Lecture 21

GUI Layout

Announcements for This Lecture

Prelim II

- Tonight in Olin 155
  - 7:30 – 9 pm
  - Will be graded tonight
  - Grades by tomorrow
- Make-up Thursday 4:30
  - For students that contacted me with prelim conflicts
  - Graded by the weekend
- Review slides are online

Assignments

- A6 due Thursday
  - Hopefully you have done all but Steganography
  - To be graded this weekend
- Assignment A7 posted Friday
  - Last assignment of semester
  - Sizeable project; longer than the previous ones
  - Will give you until Saturday after last day of classes

Announcements for This Lecture

Prelim II

- Tonight in Olin 155
  - 7:30 – 9 pm
  - Will be grade
  - Grades by ton
- Make-up Thur
  - For students t
    me with preli
  - Graded by the weekend
- Review slides are online

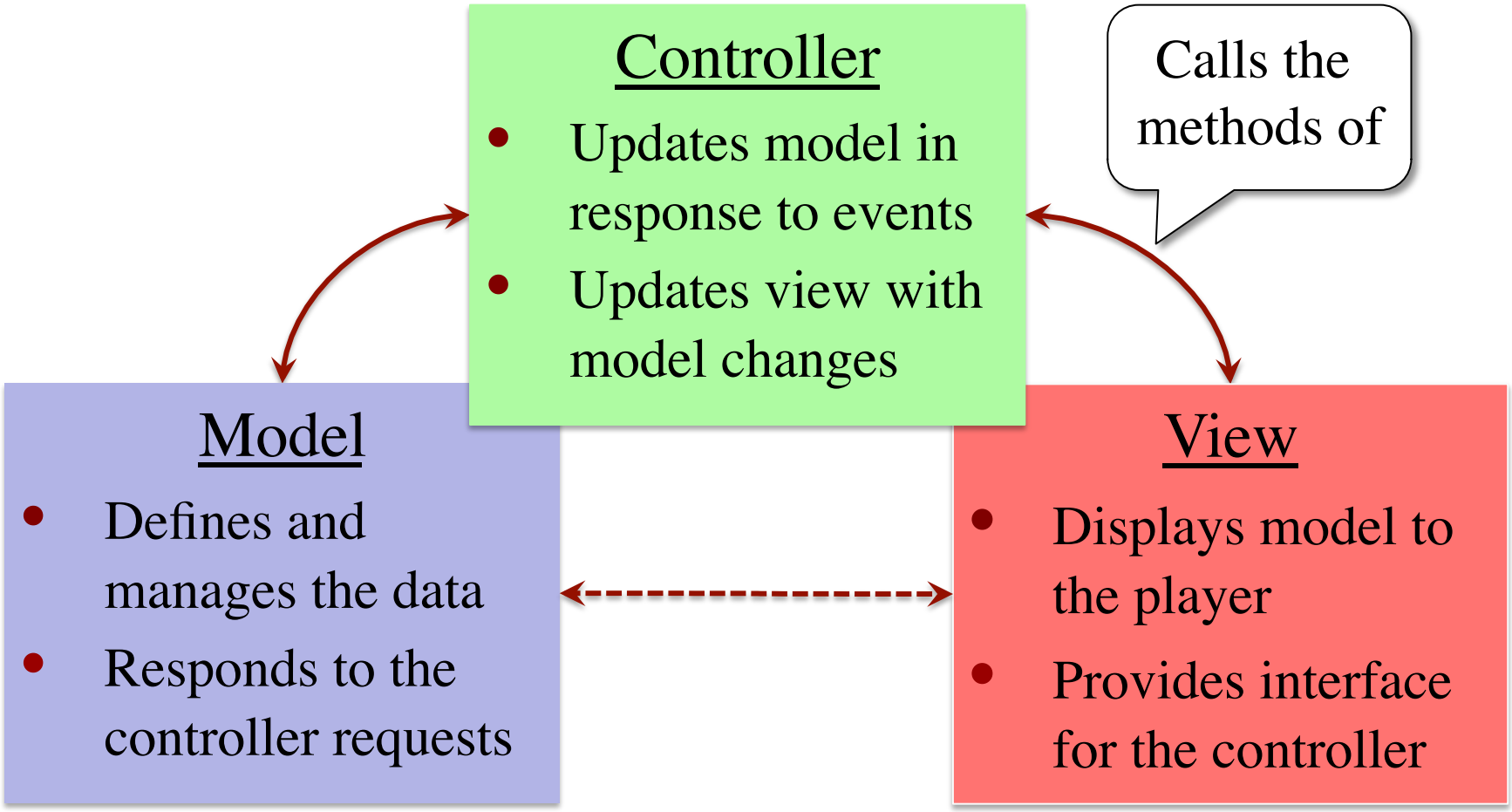
Lot of demo code today!

- Slides are more pictoral
- Need to see demos for code
- All will be posted online

Assignments

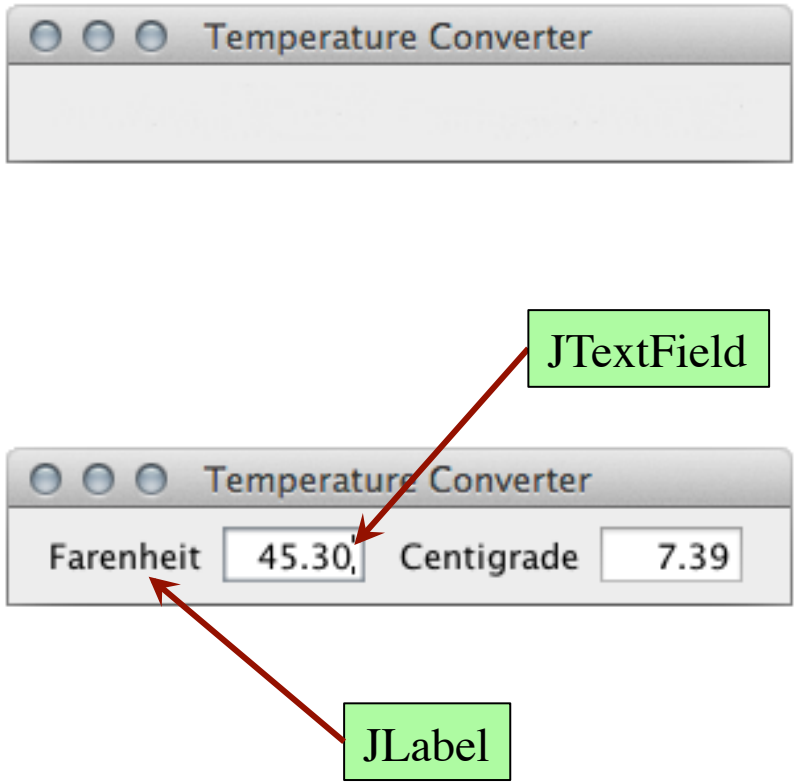
- A6 due Thursday
  - Hopefully you have done all
    nography
    ded this weekend
  - A7 posted Friday
    nment of semester
    project; longer than
    the previous ones
  - Will give you until Saturday after last day of classes

Review: Model-View-Controller Pattern



The Limitations of JFrame

- JFrame is just a Window
  - Can resize it
  - Can close it
  - Not much else
- To do more, you need **GUI components**
  - Items inside a JFrame
  - Ex: Buttons, Text Boxes
- Two main Java packages
  - java.awt: "old GUI"
  - javax.swing: "Swing GUI"



AWT vs. Swing

Abstract Window Toolkit

- Uses the standard interface
  - Mac looks like Mac
  - Windows like Windows
- Violates Java "portability"
  - **Demo**: AWTFile.java
- Very rarely used today
  - Handling input is messy
  - But superclass of Swing classes, so have to include

Swing API

- Codename that "stuck"
- Has pluggable look & feel
  - Mac can look like Windows
  - Default same on all platforms
  - Demo: SwingFile.java
- Now the default component collection in Java
  - Very easy to use
  - Programmers like uniformity

Swing Components

`JButton`: a pushbutton that can be clicked by mouse

`JCheckbox`: can be on (true) or off (false)

`JComboBox`: a popup menu of user choices

`JLabel`: a text label

`JList`: scrolling list of user-chooseable items

`JScrollbar`: a scroll bar

`JTextField`: allows editing of a single line of text

`JTextArea`: multiline region for displaying and editing text

`JPanel`: used for containing and grouping components

`JDialog`: window used for user input

`JFrame`: top-level window with frame and border

…

Buttons

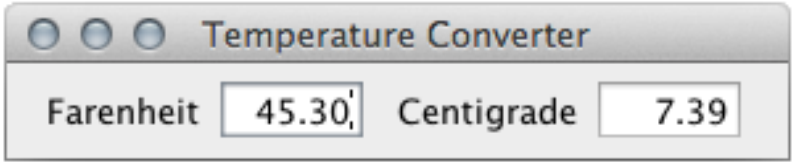
List

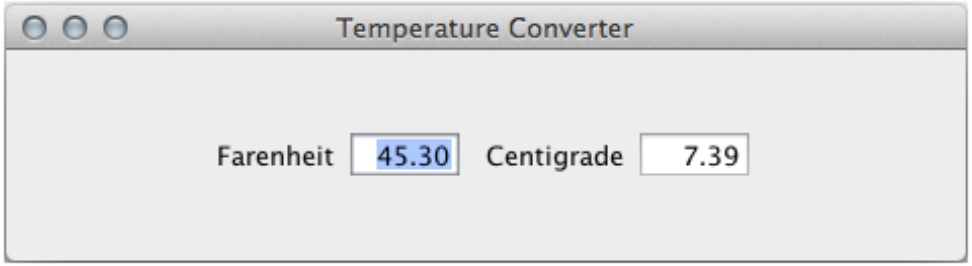
Slider

Main Challenges in GUI Applications

Layout

- Arranging items the screen
  - Java has many components
  - But where do they go?
- **Challenge**: Resizing
  - Want components to "behave nicely" as you resize
  - Change size of components
  - Change padding in between
- LayoutManagers do both





Main Challenges in GUI Applications

Layout

- Arranging items the screen
  - Java has many components
  - But where do they go?
- **Challenge**: Resizing
  - Want components to "behave nicely" as you resize
  - Change size of components
  - Change padding in between
- LayoutManagers do both

Input Handling

- Many types of input
  - button pushed
  - text typed
  - mouse clicked …
- Want app to react to input
  - Otherwise GUI looks pretty, but does nothing
- **Topic of next lecture**

Main Challenges in GUI Applications

View

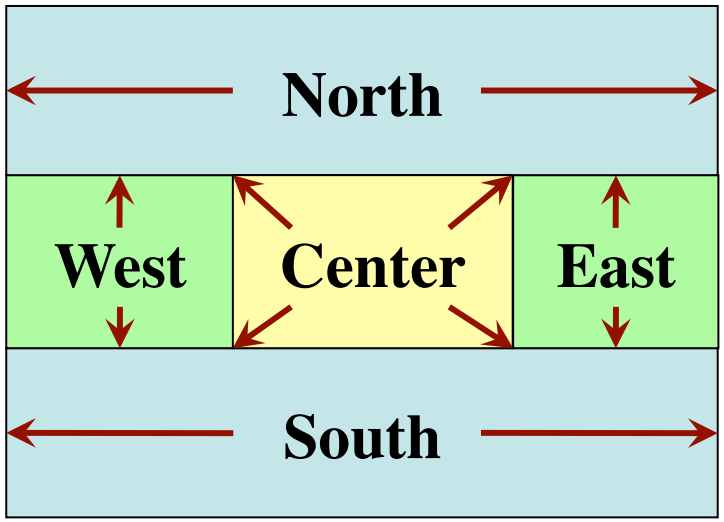
- Arranging items the screen
  - Java has many components
  - But where do they go?
- **Challenge**: Resizing
  - Want components to "behave nicely" as you resize
  - Change size of components
  - Change padding in between
- LayoutManagers do both

Controller

- Many types of input
  - button pushed
  - text typed
  - mouse clicked …
- Want app to react to input
  - Otherwise GUI looks pretty, but does nothing
- **Topic of next lecture**

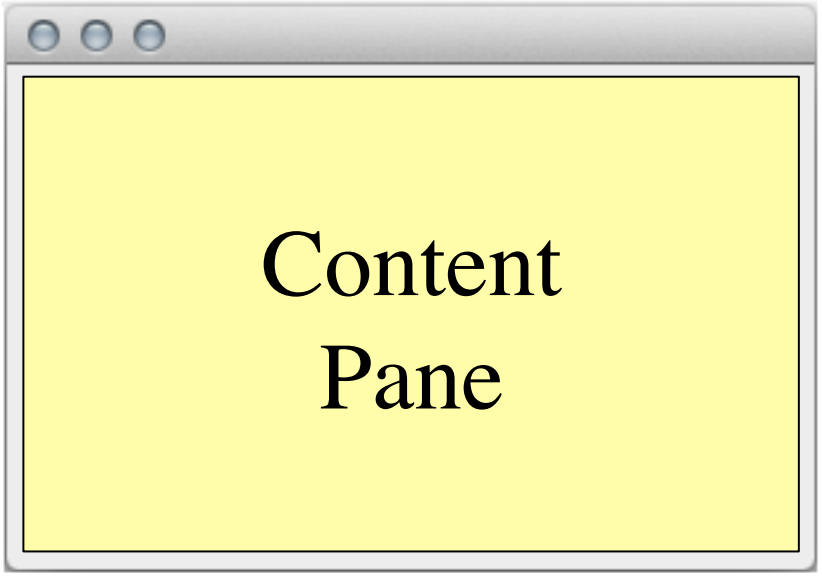
Example: BorderLayout

- Container has 5 directions
  - When add, specify direction
  - Ex: BorderLayout.CENTER
  - One component to a direction
- Resizing stretches components
  - Sides stretch vertically
  - Bottom/top stretch horizontally
  - Center stretches in both ways
- **Demo**: TestBorder.java



Adding Components to JFrames

- Never add to JFrame directly.
  - `getContentPane().add(...)`
- getContentPane() provides double buffering
  - Draws components to image, then displays image
  - Faster way to redraw if window moves



Alternate Layouts: FlowLayout

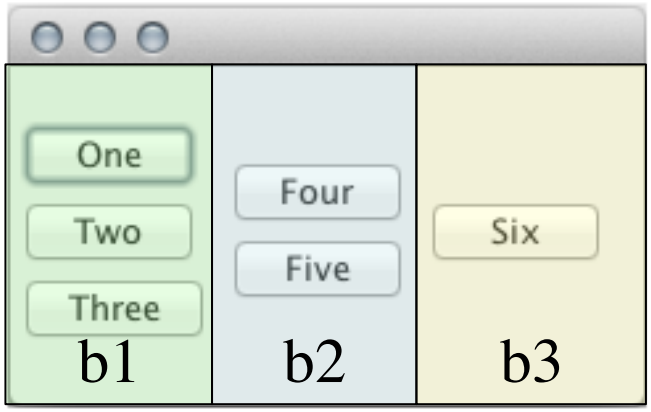
- Use a left-to-right "flow"
  - If one row fills, start on next row
  - Components "wrap" around
  - By default, components are centered in the container.
- Resizing affects layout only
  - Components retain their size
  - But the wrap may be affected
- **Demo**: TestFlow.java

Alternate Layouts: GridLayout

- Uses a rectangular grid
  - Specify no. of rows, columns
  - Ex: new GridLayout(3,2);
  - Tries to fill each gridbox
  - Leaves blanks if cannot
- Resizing affects components
  - Stretched to equal size
  - Cannot make different sizes
- **Demo**: TestGrid.java

BoxLayout: Ideal for Nesting

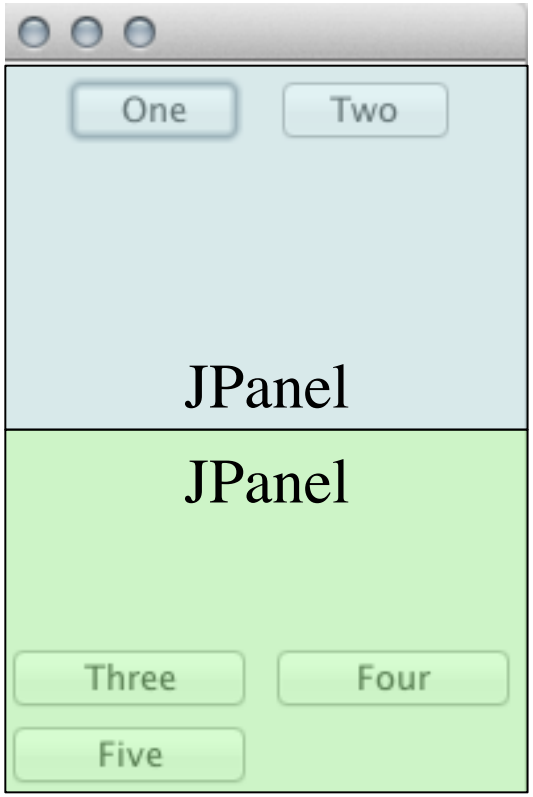
- BoxLayout
  - Arranges components in line
  - No wrap (like FlowLayout)
  - Either horizontal/vertical
- Box: JPanel w/ BoxLayout
  - Box b1= **new** Box(BoxLayout.Y_AXIS);
  - Makes layout quick
- **Demo**: BoxGrouping.java



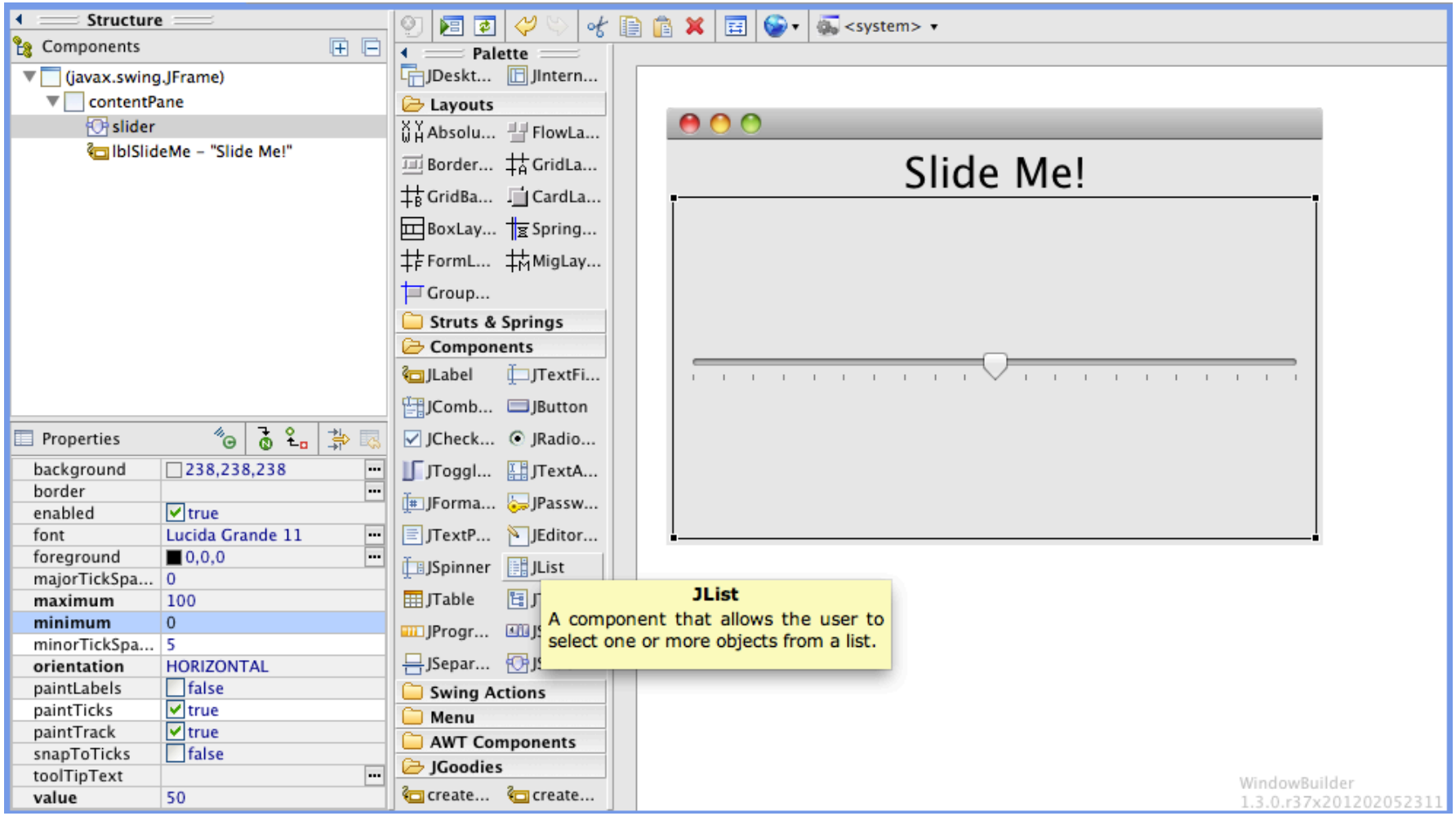
- Nested boxes
  - Three vertical boxes
  - Inside horizontal box

Nesting Layouts

- Want more interesting layouts
  - **Idea**: nest layouts in each other
  - Can get fine padding control
- Useful class: JPanel
  - Invisible component
  - Container for other components
  - Can take a LayoutManager
- **Demo**: PanelGrouping.java

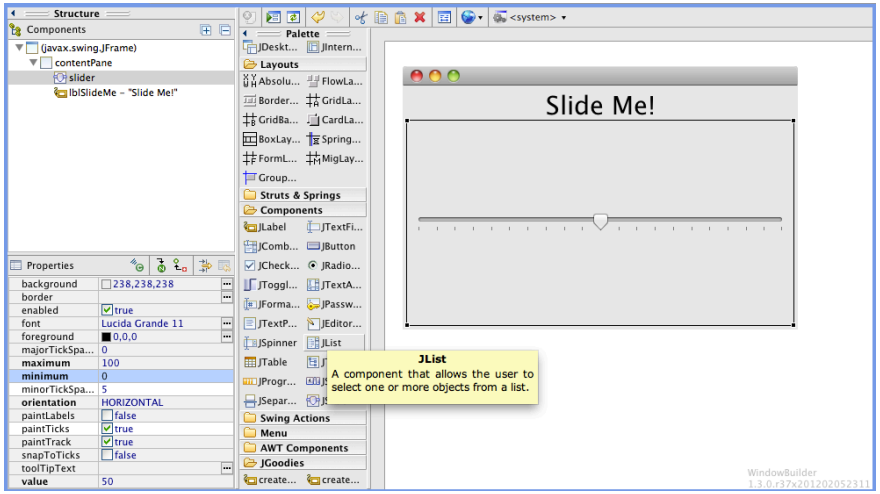


Designing GUIs in the Eclipse IDE



Designing GUIs in the Eclipse IDE

- Integrated Development Environment
  - Editor for managing complex programs
  - Often have debuggers
  - DrJava is simple example
- Eclipse is THE Java IDE
  - Use in later classes
  - Defacto IDE in industry
- Has tools for visual GUI design
  - Layout code generated automatically



Want to Learn More about GUIs

- Design a GUI by modifying an existing one
  - There are a lot of details; hard to get them all right
  - Easier when you are cribbing from elsewhere
  - Look at Assignment A6 (and A7)
- Lots of support on the PLive CD (Chapter 17)
  - That chapter shows you code for everything
  - Get the code from the CD and compile it yourself
- **Next Time**: How to make them do something