Cours R2.02
Introduction à l'Interaction Humain-Machine

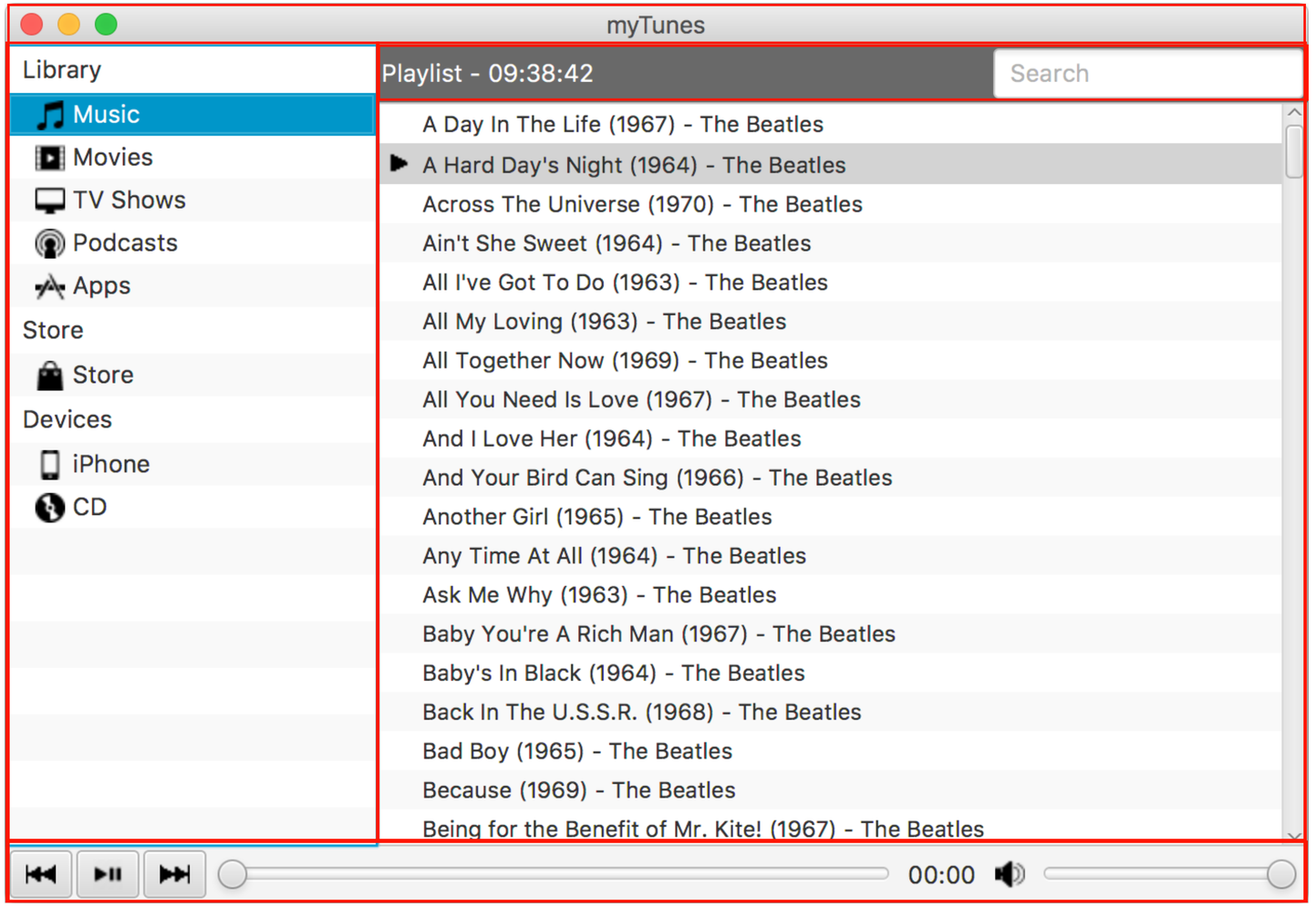
Cours 1 : gestionnaires de placement

Géry Casiez https://gery.casiez.net
Département informatique - IUT de Lille - Université de Lille

Plan du cours en 9 semaines

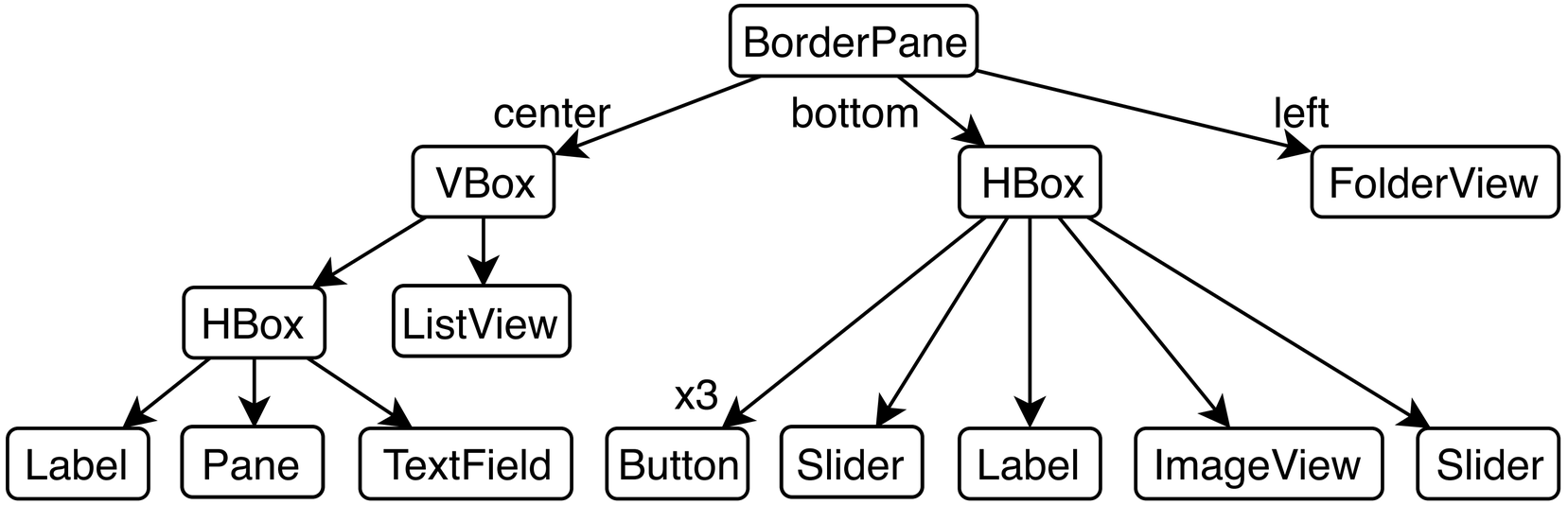
1. **Introduction à l'interaction, placement**
2. Programmation événementielle
3. Widgets et événements (1/2)
4. Widgets et événements (2/2)
5. Conception et prototypage (1/2)
6. Conception et prototypage (2/2)
7. Heuristiques et recommandations
8. Modèles et théories
9. Méthodes d'évaluation des IHM

Principe général

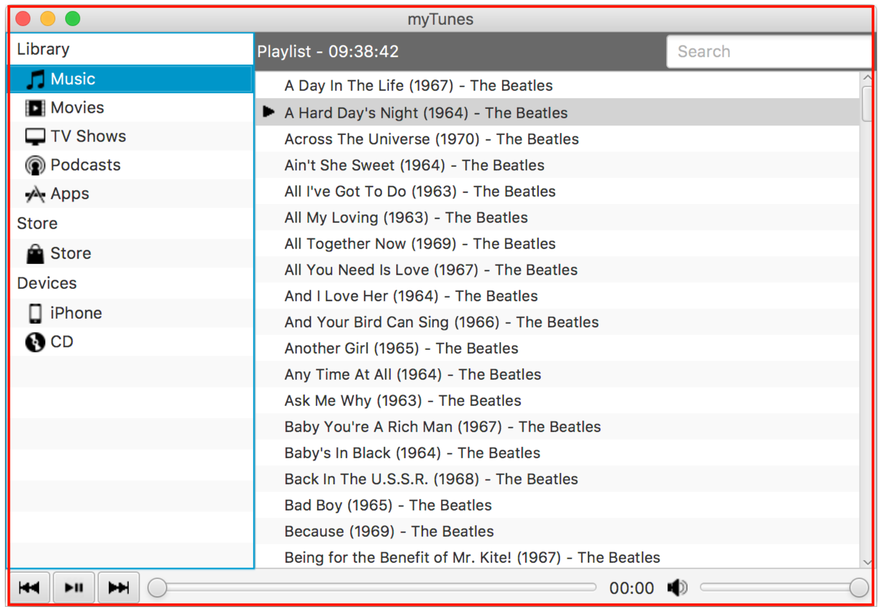


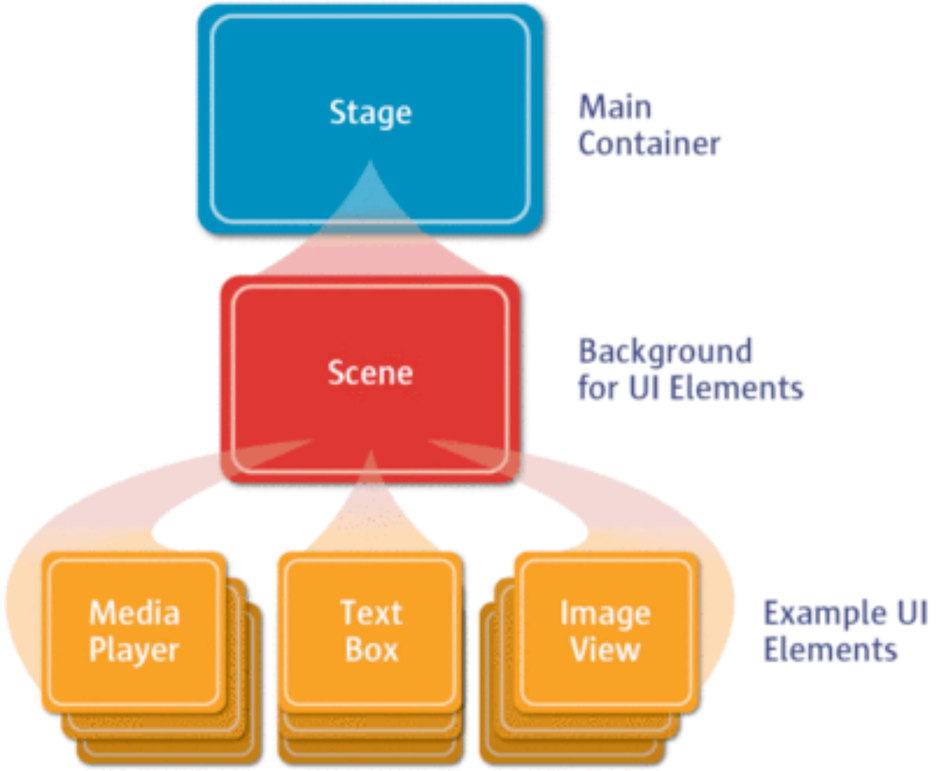
Widgets et conteneurs

Les widgets doivent être placés dans des conteneurs.

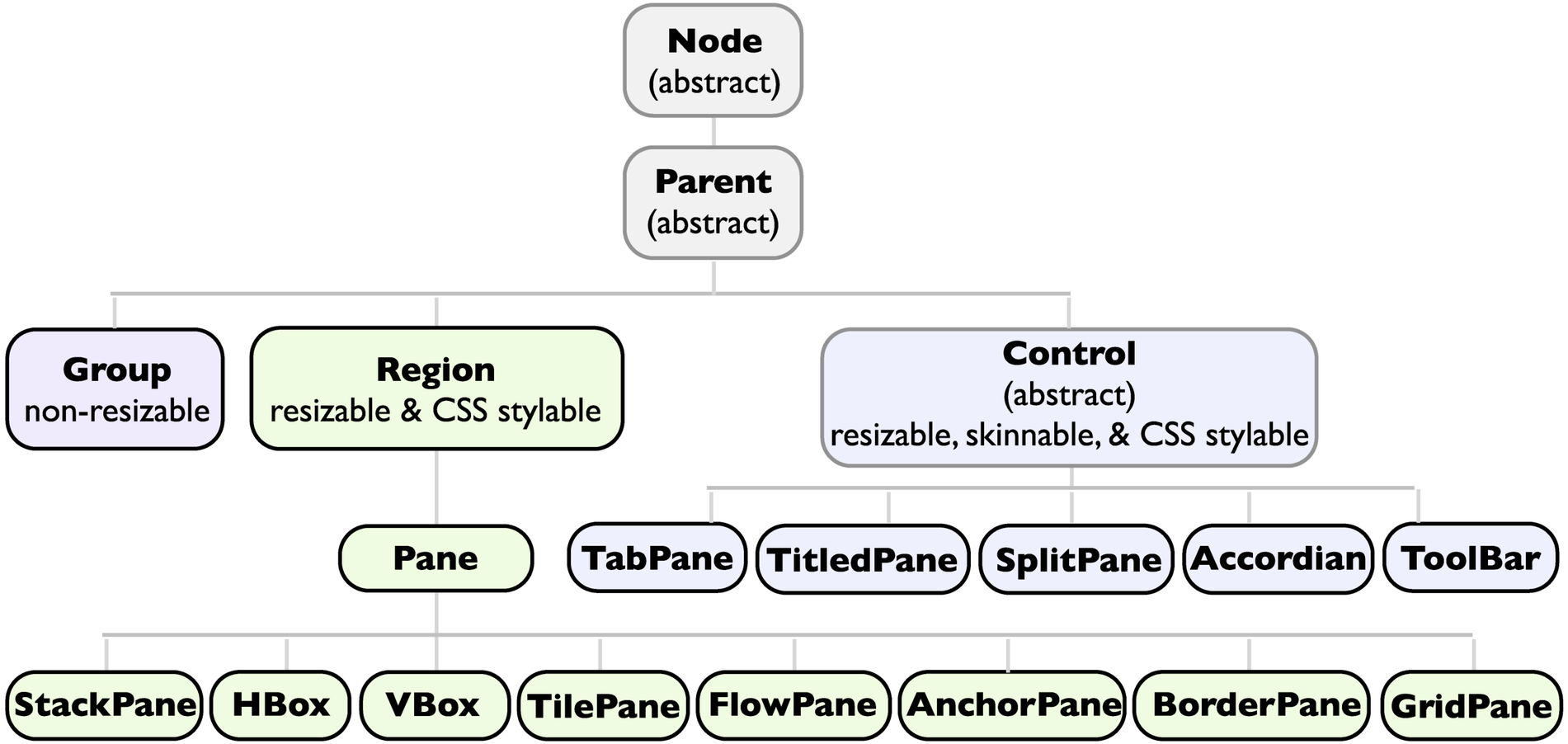


Stage et Scene

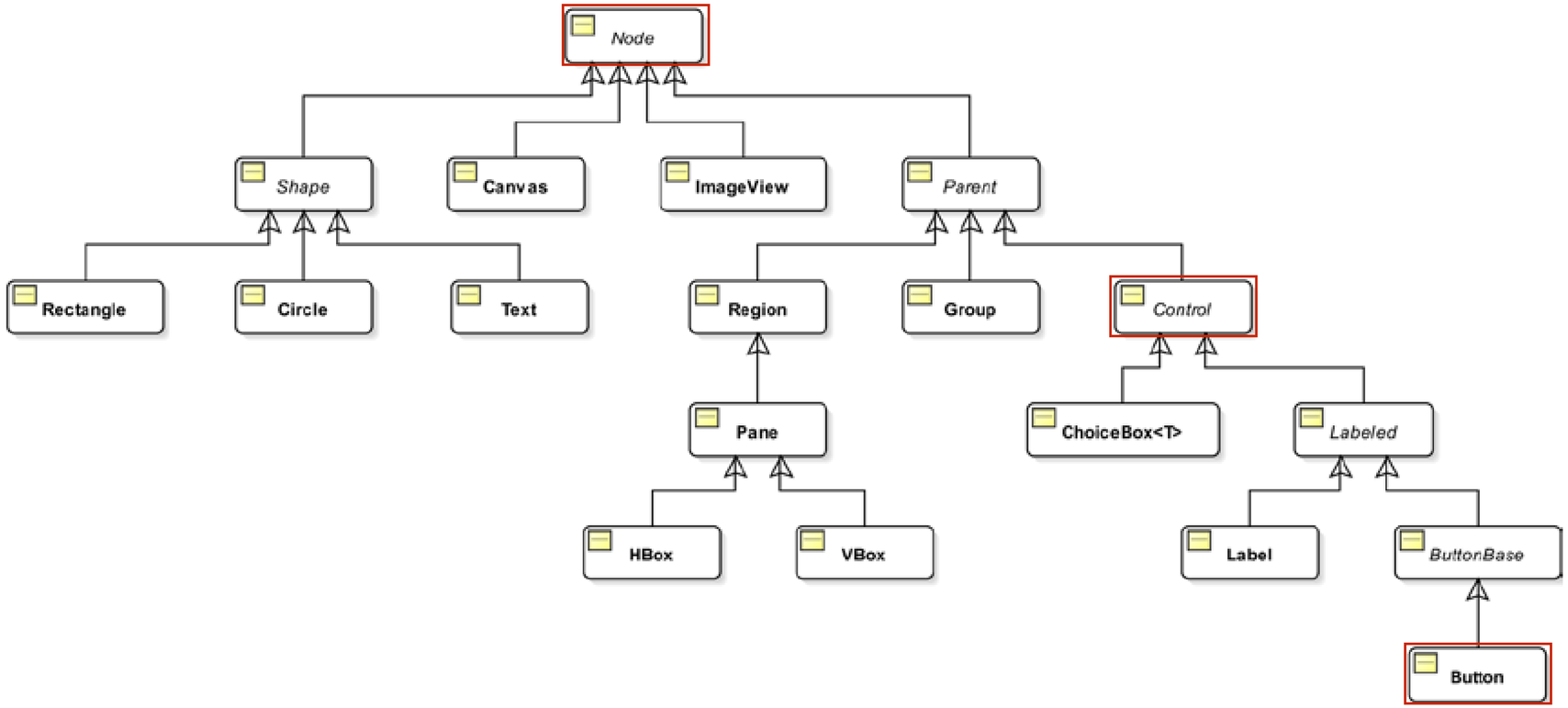




Conteneurs (Pane) et Node



Control et Node



Pane

Pane est un noeud (Node) particulier dont le rôle est de contenir d'autres noeuds stockés sous la forme d'une *ObervableList<Node>*

Contient le nombre et la liste des noeuds

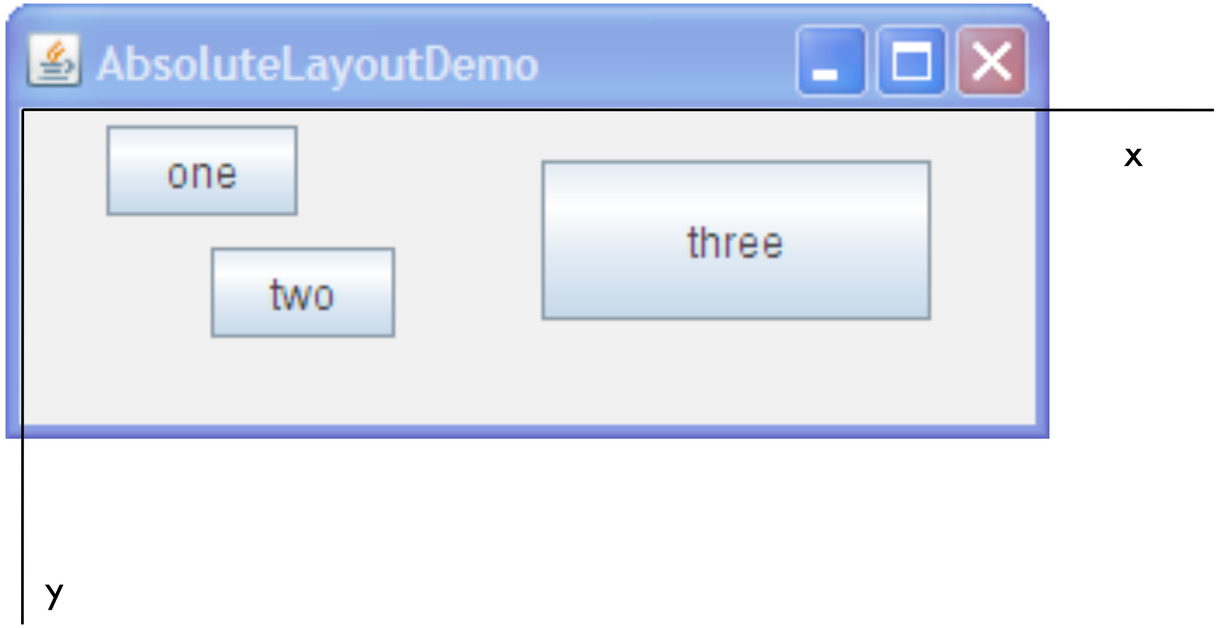
Parmi l'ensemble des noeuds que contient un Pane peut se trouver une instance de Pane

On crée ainsi une hiérarchie de noeuds qui est appelée graphe de scène

Positionnement absolu

Définition de la taille et de la position de chaque composant

L'origine est le coin supérieur gauche du conteneur



Gestionnaires de placement (layout manager)

Gestionnaires de placement : placement dynamique des enfants d'un nœud du graphe de scène

- Calcul de position et des dimensions de chaque Node
- Généralement, imbrication géométrique d'un widget dans son parent

Contraintes

- Taille « naturelle » de chaque fils 

- Taille imposée par le parent
- Contraintes de placement spécifiées par le programmeur

Stratégies de placement

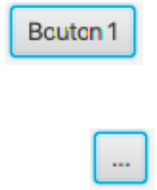
Comment un conteneur agence-t-il visuellement les widgets qu'il contient ?

Que se passe-t-il quand on agrandit la fenêtre ? Les widgets doivent-ils s'agrandir ? Ajoute-t-on de l'espace ? Où ?

Minimum, maximum, preferred sizes

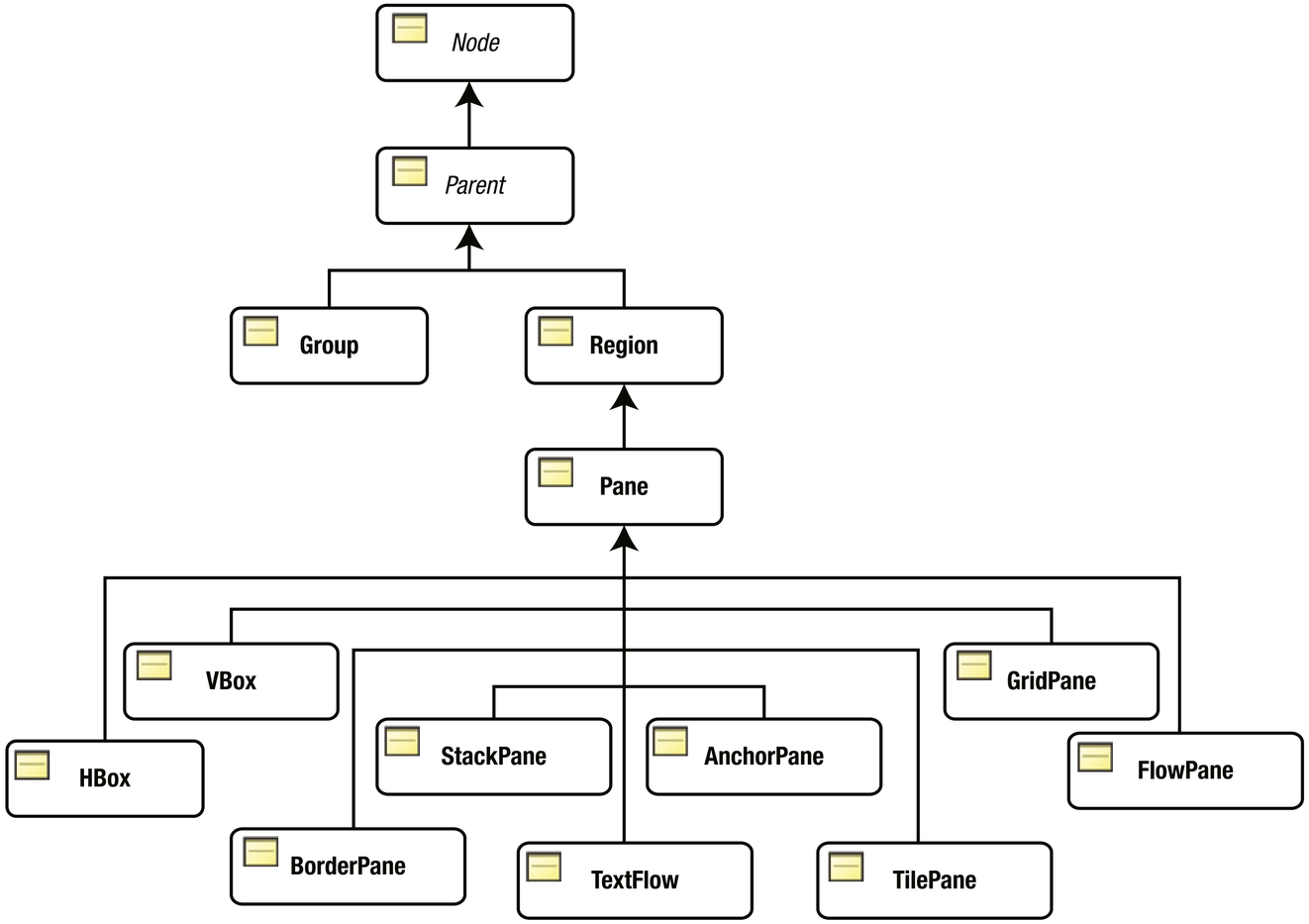
Chaque noeud a quatre tailles :

- La taille idéale (preferred size)
- La taille minimale (minimum size)
- La taille maximale (maximum size)
- La taille réelle



Un layout va essayer de rendre la taille réelle la plus proche possible de la taille préférée compte tenu des contraintes dues à sa stratégie et dues à la taille réelle du conteneur

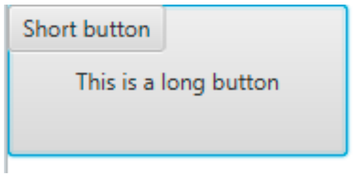
Layout Pane Classes



Group

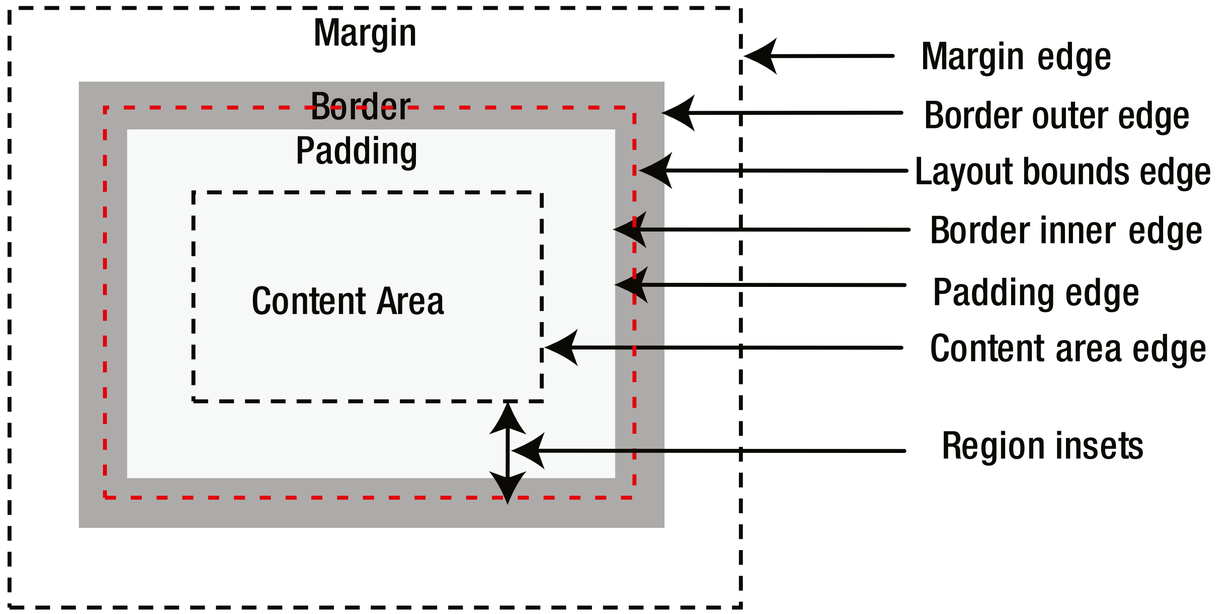
Pas de positionnement des noeuds réalisé (tous positionnés en (0,0) par défaut)

Tous les effets, transformations et changements de propriétés sont appliqués à tous les noeuds du groupe



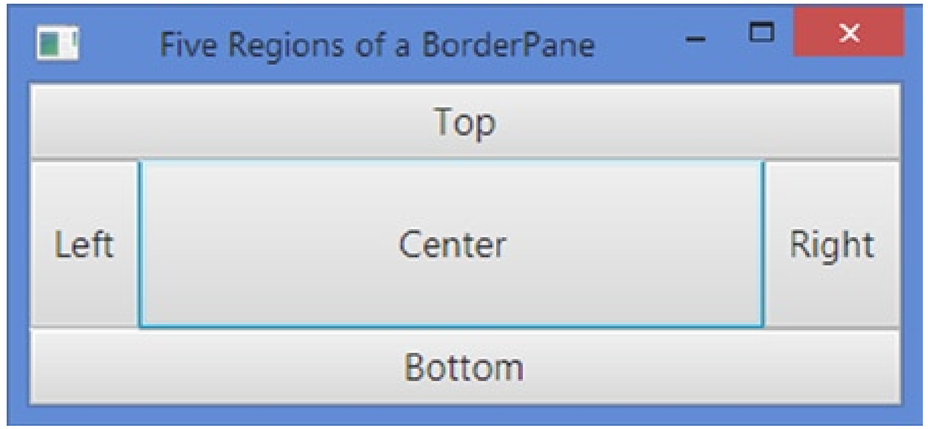
Region

Classe dont hérite les gestionnaires de placement



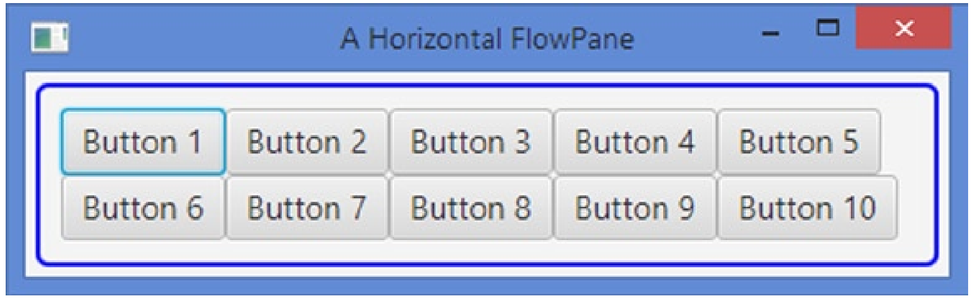
BorderPane

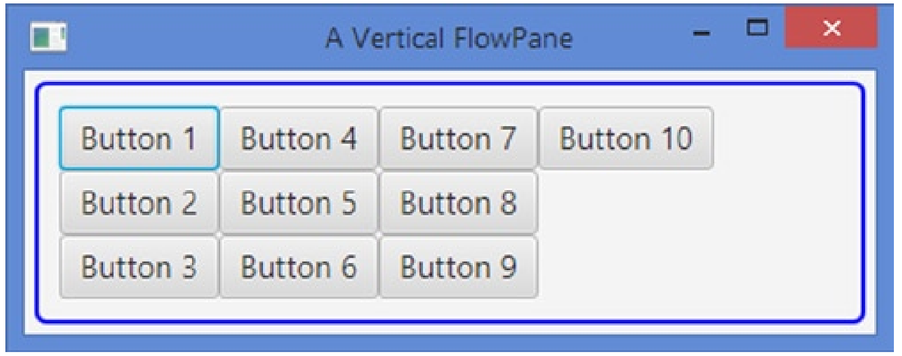
Satisfait les exigences de la plupart des applications de bureau.



FlowPane

Place les noeuds les uns à côté des autres ou les uns en dessous des autres

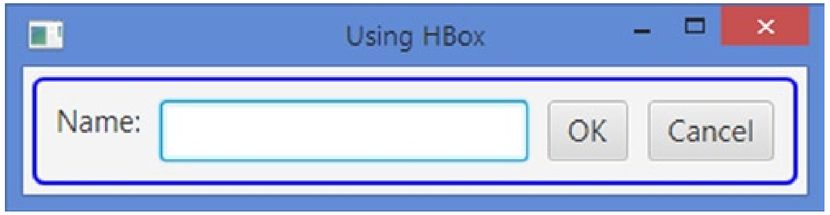




HBox

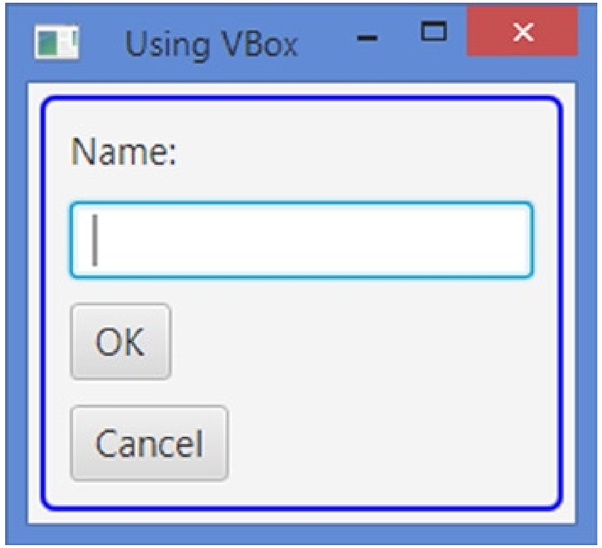
Place les noeuds sur une ligne horizontale

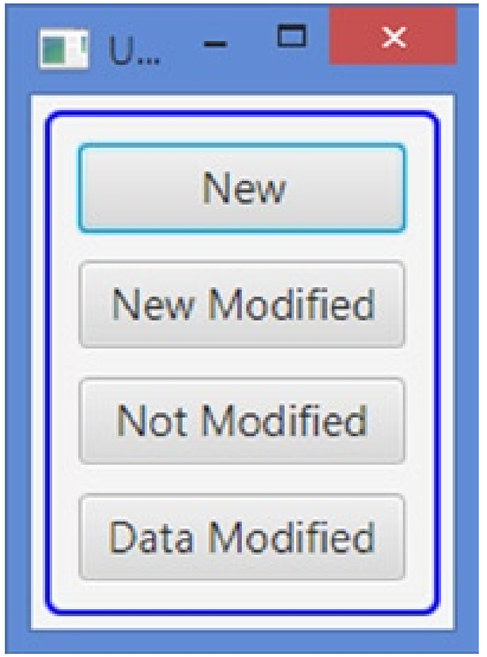
Possibilité de contrôler priorité d'agrandissement de certains noeuds



VBox

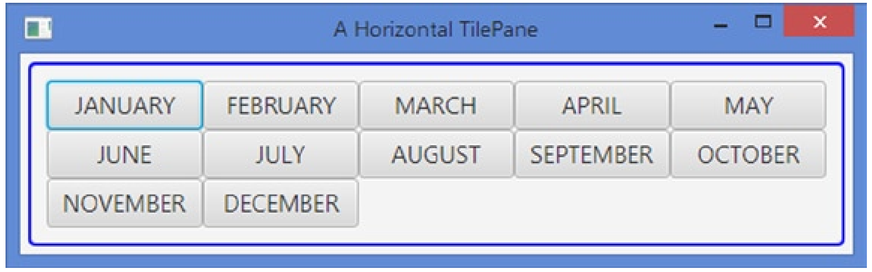
Place ses noeuds suivant une seule colonne

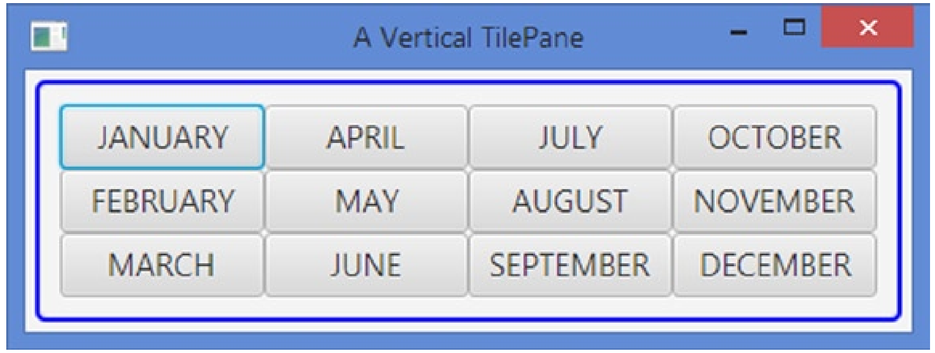




TilePane

Disposition des noeuds selon une grille dont toutes les cellules ont la même taille

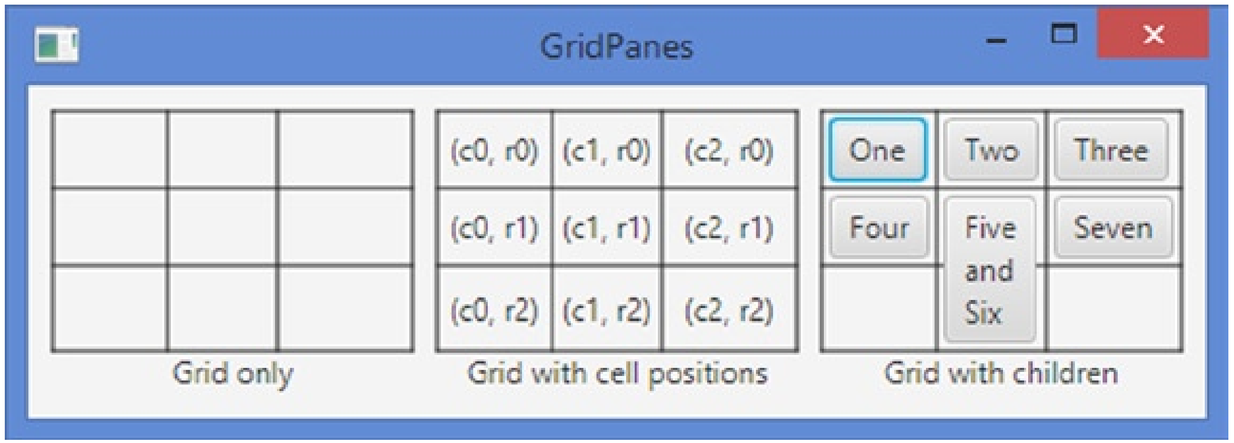




GridPane

Utilisation d’une grille

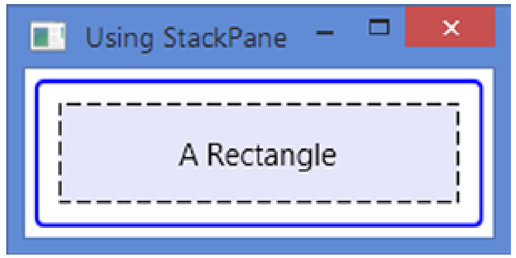
Un noeud peut occuper plusieurs cellules



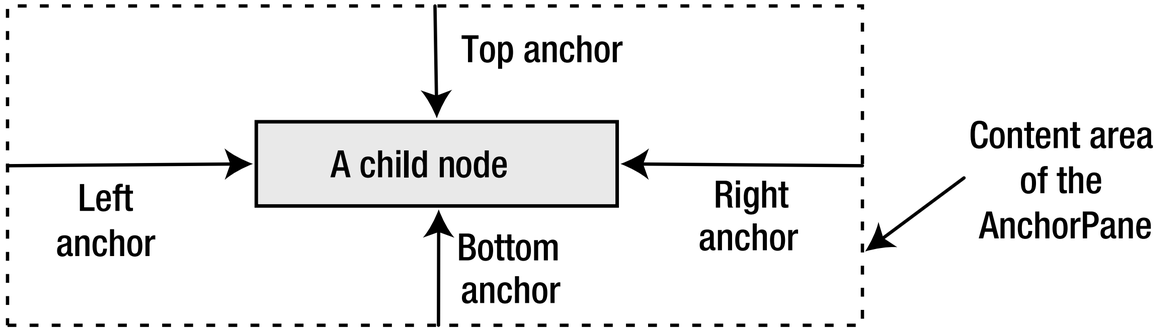
StackPane

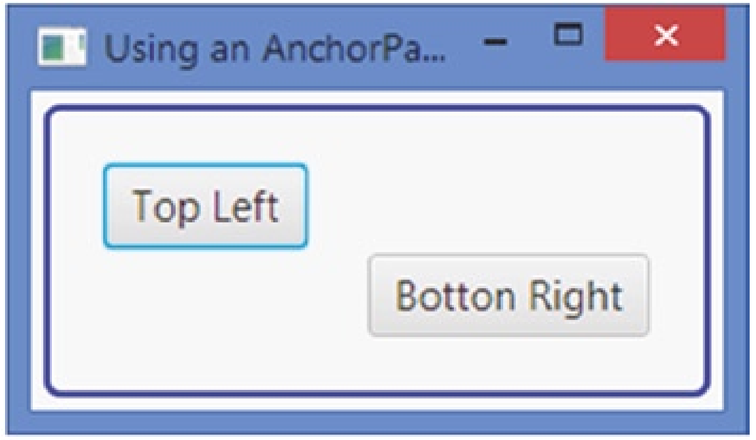
Empile les noeuds les uns au dessus des autres

Utile pour créer des effets visuels

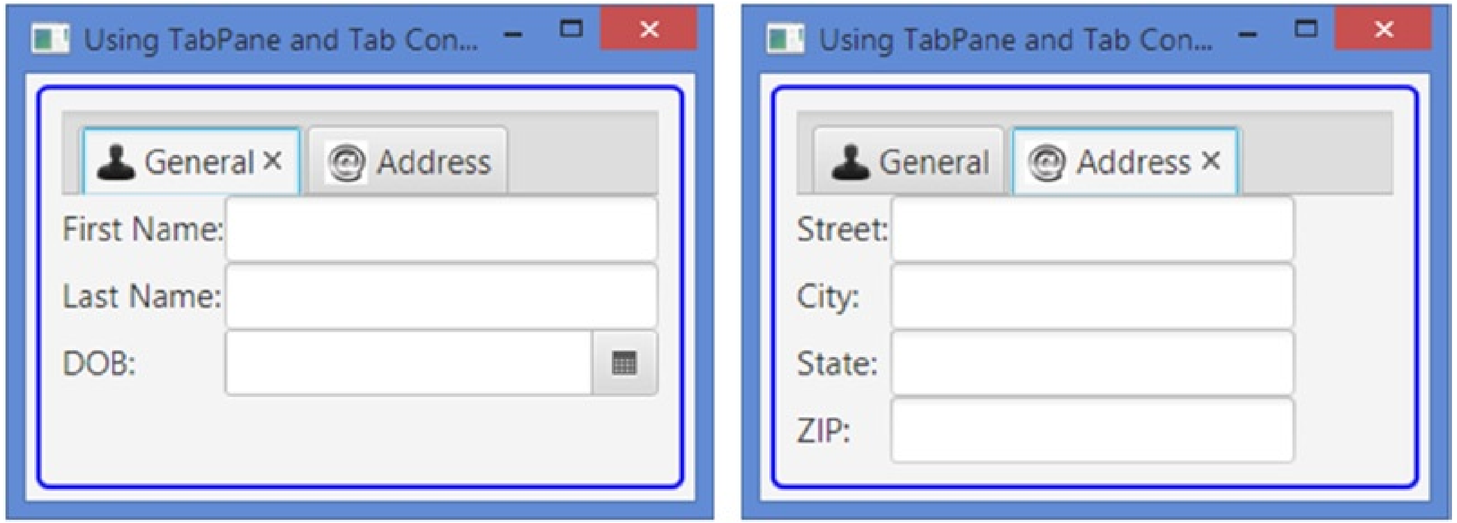


AnchorPane

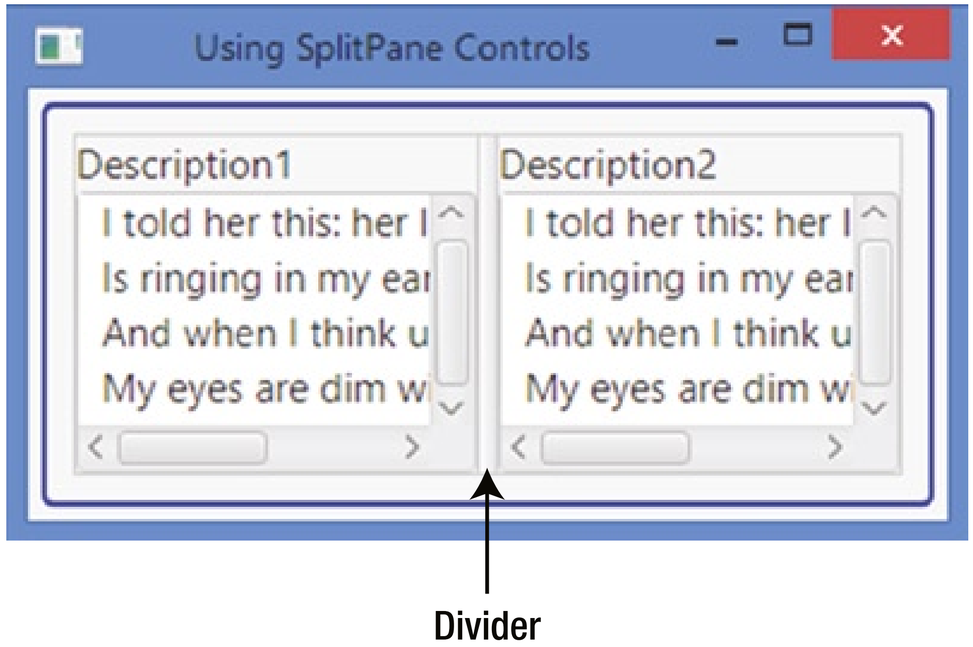

Top anchor
A child node
Left
anchor
Right
anchor
Bottom
anchor
Content area
of the
AnchorPane


Using an AnchorPa...
Top Left
Botton Right

TabPane


Using TabPane and Tab Con...
General
Address
First Name:
Last Name:
DOB:
Using TabPane and Tab Con...
General
Address
Street:
City:
State:
ZIP:

SplitPane


Using SplitPane Controls
Description1
I told her this: her l
ls ringing in my ear
And when I think u
My eyes are dim wi
Description2
I told her this: her l
ls ringing in my ear
And when I think u
My eyes are dim wi
Divider

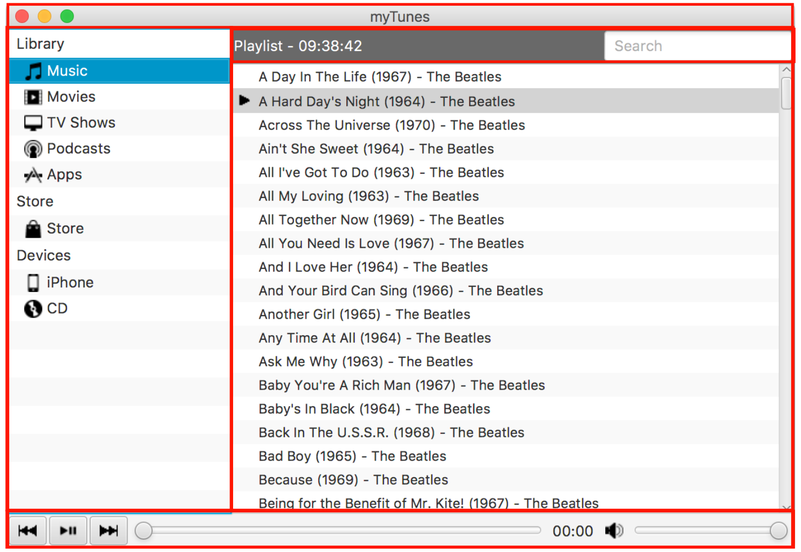
Toolbar

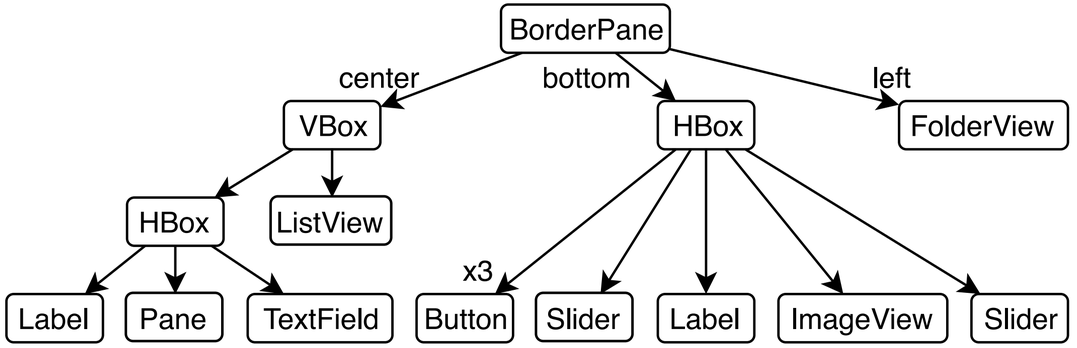
A toolbar with no overflow

A toolbar with an overflow

An overflow button

Exemple


myTunes
Library
Music
Movies
TV Shows
Podcasts
Apps
Store
Store
Devices
iPhone
CD
Playlist - 09:38:42
Search
A Day In The Life (1967) - The Beatles
A Hard Day's Night (1964) - The Beatles
Across The Universe (1970) - The Beatles
Ain't She Sweet (1964) - The Beatles
All I've Got To Do (1963) - The Beatles
All My Loving (1963) - The Beatles
All Together Now (1969) - The Beatles
All You Need Is Love (1967) - The Beatles
And I Love Her (1964) - The Beatles
And Your Bird Can Sing (1966) - The Beatles
Another Girl (1965) - The Beatles
Any Time At All (1964) - The Beatles
Ask Me Why (1963) - The Beatles
Baby You're A Rich Man (1967) - The Beatles
Baby's In Black (1964) - The Beatles
Back In The U.S.S.R. (1968) - The Beatles
Bad Boy (1965) - The Beatles
Because (1969) - The Beatles
Being for the Benefit of Mr. Kite! (1967) - The Beatles
00:00


BorderPane
center
bottom
left
VBox
HBox
FolderView
HBox
ListView
x3
Label
Pane
TextField
Button
Slider
Label
ImageView
Slider

Ajout de noeuds au conteneur

Méthode getChildren() du conteneur

Puis utilisations des méthodes:

add(Node e)

addAll(Node… Elements)

Java varargs …

Syntaxe qui permet de définir un nombre variable d'arguments **de même type**

Sans varargs :

```
public String myMethod()
public String myMethod(String value)
public String myMethod(String val1, String val2)
```

ou

```
public String myMethod(String[] val)
```

avec

```
public String myMethod(String… val)
```

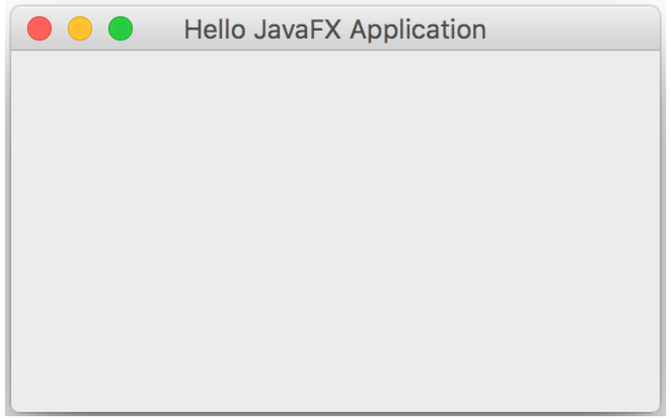
Java varargs …

Règles :

- une méthode ne peut avoir qu'un seul paramètre varargs
- le paramètre varargs doit être le dernier

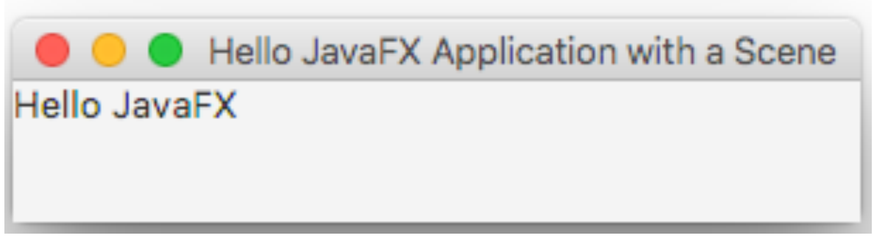
JavaFX Hello World


```
import javafx.application.Application;
import javafx.stage.Stage;

public class HelloJavaFX extends Application {

    public void start(Stage stage) {
        stage.setTitle("Hello JavaFX Application");
        stage.show();
    }

    public static void main(String[] args) {
        Application.launch(args);
    }
}
```



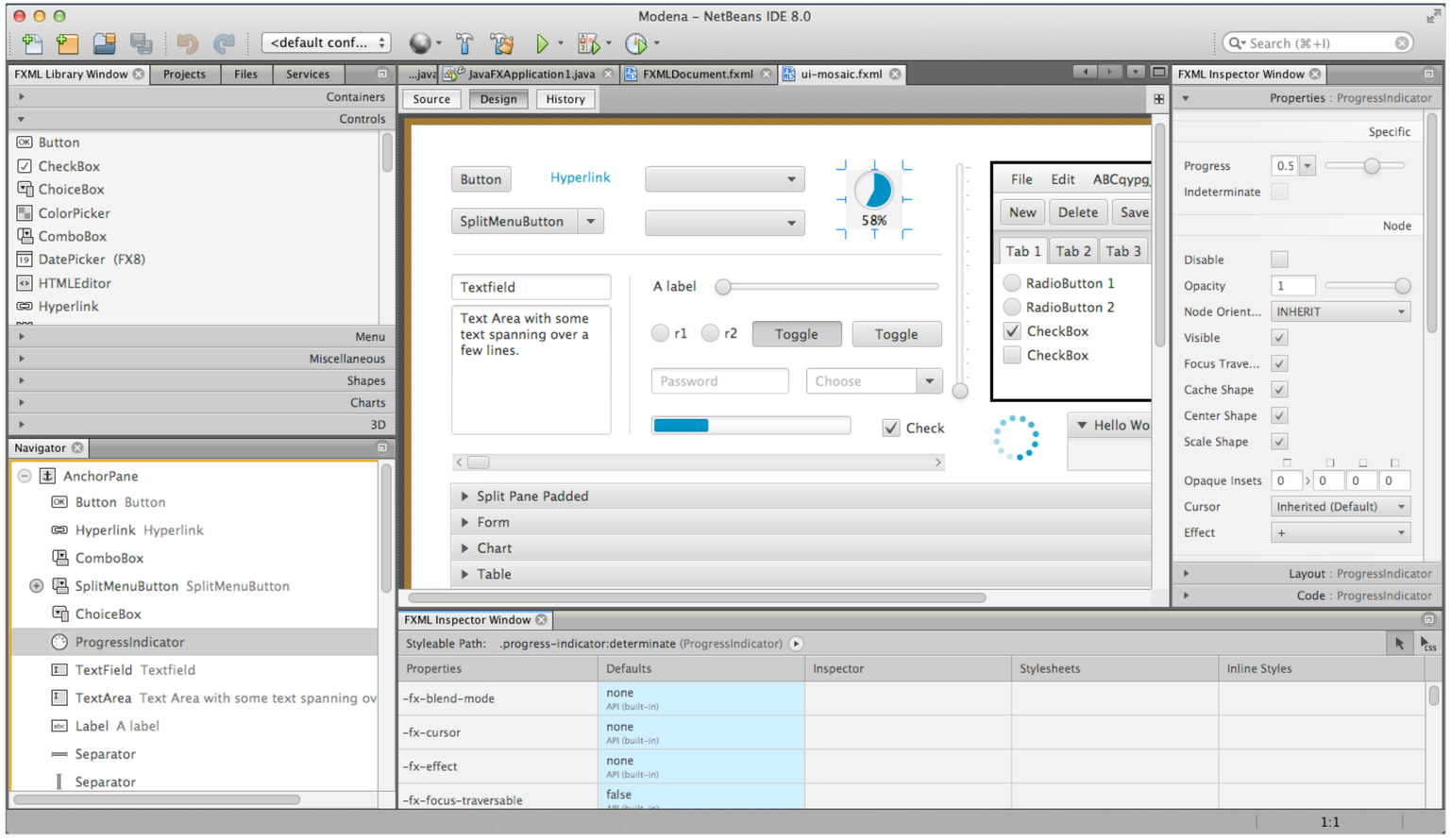
Exemple

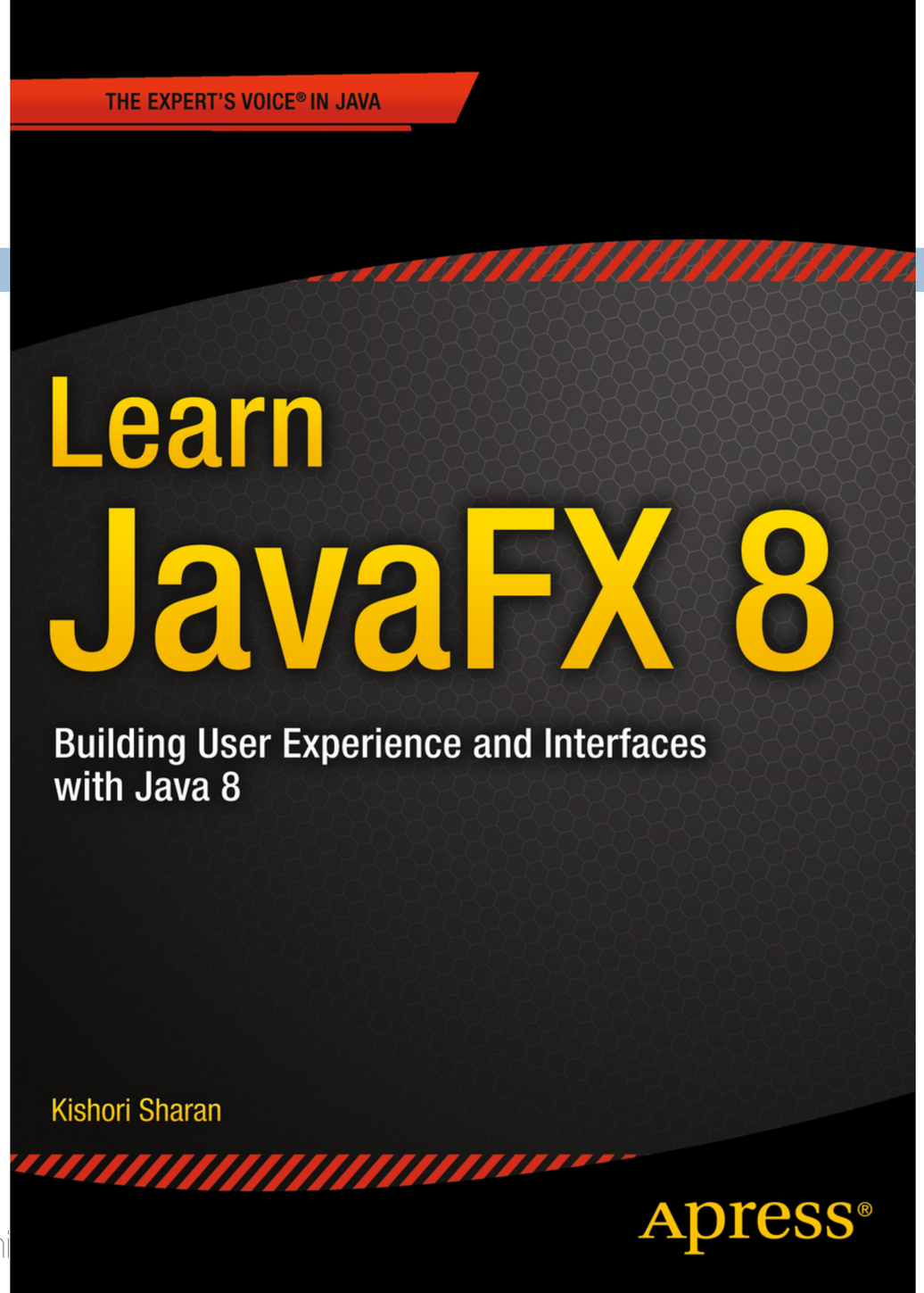

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class SimpleScene extends Application {

    public void start(Stage stage) {
        VBox root = new VBox();
        Label msg = new Label("Hello JavaFX");
        root.getChildren().add(msg);

        Scene scene = new Scene(root, 300, 50);
        stage.setScene(scene);
        stage.setTitle("Hello JavaFX Application with a Scene");
        stage.show();
    }

    public static void main(String[] args) {
        Application.launch(args);
    }
}
```



Générateurs d'interfaces

JavaFX Scene Builder




THE EXPERT'S VOICE® IN JAVA
Learn
JavaFX 8
Building User Experience and Interfaces with Java 8
Kishori Sharan
Apress®

A retenir

Utilisation de conteneurs pour regrouper des Nodes

Fenêtre = Stage

Différentes stratégies de positionnement