TD Introduction aux Systèmes Interactifs

https://www.lri.fr/~prouzeau/ISI/Sujets/index.html

Arnaud Prouzeau
prouzeau@lri.fr

Certains slides de cette présentation viennent d’anciennes présentations de Cédric Fleury et David Bonnet.

Introduction

Programmation de systèmes interactifs

En cours :

- Interface
- Interaction
- Visualisation
- …

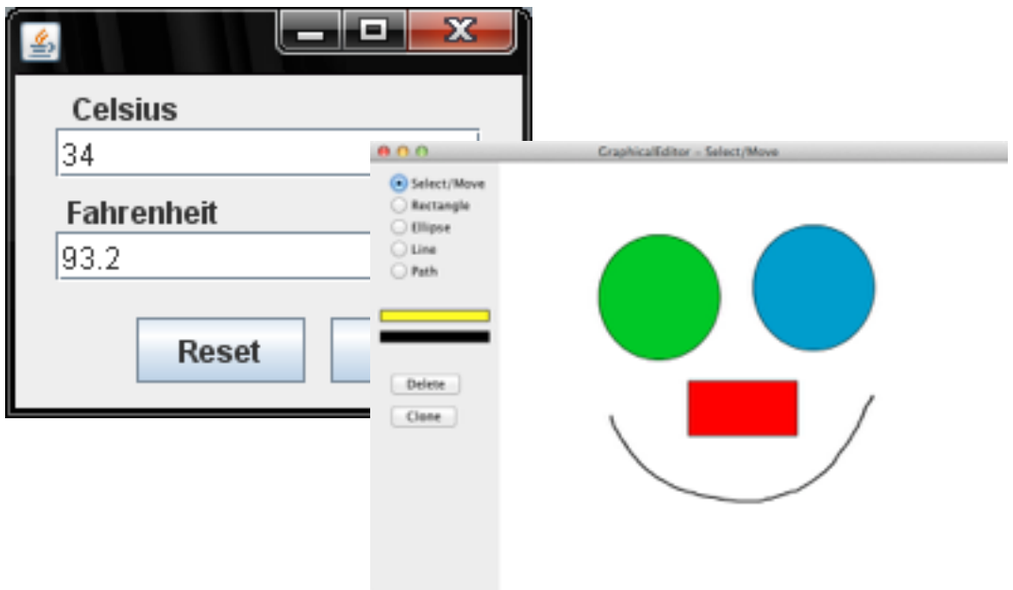
Programmation de systèmes interactifs

En cours :

- Interface
- Interaction
- Visualisation
- ...

En TD (Java) :

- Layout
- Evénement
- Fenêtre

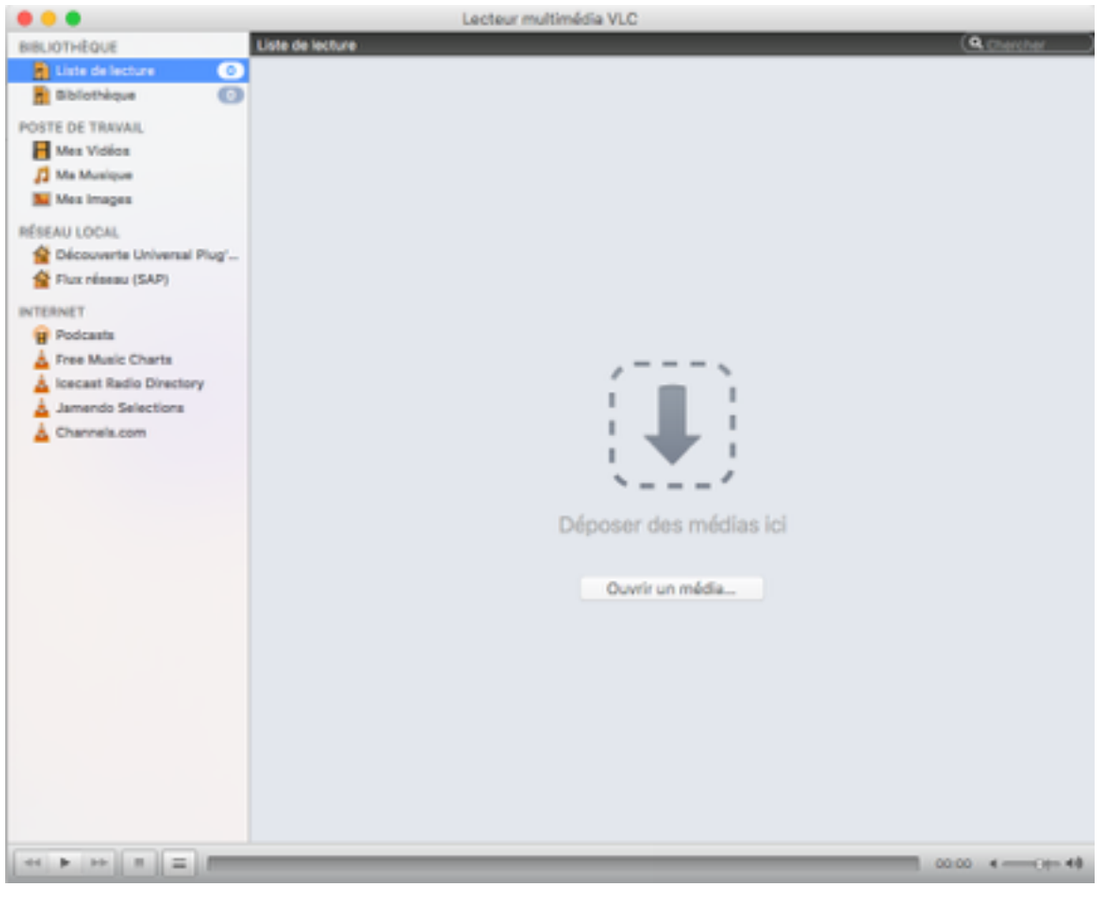


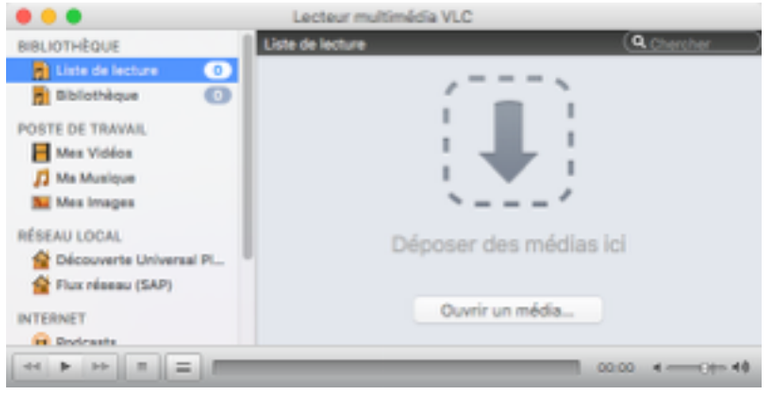
LayoutManager

- Positionnement des elements graphiques
- Robuste au redimensionnement

LayoutManager

- Positionnement des elements graphiques
- Robuste au redimensionnement





Construire son interface

Etape 1 : Identifier le/les LayoutManager à utiliser.

Construire son interface

Etape 1 : Identifier le/les LayoutManager à utiliser.

Etape 2 : Construire l’interface.

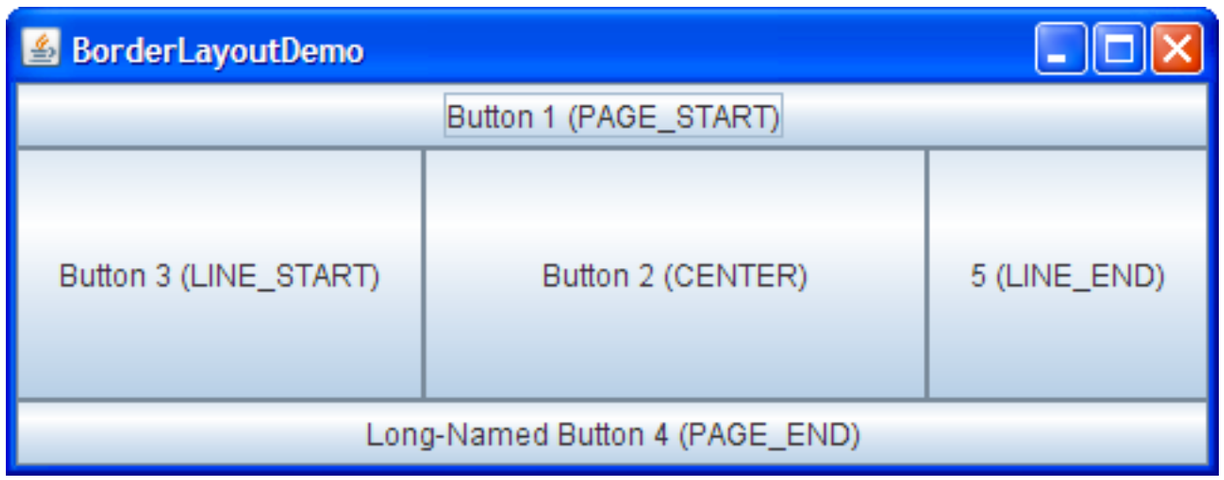
Construire son interface

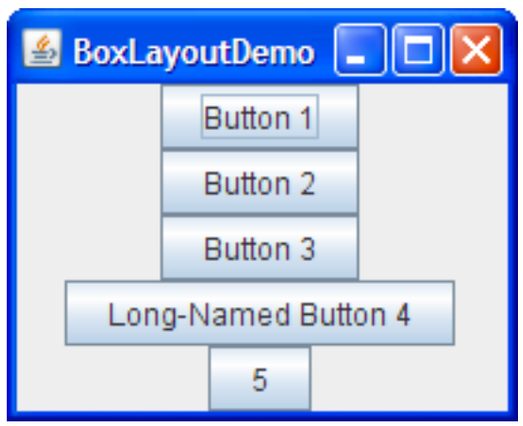
Etape 1 : Identifier le/les LayoutManager à utiliser.

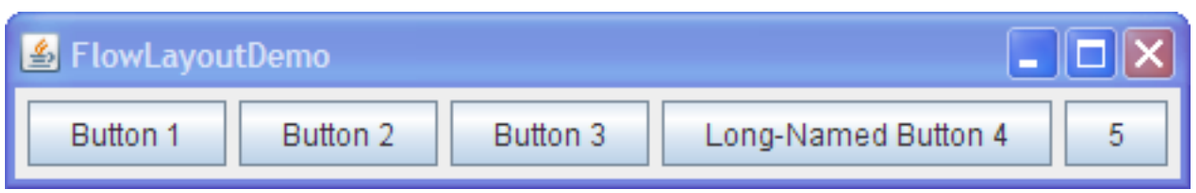
Etape 2 : Construire l’interface.

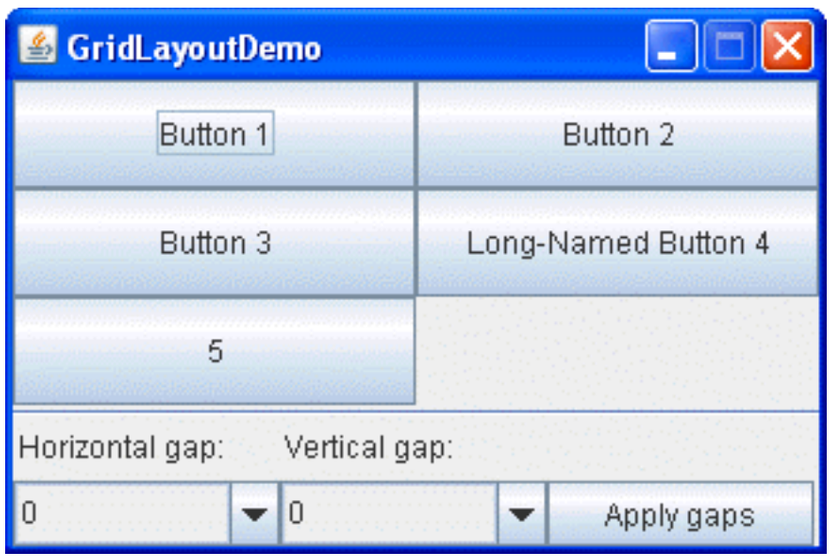
Etape 3 : Rajouter les événements.

Différents LayoutManager en Java



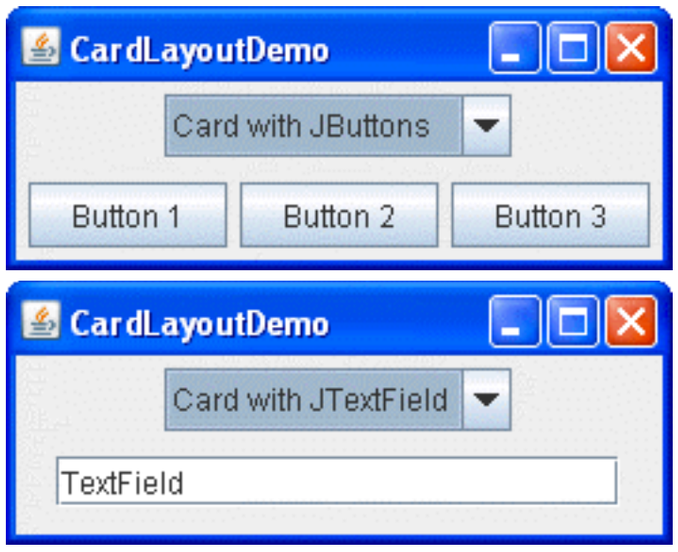


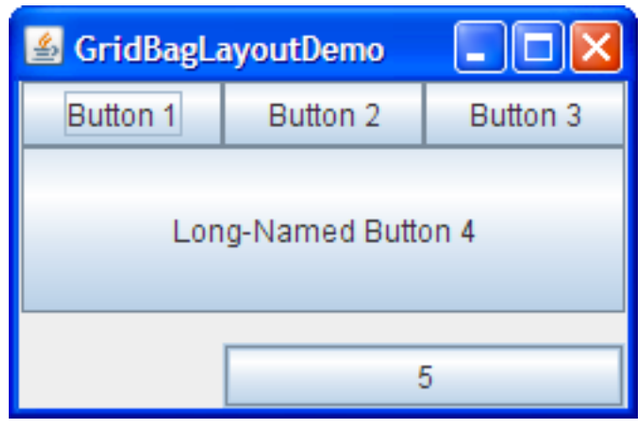


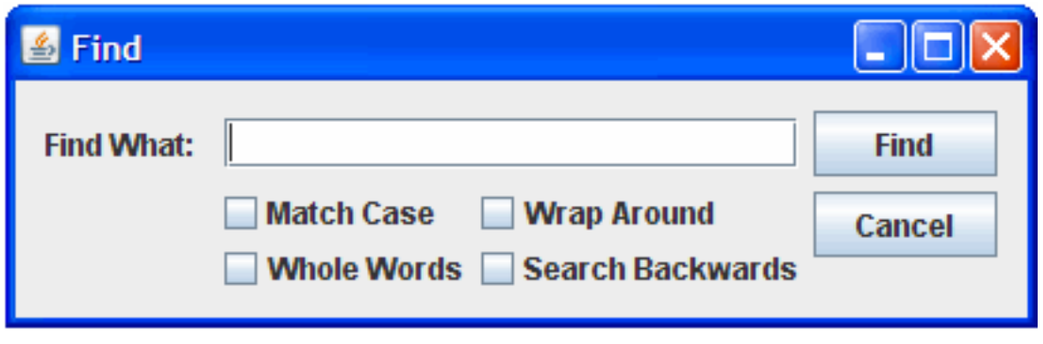


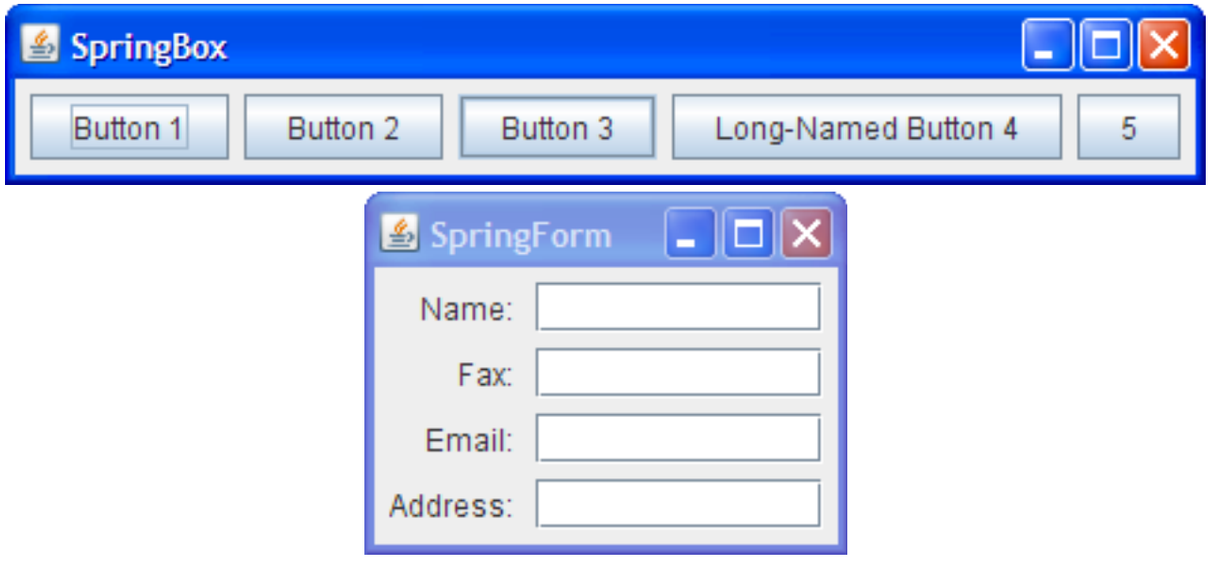
Source : https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

Différents LayoutManager en Java



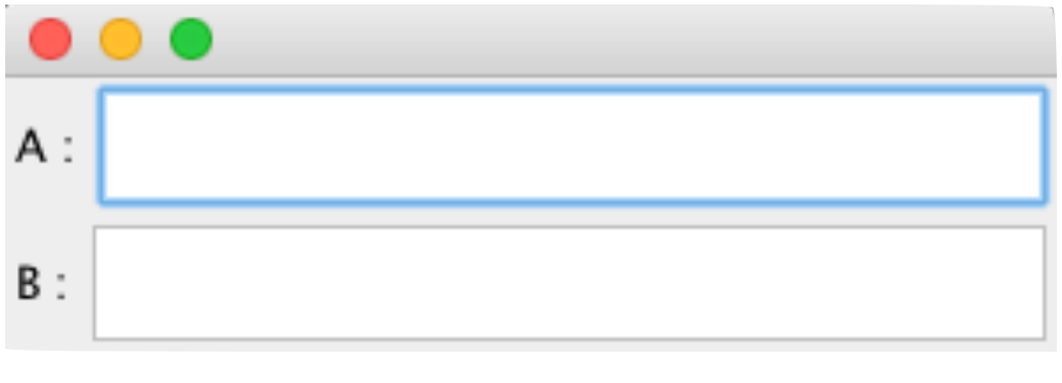






Source : https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

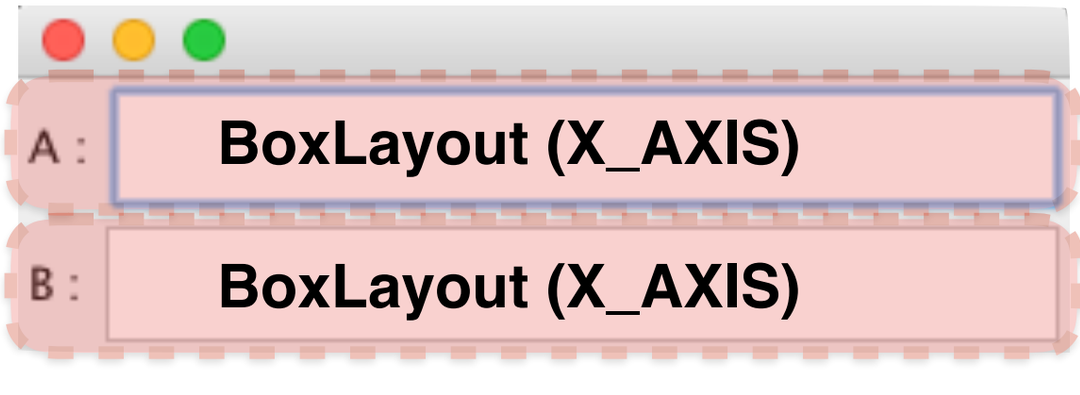
Construire son interface : Exemple



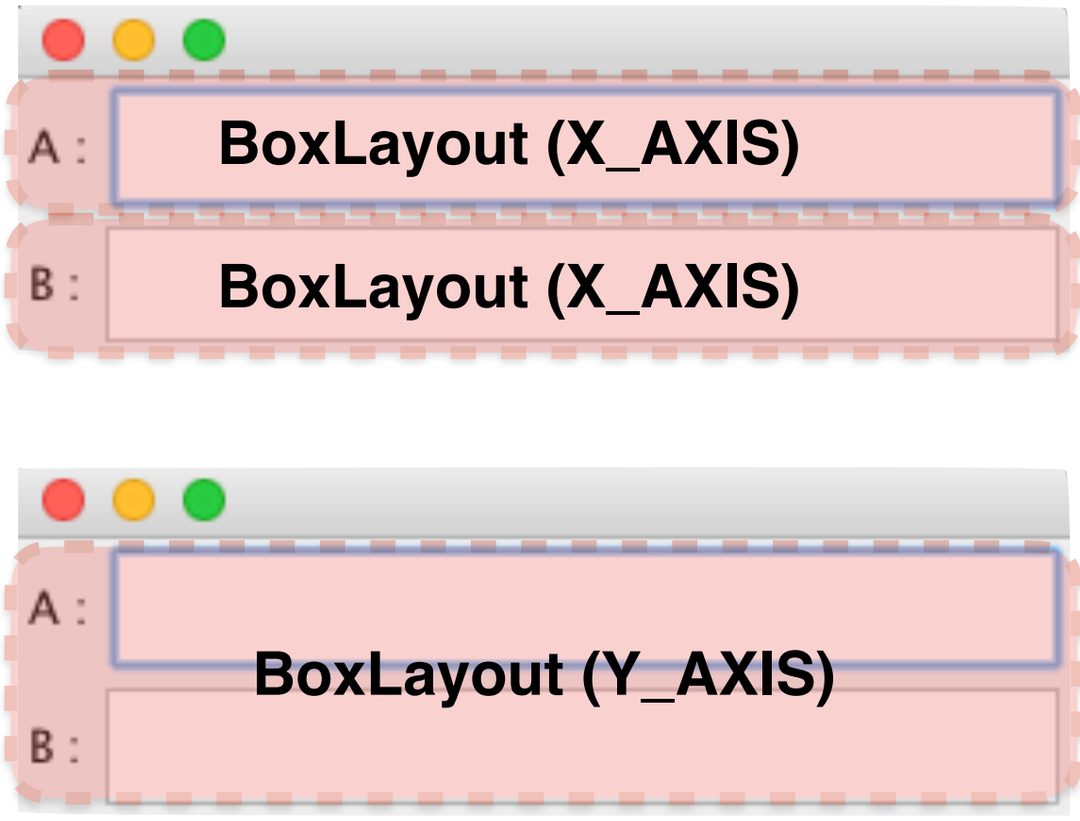
Etape 1 : Identifier le/les LayoutManager à utiliser.

Meilleur outil pour cela : Papier/Crayon !

Construire son interface : Exemple



Construire son interface : Exemple



Construire son interface : Exemple

Etape 2 : Construire l’interface.

Construire son interface : Exemple

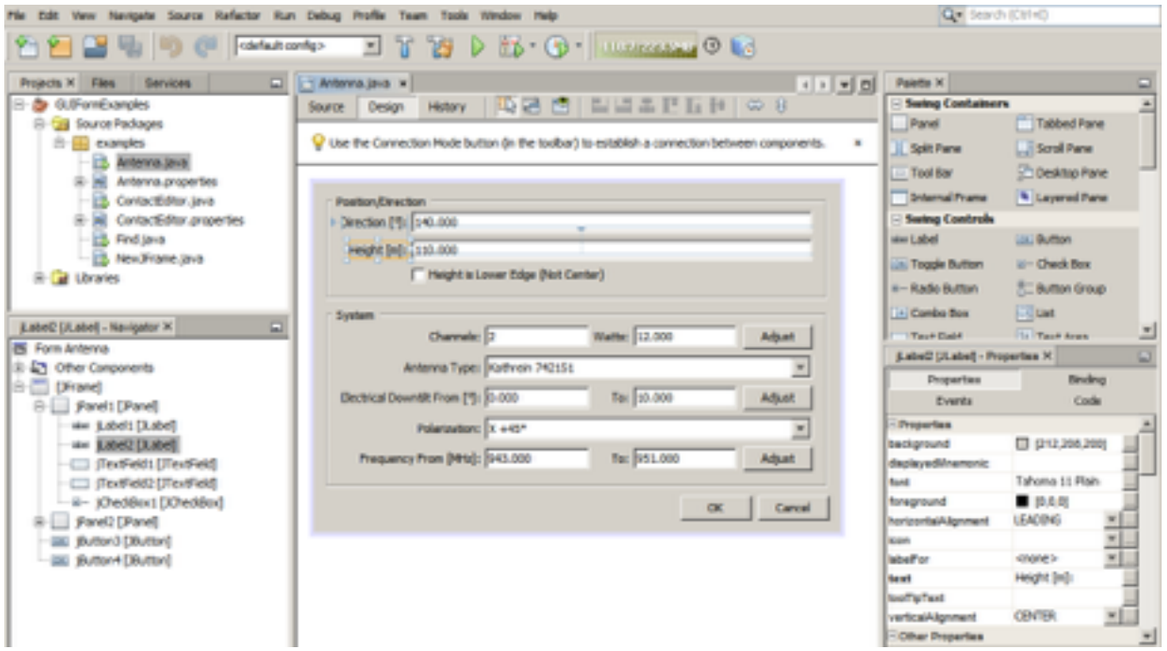
```
Window
    JPanel
      JLabel "A"
      JTextField

  JPanel
    JLabel "B"
    JTextField
```

Structure

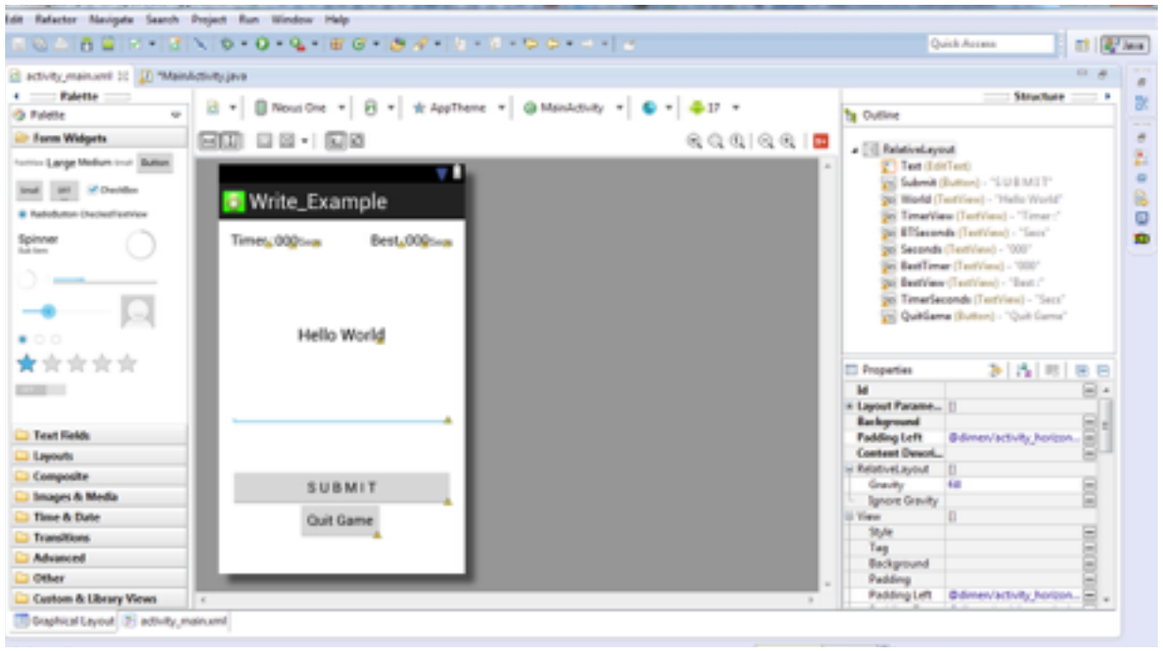
```
Container panel = getContentPane();
panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

JPanel panelA = new JPanel();
panelA.setLayout(new BoxLayout(panelB, BoxLayout.Y_AXIS));
panel.add(panelA);
panelA.add(new JLabel("A"));
panelA.add(new JTextField(15));

JPanel panelB = new JPanel();
panelB.setLayout(new BoxLayout(panelB, BoxLayout.Y_AXIS));
panel.add(panelB);
panelB.add(new JLabel("B"));
panelB.add(new JTextField(15));
```

Code

Gui Builder



Source : https://netbeans.org/



Source : stackoverflow.com/

- Utile pour construire des interfaces avancées.

- Complexe à utiliser.

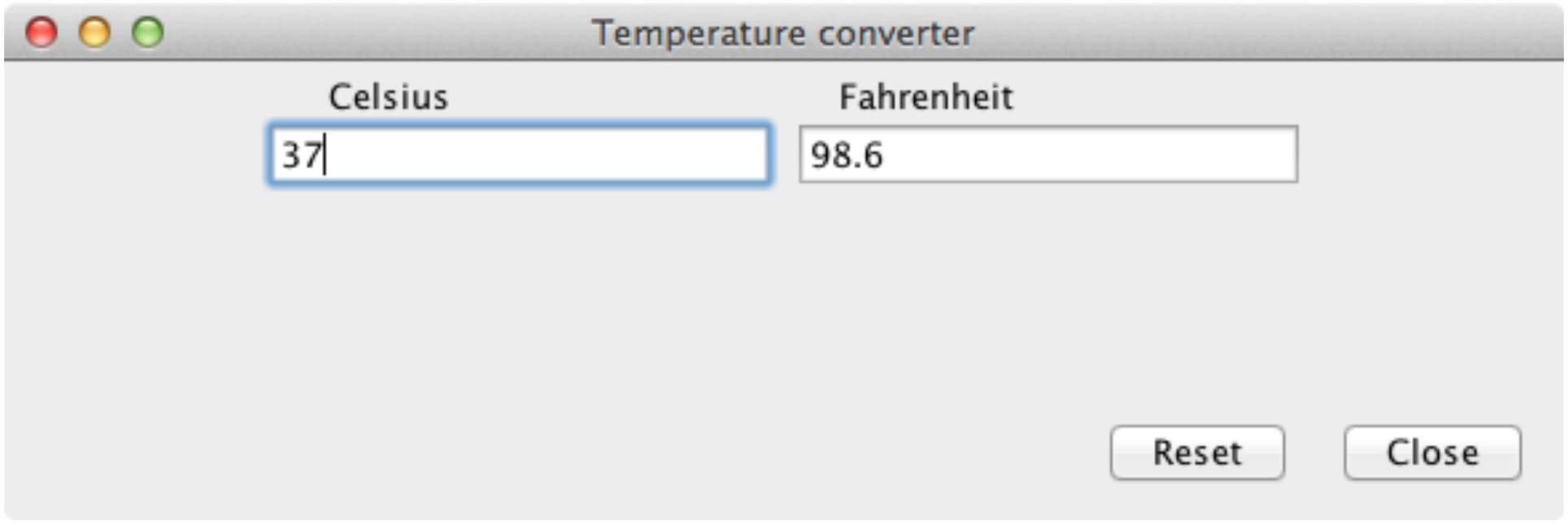
- Pas pendant ce cours.

Quelques sites utiles

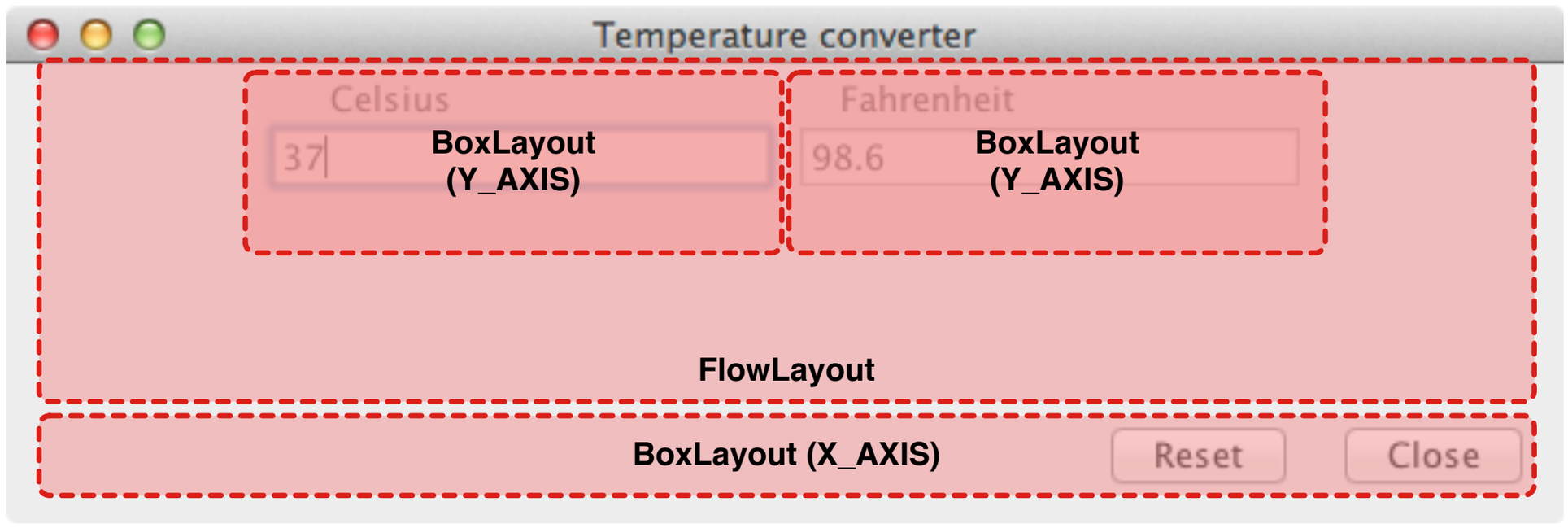
- La doc oracle : https://docs.oracle.com/javase/8/docs/api/

- Stack Overflow : http://stackoverflow.com/

Exercice 1

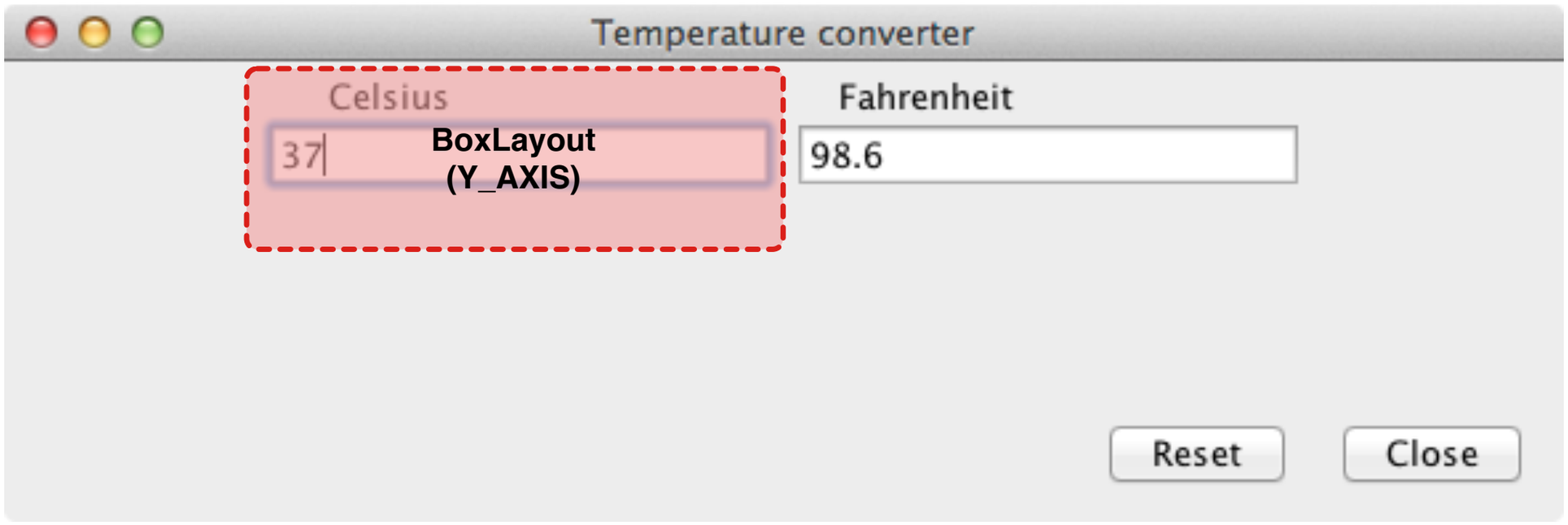


Avant de coder on identifie la structure de la fenêtre, puis on identifie les LayoutManager à utiliser.

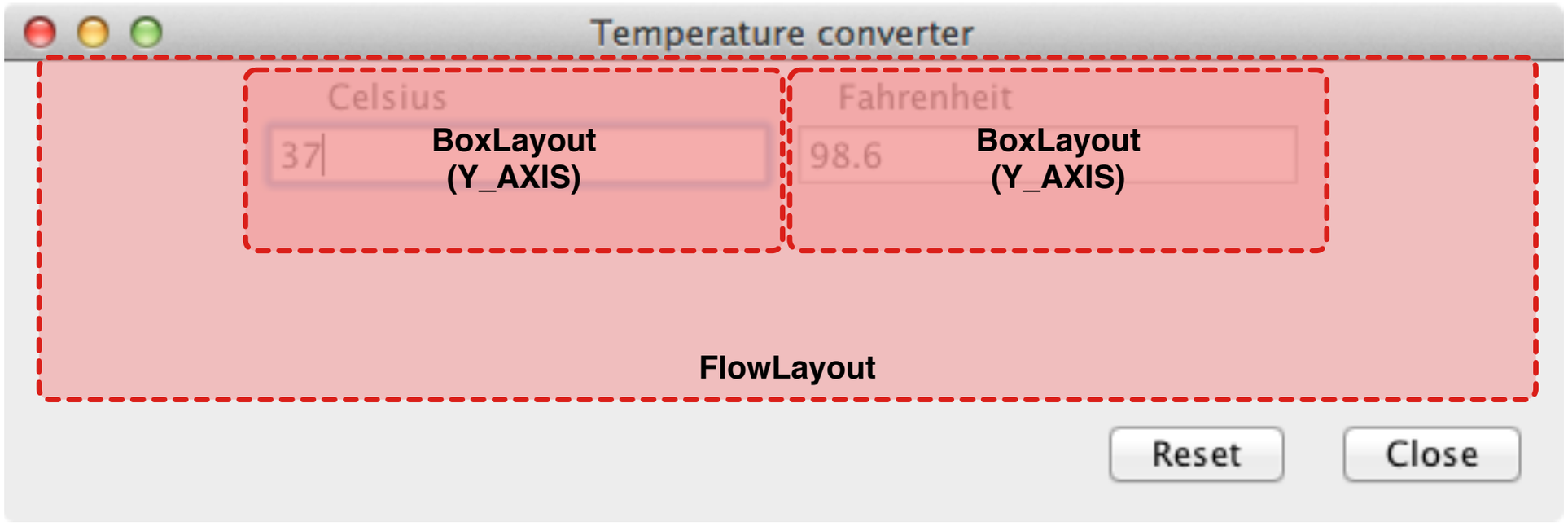


Dans cet exemple, nous avons subdivisé le layout en différent **JPanels** et choisi un LayoutManager spécifique pour chacun (**BoxLayout** et **FlowLayout**).

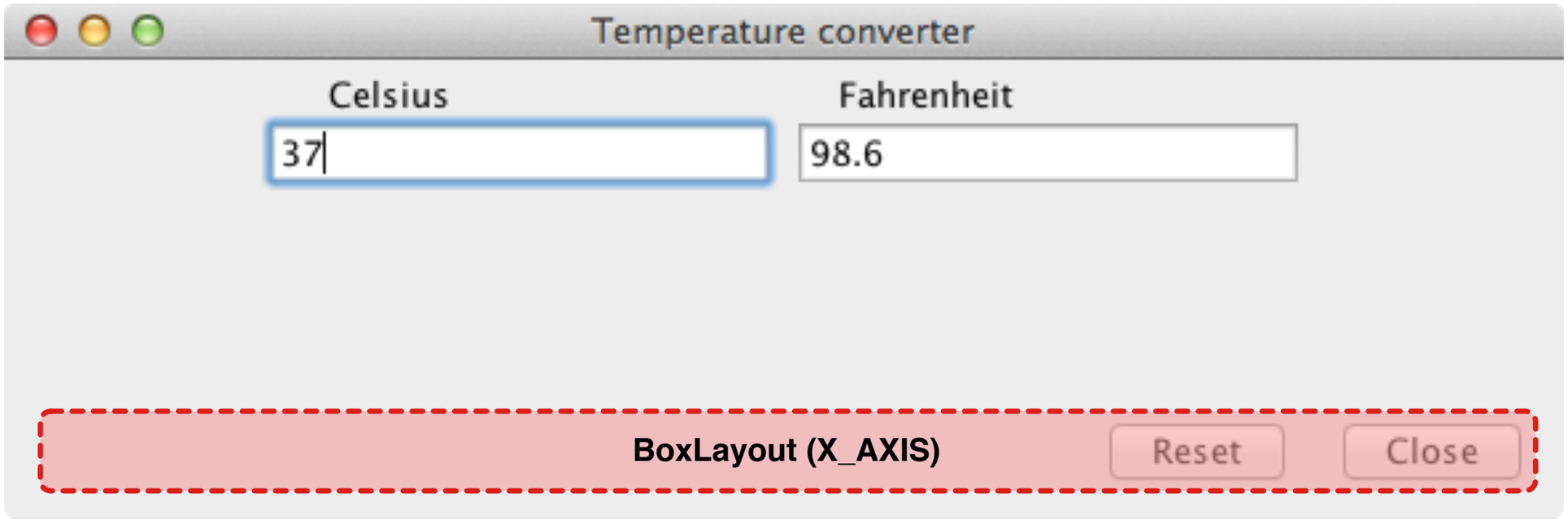
Il peut y avoir plusieurs solutions.



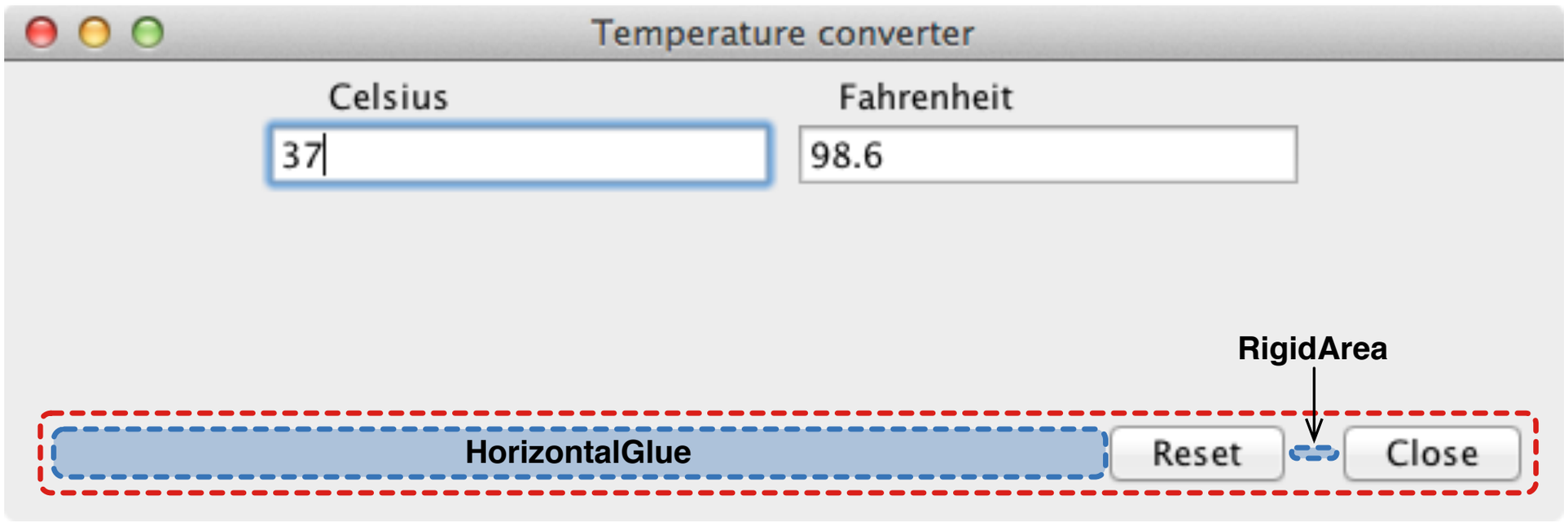

```
JPanel paneC = new JPanel();
paneC.setLayout(new BoxLayout(paneC, BoxLayout.Y_AXIS));
paneC.add(labelC);
paneC.add(textFieldC);
```



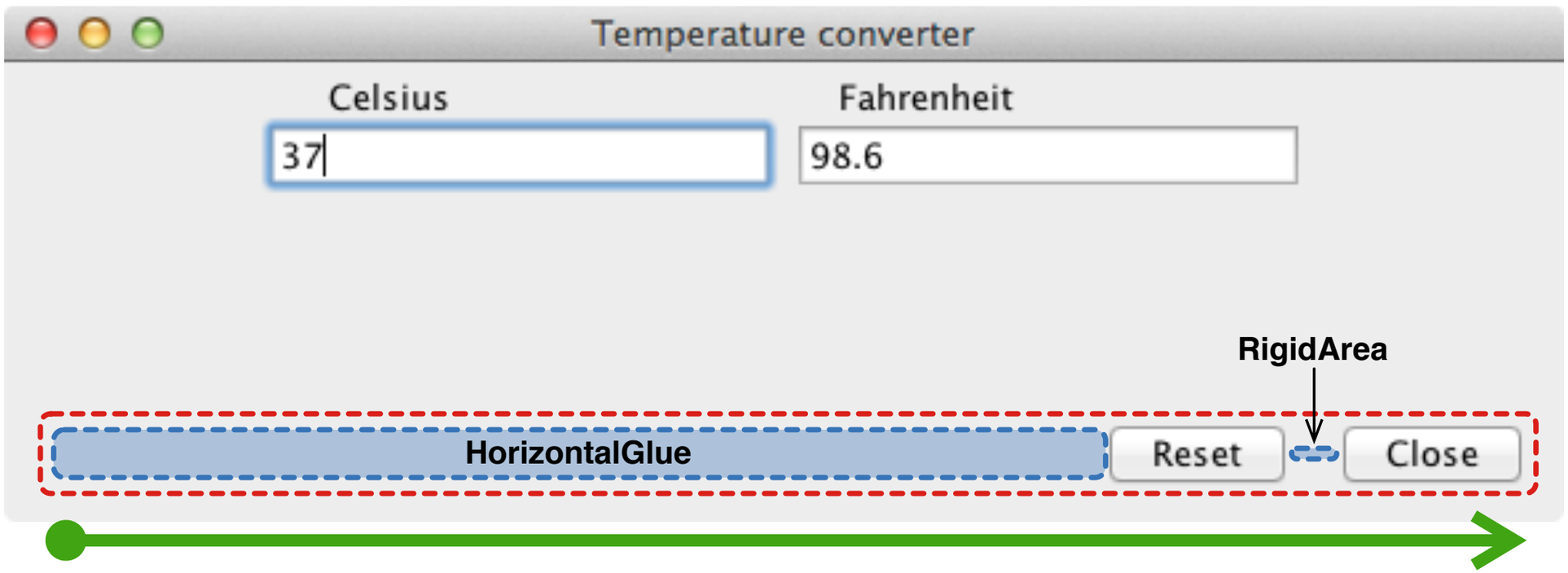

```
JPanel tempPane = new JPanel();
tempPane.add(paneC);
tempPane.add(paneF);
```




```
JPanel buttonPane = new JPanel();
buttonPane.setLayout(new BoxLayout(buttonPane, BoxLayout.X_AXIS));
buttonPane.setBorder(BorderFactory.createEmptyBorder(5, 10, 10, 10));
buttonPane.add(Box.createHorizontalGlue());
buttonPane.add(buttonReset);
buttonPane.add(Box.createRigidArea(new Dimension(10, 0)));
buttonPane.add(buttonClose);
```

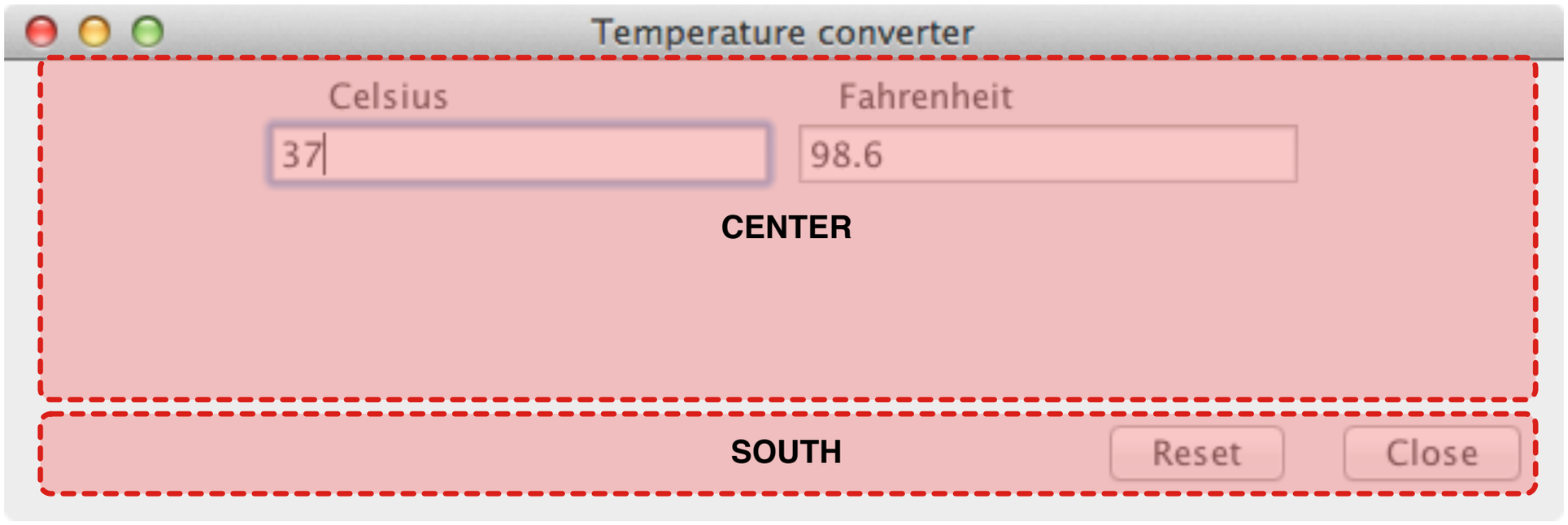



```
JPanel buttonPane = new JPanel();
buttonPane.setLayout(new BoxLayout(buttonPane, BoxLayout.X_AXIS));
buttonPane.setBorder(BorderFactory.createEmptyBorder(5, 10, 10, 10));
buttonPane.add(Box.createHorizontalGlue());
buttonPane.add(buttonReset);
buttonPane.add(Box.createRigidArea(new Dimension(10, 0)));
buttonPane.add(buttonClose);
```




```
JPanel buttonPane = new JPanel();
buttonPane.setLayout(new BoxLayout(buttonPane, BoxLayout.X_AXIS));
buttonPane.setBorder(BorderFactory.createEmptyBorder(5, 10, 10, 10));
buttonPane.add(Box.createHorizontalGlue());
buttonPane.add(buttonReset);
buttonPane.add(Box.createRigidArea(new Dimension(10, 0)));
buttonPane.add(buttonClose);
```

L’ordre d’insertion est important: les éléments sont ajouté **de la gauche vers la droite** pour les LayoutManager horizontaux et **du haut vers le bas** pour les LayoutManager verticaux.




```
Container mainPane = getContentPane();
mainPane.setLayout(new BorderLayout());
mainPane.add(tempPane, BorderLayout.CENTER);
mainPane.add(buttonPane, BorderLayout.SOUTH);
```

Ajout des listeners

Une fois que l'interface est faite, on ajoute les event listeners.

Referez-vous aux slides de cours ou à la doc Java Swing pour trouver le bon event listener pour un élément.

Chaque composant a des listeners différent.

Par exemple, **JTextField** peut avoir un **ActionListener** ou un **KeyListener**.