

TA Programming of Interactive Systems

<http://davidbonnet.com/is>

David Bonnet
bonnet@lri.fr

Cédric Fleury
cfleury@lri.fr

Layout Example

Exercise 1

Building the interface

Always start by **laying out** the widgets in the window.

Handle the functionality with the event listeners **after**.

Use **JPanels** to **structure** and **sub-divide** the layout.

Assign **LayoutManagers** to **JPanels** to define a specific layout.

Building the interface

Example of structure and resulting code:

Window

 JPanel

 JLabel "A"

 JTextField

 JPanel

 JLabel "B"

 JTextField

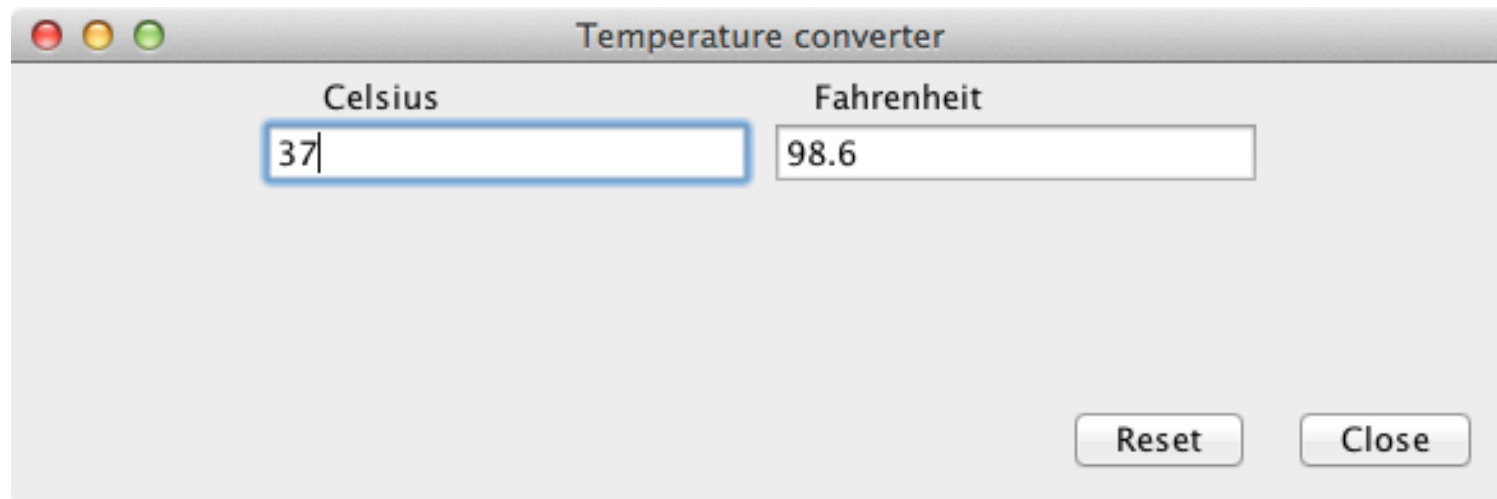
```
Container panel = getContentPane();
```

```
JPanel panelA = new JPanel();  
panel.add(panelA);  
panelA.add(new JLabel("A"));  
panelA.add(new JTextField(5));
```

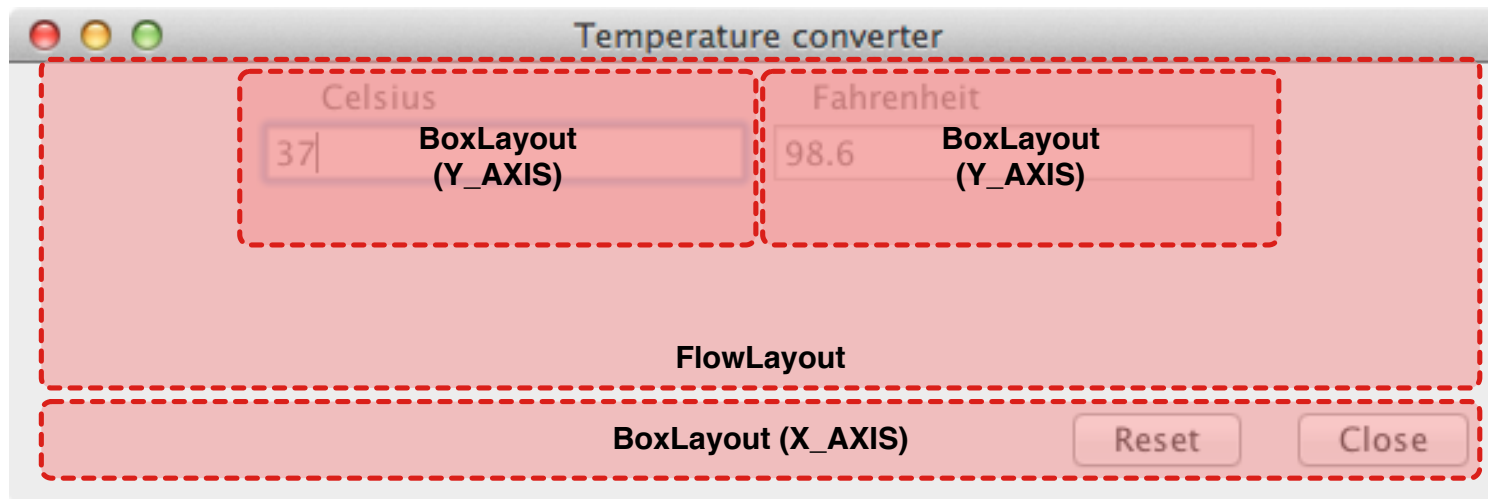
```
JPanel panelB = new JPanel();  
panel.add(panelB);  
panelB.add(new JLabel("B"));  
panelB.add(new JTextField(5));
```

Structure

Code

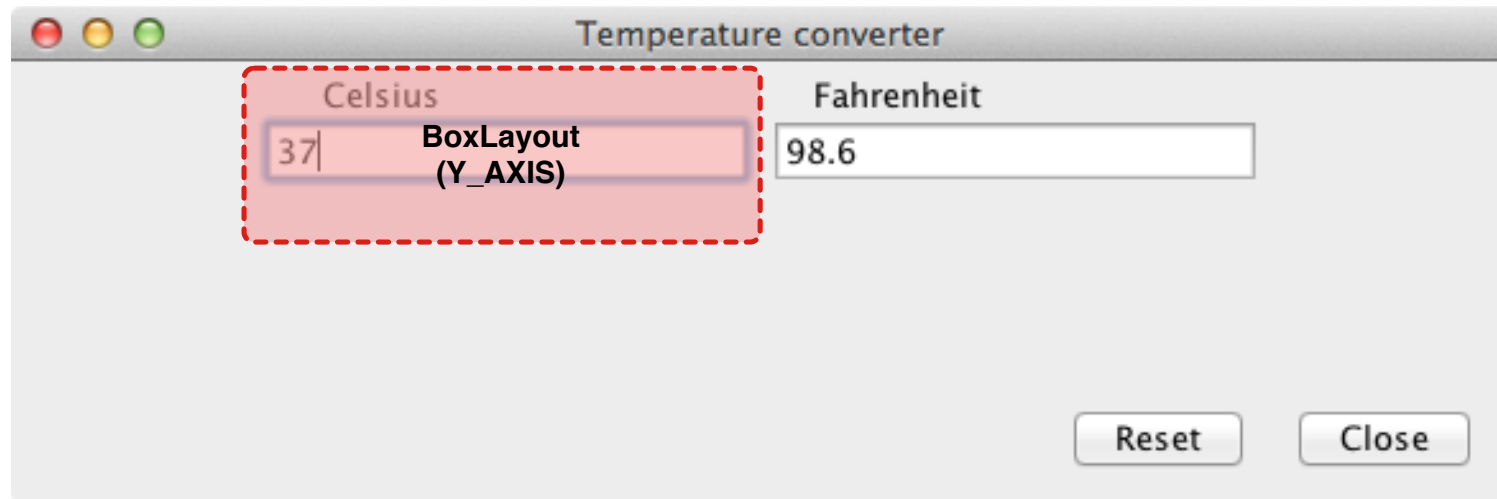


Before writing any code for the layout, **identify a structure** that sub-divides nicely into rectangular areas.

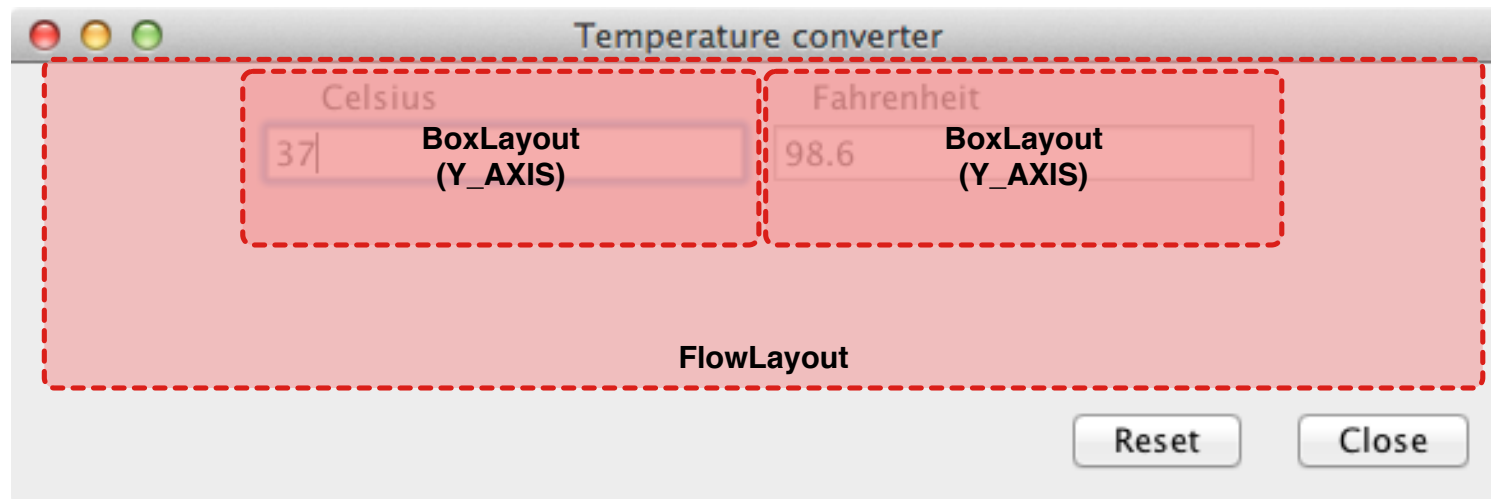


In this example, we have sub-divided the layout into different **JPanels** and chosen a specific layout (**BoxLayout** and **FlowLayout**) for each of them.

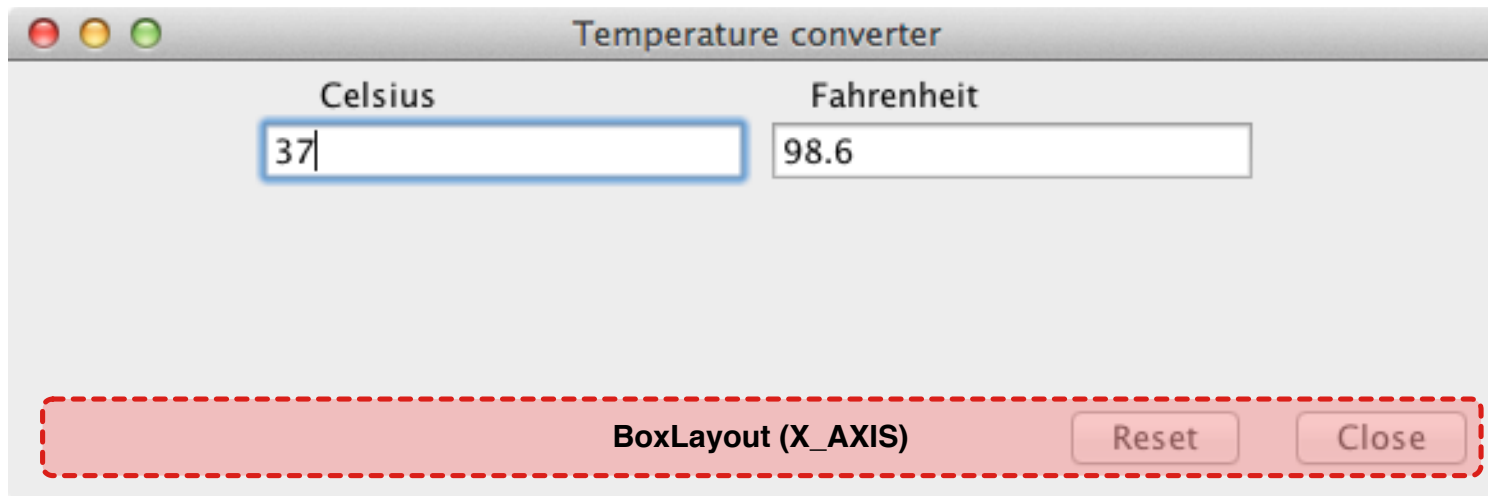
Note that there is not a unique solution.



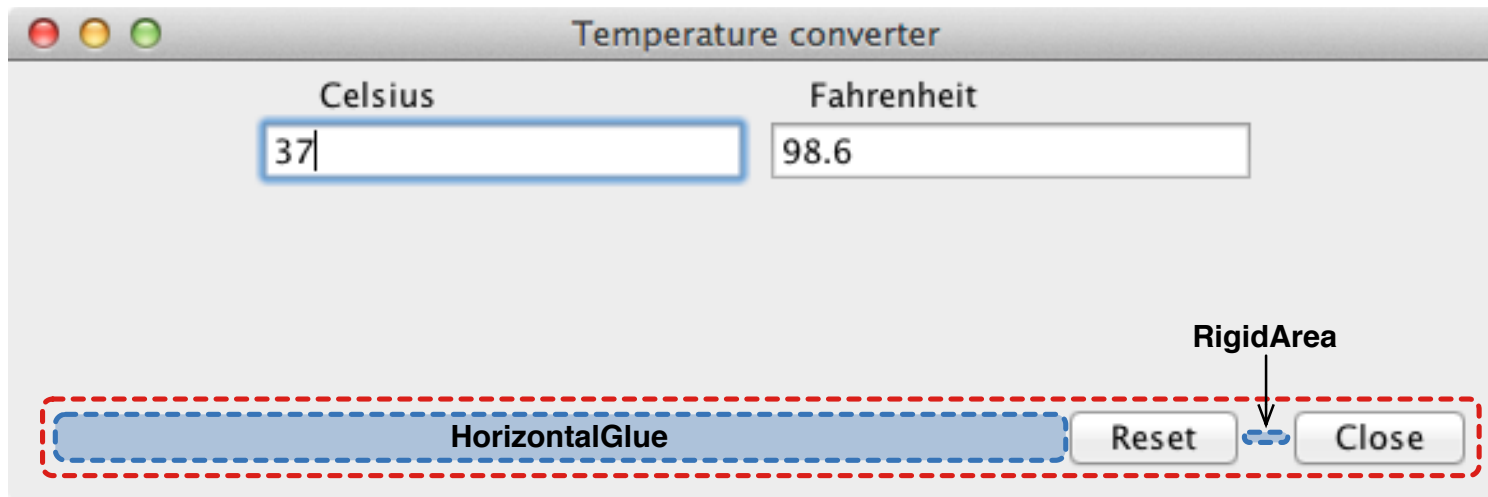
```
JPanel paneC = new JPanel();  
paneC.setLayout(new BorderLayout(paneC, BorderLayout.Y_AXIS));  
paneC.add(labelC);  
paneC.add(textFieldC);
```



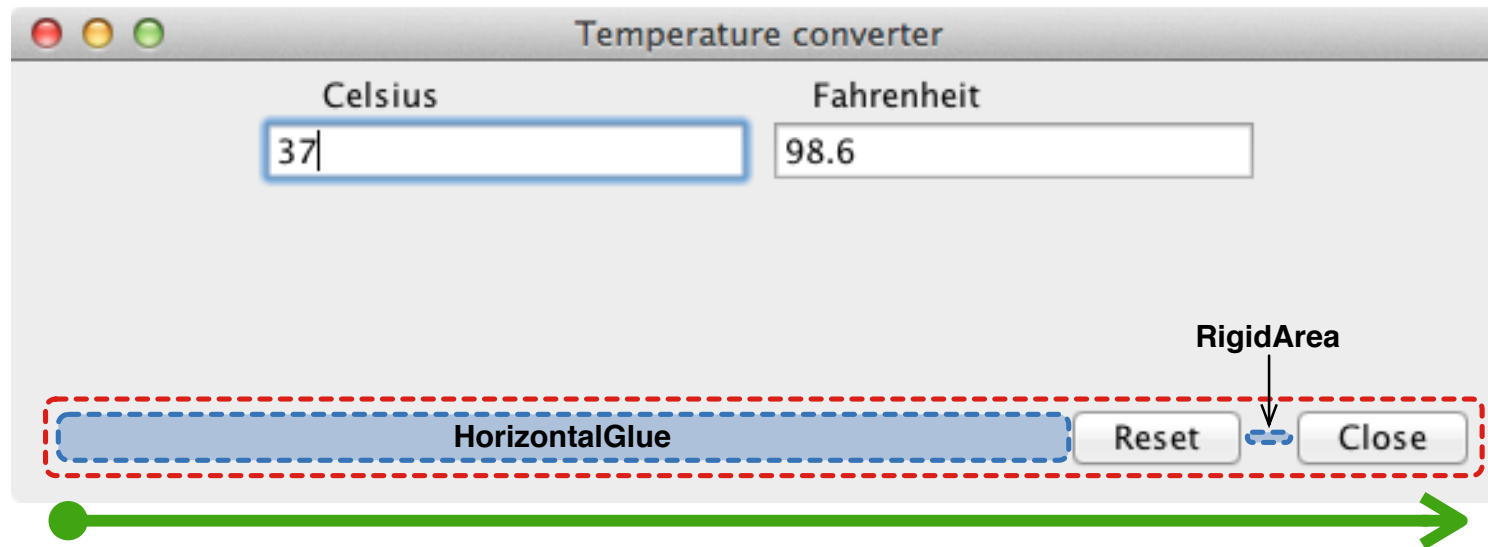
```
JPanel tempPane = new JPanel();  
tempPane.add(paneC);  
tempPane.add(paneF);
```

```
JPanel buttonPane = new JPanel();  
buttonPane.setLayout(new BorderLayout(buttonPane, BorderLayout.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);
```

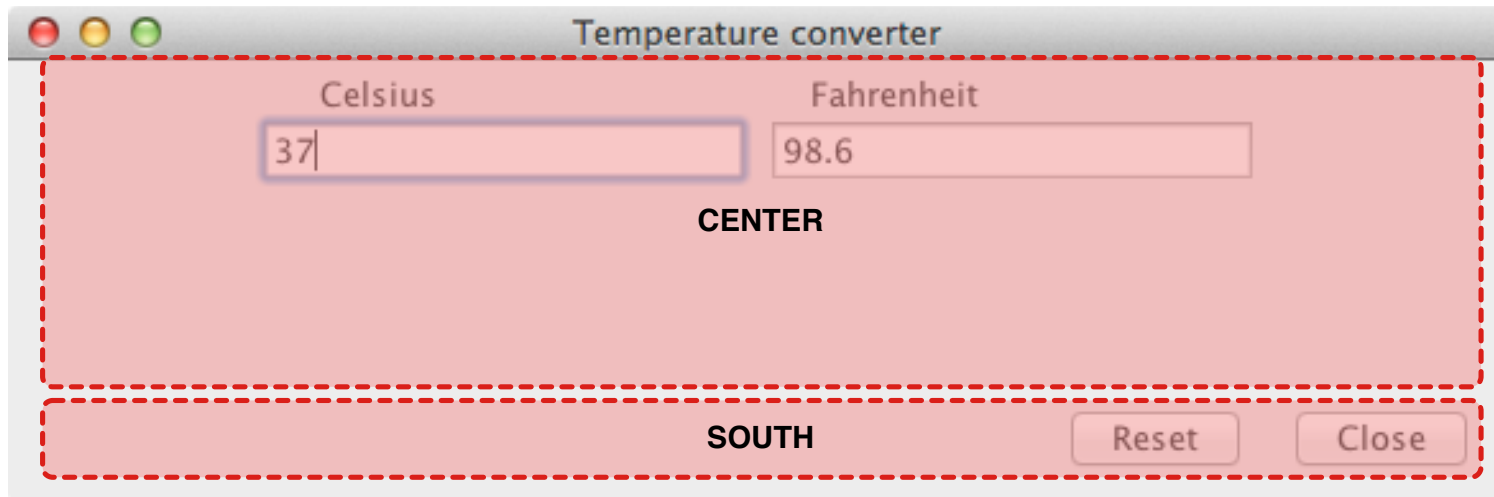


```

JPanel buttonPane = new JPanel();
buttonPane.setLayout(new BorderLayout(buttonPane, BorderLayout.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);

```

Insertion order is important: items are added from **left to right** for horizontal layouts and **top to bottom** for vertical layouts.



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

Adding listeners

Once the interface is laid out, add the event listeners.

Refer to the lecture slides or the Java Swing documentation to find the right type of listener for a given widget.

Note that each widget can support different types of listeners.

For example, the **JTextField** widget supports the **ActionListener** and **KeyListener**.