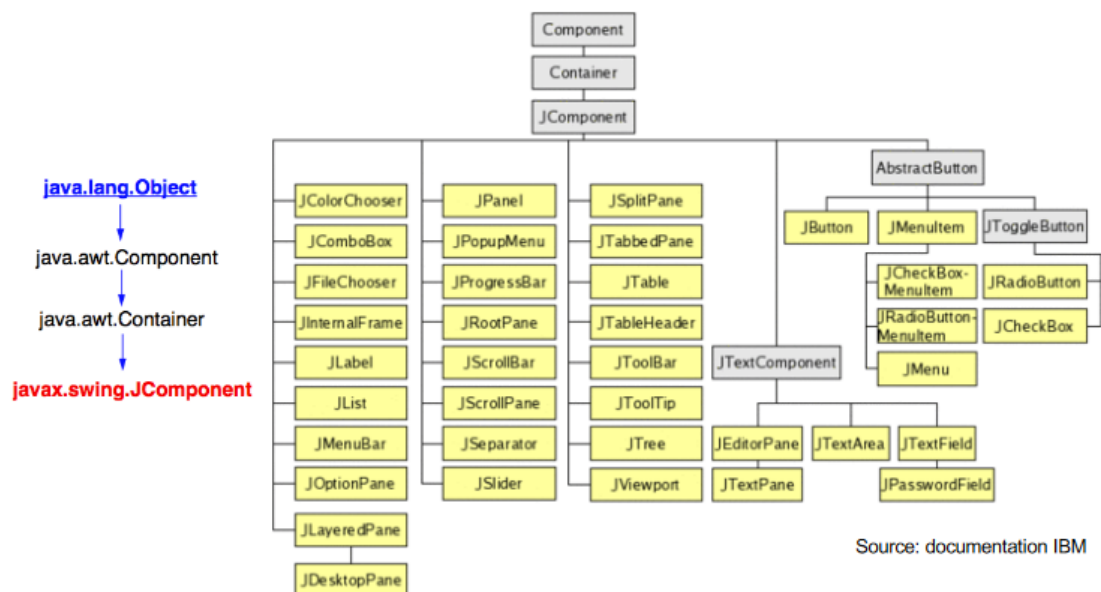
### 3. Packages :

Swing contient plusieurs packages :

javax.swing	package principal : il contient les interfaces, les principaux composants, les modèles par défaut
javax.swing.border	Classes représentant les bordures
javax.swing.colorchooser	Classes définissant un composant pour la sélection de couleurs
javax.swing.event	Classes et interfaces pour les événements spécifiques à Swing. Les autres événements sont ceux d'AWT (java.awt.event)
javax.swing.filechooser	Classes définissant un composant pour la sélection de fichiers
javax.swing.plaf	Classes et interfaces génériques pour gérer l'apparence
javax.swing.plaf.basic	Classes et interfaces de base pour gérer l'apparence
javax.swing.plaf.metal	Classes et interfaces pour définir l'apparence Metal qui est l'apparence par défaut
javax.swing.table	Classes définissant un composant pour la présentation de données sous forme de tableau
javax.swing.text	Classes et interfaces de bases pour les composants manipulant du texte
javax.swing.text.html	Classes permettant le support du format HTML
javax.swing.text.html.parser	Classes permettant d'analyser des données au format HTML
javax.swing.text.rtf	Classes permettant le support du format RTF
javax.swing.tree	Classes définissant un composant pour la présentation de données sous forme d'arbre
javax.swing.undo	Classes permettant d'implémenter les fonctions annuler/refaire

### 4. Les composants Swing:

## Composants Swing

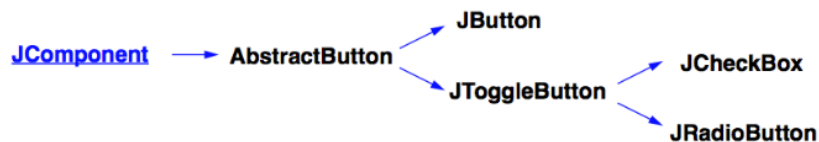


Toutes les classes et composants Swing sont détaillés sur ce site ( <https://www.jmdoudoux.fr/java/dej/chap-swing.htm#swing-1> ) où on explique en détails le fonctionnement des fonctions/méthodes et la personnalisation des composants de la bibliothèque SWING comme par exemple :

- Dimensionner la fenêtre/ la centrer sur l'écran.
- Ajout de boutons/titres.
- Contrôler l'action de l'appui sur un bouton.
- Les cases à cocher ( notamment pour le choix du niveau sur le jeu )

-Exemples pour les boutons :  
On a différents types de boutons ....

## Boutons



JButton



JCheckbox :  
choix indépendants

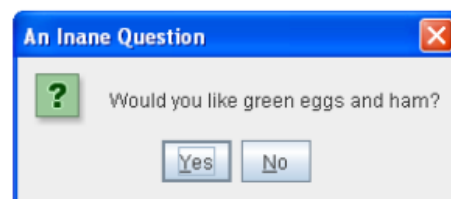
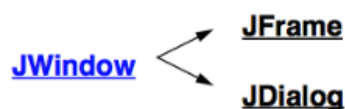


JRadioButton :  
choix exclusif : cf. **ButtonGroup**

Source: documentation Java Oracle

-Exemples pour les fenêtres si on en a besoin :

## Fenêtres



**JFrame** : fenêtre principale de l'application

**JDialog** : fenêtre secondaire

- normalement **dépendante** de la JFrame :
  - pas d'icônes séparées, toujours au dessus de la JFrame
- souvent **temporaire** et **modale** :

- Événements Java :

## Événements Java

### Événements AWT et Swing

- **objets** correspondant à des catégories d'événements
- les principaux héritent de **java.awt.event.AWTEvent**

### Événements de “bas niveau”

- **MouseEvent** appuyer, relacher, bouger la souris ...
  - **KeyEvent** appuyer, relacher une touche clavier...
  - **WindowEvent** fermeture des fenêtres ....
  - **FocusEvent** focus clavier (= où vont les caractères tapés au clavier)
- etc.

### Événements de “haut niveau”

- **ActionEvent** **activer** un bouton, un champ textuel ...  
**abstraction** des événements de bas niveau
  - **TextEvent** modification du texte entré
- etc.

Pour le déplacement des personnages à l'aide des touches du clavier, on pourra utiliser le “FocusEvent” qui sert à reconnaître les caractères tapés au clavier...

Liens externes utilisés :

-Playlist jeu Mario 2D :

[https://www.youtube.com/watch?v=MwzaL6WwgV8&list=PLDKRpuLSe0f\\_4YgSROSWf\\_3wsdXwqefw9&index=1](https://www.youtube.com/watch?v=MwzaL6WwgV8&list=PLDKRpuLSe0f_4YgSROSWf_3wsdXwqefw9&index=1)

-Classes et composants Swing:

<https://www.jmdoudoux.fr/java/dej/chap-swing.htm#swing-1>

-Cours sur SWING :

[https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiSsoTUiPXzAhUPz4UKHbhsCbAQFnoECAIQAQ&url=https%3A%2F%2Fperso.telecom-paristech.fr%2Ffelc%2Fswing%2FJavaSwing.pdf&usg=AOvVaw3e67OvwP\\_Xx0sP0e1vW9eZ](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiSsoTUiPXzAhUPz4UKHbhsCbAQFnoECAIQAQ&url=https%3A%2F%2Fperso.telecom-paristech.fr%2Ffelc%2Fswing%2FJavaSwing.pdf&usg=AOvVaw3e67OvwP_Xx0sP0e1vW9eZ)