

# CS120: More GUIs

W. Michael Petullo

University of Wisconsin–La Crosse

As of October 6, 2021



# More GUIs: NetBeans Builder



The screenshot displays the NetBeans Builder interface for the file `Antenna.java`. The main workspace shows a GUI design with a central panel containing several components: `jLabel1 [JLabel]`, `jLabel2 [JLabel]`, `jTextField1 [JTextField]`, `jTextField2 [JTextField]`, and `jCheckBox1 [JCheckBox]`. The `jLabel2` component is selected, and its `Properties` dialog is open. The dialog is divided into two sections: **Position/Direction** and **System**.

**Position/Direction Section:**

- `Direction [°]`: 140.000
- `Height [m]`: 110.000
- Height is Lower Edge (Not Center)

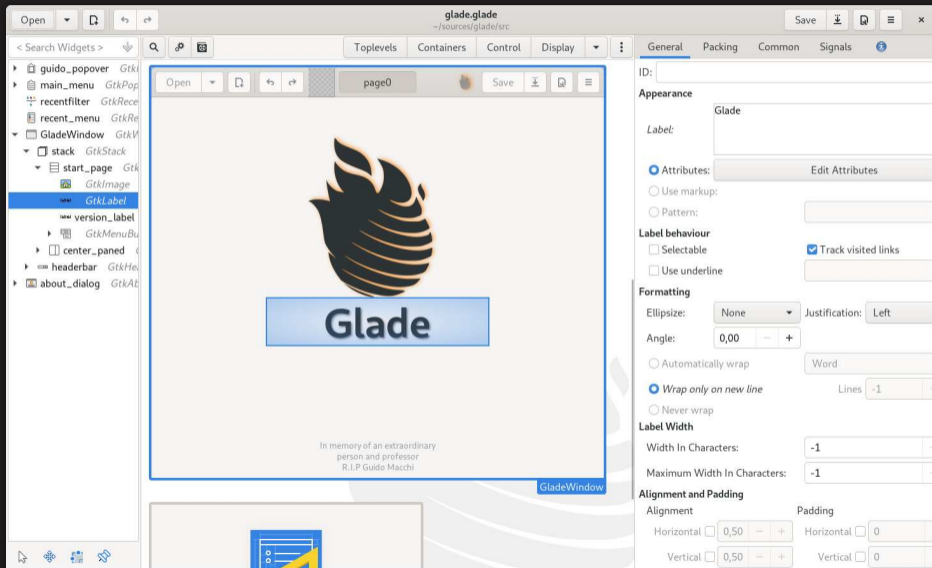
**System Section:**

- `Channels`: 2
- `Watts`: 12.000
- `Antenna Type`: Kathrein 742151
- `Electrical Downtilt From [°]`: 0.000
- `To`: 10.000
- `Polarization`: X +45°
- `Frequency From [MHz]`: 943.000
- `To`: 951.000

The `Properties` dialog also includes a `Background` color picker set to [212,208,200] and a `Text` color picker set to [0,0,0]. The `Text` property is currently set to `Height [m]:`.

The **Palette** on the right lists various Swing Containers and Controls. The **Properties** window for `jLabel2` is also visible, showing the `Properties` tab with various attributes like `background`, `font`, `foreground`, `horizontalAlignment`, `icon`, `labelFor`, `text`, `toolTipText`, `verticalAlignment`, and `Other Properties`.

# More GUIs: Glade



The screenshot displays the Glade GUI designer interface. The main window, titled "glade.glade", shows a preview of a window with a blue border and a title bar containing "page0" and "Save" buttons. The window content features a large black and orange flame logo above a blue rectangular button with the text "Glade". Below the button, there is a small text block: "In memory of an extraordinary person and professor R.I.P Guido Macchi".

The left sidebar shows a widget tree with the following structure:

- guido\_popover GtkPopover
- main\_menu GtkPopover
- recentfilter GtkRecentFilter
- recent\_menu GtkRecentFilter
- GladeWindow GtkWindow
  - stack GtkStack
    - start\_page GtkRecentFilter
      - GtkImage
      - GtkLabel (selected)
      - version\_label
  - GtkMenuButton
    - center\_paned
  - headerbar GtkHeaderBar
  - about\_dialog GtkAboutDialog

The right sidebar shows the "General" properties panel for the selected widget:

- ID: (empty)
- Appearance
  - Label: Glade
  - Attributes: Edit Attributes
  - Use markup: (radio button)
  - Pattern: (text field)
- Label behaviour
  - Selectable: (checkbox)
  - Track visited links: (checked checkbox)
  - Use underline: (checkbox)
- Formatting
  - Ellipsize: None
  - Justification: Left
  - Angle: 0,00
  - Automatically wrap: (radio button)
  - Wrap only on new line: (checked radio button)
  - Never wrap: (radio button)
  - Word: (text field)
  - Lines: -1
- Label Width
  - Width In Characters: -1
  - Maximum Width In Characters: -1
- Alignment and Padding
  - Alignment: Horizontal 0,50, Vertical 0,50
  - Padding: Horizontal 0, Vertical 0



# Hello.java



---

Interface	Methods	Establish Listener
<code>ActionListener</code>	<code>actionPerformed(ActionEvent e)</code> <code>keyPressed(KeyEvent e)</code>	<code>addActionListener(...)</code>
<code>KeyListener</code>	<code>keyReleased(KeyEvent e)</code> <code>keyTyped(KeyEvent e)</code> <code>mouseClicked(MouseEvent e)</code> <code>mouseEntered(MouseEvent e)</code>	<code>addKeyListener(...)</code>
<code>MouseListener</code>	<code>mouseExited(MouseEvent e)</code> <code>mousePressed(MouseEvent e)</code> <code>mouseReleased(MouseEvent e)</code>	<code>addMouseListener(...)</code>

---



Action-type events are widget-specific:

- ▶ A JButton's action indicates a click
- ▶ A JFileChooser's action indicates a selected file
- ▶ A JTextField's action indicates enter has been pressed



## Clock.java



# File.java





# More GUIs: KeyListener

Method	Purpose
<code>keyPressed(KeyEvent e)</code>	Key pressed over widget
<code>keyReleased(KeyEvent e)</code>	Key released over widget
<code>keyTyped(MouseEvent e)</code>	<u>Printable</u> key pressed over widget

Printable keys do not include the arrow keys, for example. Pressing and releasing 'A' will trigger all three methods. Pressing and releasing an arrow key will trigger the first two methods.

```
public void keyPressed(KeyEvent k){
    if (k.getKeyCode() ==
        KeyEvent.VK_UP) {
        // Do something.
    }
}
```

```
public void keyTyped(KeyEvent k){
    if (k.getKeyChar() == 'a') {
        // Do something.
    }
}
```

---

Method	Purpose
<code>mouseClicked(MouseEvent e)</code>	Mouse button pressed and released over widget
<code>mouseEntered(MouseEvent e)</code>	Mouse over widget
<code>mouseExited(MouseEvent e)</code>	Mouse no longer over widget
<code>mousePressed(MouseEvent e)</code>	Mouse button pressed while over widget
<code>mouseReleased(MouseEvent e)</code>	Mouse button released while over widget

---

```
public void mouseClicked(MouseEvent m){
    if (m.getButton == MouseEvent.BUTTON1) {
        // Do something.
    } else if (m.getButton == MouseEvent.BUTTON2) {
        // Do something else.
    }
}
```



## Mouse.java



Refer to the Java API documentation for a  
wealth of widgets and features

`https:`

`//docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html`



Graded Homework Aquinas: button in Java

Ungraded Labs Aquinas: hello3 and typist in Java

<https://www.flyn.org/courses/cs120-2021-fall/schedule>