

GUI and Event Handling

CSC 116 – Section 002
April 20, 2005

© 2005 Sarah E. Smith

GUI Packages

- `java.awt`
 - Contains drawing and graphic manipulation classes
- `java.swing`
 - More advance GUI interface components
 - Components should work in a similar fashion on most platforms

© 2005 Sarah E. Smith

Window

- The Window is a JFrame object
- Set the size:
 - setSize(int width, int height);
- Set location (from top left corner):
 - setLocation(int horizontal, int vertical);
- Set title:
 - setTitle(String title);

© 2005 Sarah E. Smith

Window (2)

- The WindowListener - used to listen for any events (user input) and determines what to do for a given event
 - May be any object that extends WindowAdapter
 - like ActionListener
 - Assign WindowAdapter using the addWindowListener(WindowAdapter adapter) method

© 2005 Sarah E. Smith

Window (3)

- getContentPane() method of the JFrame class returns the container of the window
 - You cannot add components directly to the frame
- You will want to add your GUI components to the container - effectively adding the components to the board

© 2005 Sarah E. Smith

JPanel

- A container for multiple objects
- Layouts:
 - Flow - build one row at a time
 - Border - one item in each border area
 - Grid - column by row grid
- Constructor:
 - JPanel() //Flow Layout
 - JPanel(LayoutManager layout)

© 2005 Sarah E. Smith

JLabel

- Simple graphical object
- Contains text that is used as a label in the GUI
- This label can only be changed through the Java file, a user may not change the label by editing directly on the label.
- Constructor
 - JLabel(String text)

© 2005 Sarah E. Smith

JButton

- A push button
- Contains text or graphical representation of what the button does
- Constructor:
 - JButton()
 - JButton(String text)
- Set text:
 - setText(String text)
- Set enabled:
 - setEnabled(boolean enable)

© 2005 Sarah E. Smith

JTextField

- A single line text display or text input area
- Constructors:
 - JTextField()
 - JTextField(int columns)
 - JTextField(String text)
 - JTextField(String text, int columns)
- Set editable:
 - setEditable(boolean mayEdit)
- Set length (# of columns):
 - setColumns(int length)

© 2005 Sarah E. Smith

JTextArea

- A single line text display or text input area
- Constructors:
 - JTextArea()
 - JTextArea(int rows, int columns)
 - JTextArea(String text)
 - JTextArea(String text, int rows, int columns)
- Methods
 - setEditable(boolean mayEdit)

© 2005 Sarah E. Smith

JTextArea (2)

- Set length (# of columns):
 - `setColumns(int length)`
- Set wrap
 - `setWrapStyleWord(boolean words)` //true-wrap on space
 - `setLineWrap(boolean wrap)`//wrap at end of line

© 2005 Sarah E. Smith

JComboBox

- A pop-up menu that holds a finite number of choices for a user to select from
- Constructor
 - `JComboBox()`
 - `JComboBox(Object [] items)`
 - `JComboBox(Vector items)`

© 2005 Sarah E. Smith

JComboBox(2)

- Methods
 - addItem(Object item)
 - getItemAt(int index) //returns an Object
 - getItemCount() //returns an int
 - getSelectedItem() //returns an Object
 - removeAllItems()
 - removeItem(Object item)
 - removeItemAt(int index)

© 2005 Sarah E. Smith

Ways of Monitoring a Window

- ActionListener
 - monitors components inside the window
- WindowAdapter
 - monitors events on the window like closing and resizing
- Mouse Adapter and MouseMotionAdapter
 - monitors mouse events in the window

© 2005 Sarah E. Smith

ActionListener

- Any object that implements ActionListener
- Handles the results of an object action
 - Pressing a button
 - Selecting an item from a combo box
- We specify that we want the GUI to listen for an action on a certain component by calling the `addActionListener(ActionListener a)` on the component.
 - `btn.addActionListener(ActionListener a)`

© 2005 Sarah E. Smith

ActionListener (2)

- The `actionPerformed(ActionEvent e)` method is called when an action occurs on a component
- If more than one component has an ActionListener, you determine which component to execute code on by determining the `ActionEvent` that occurred.

© 2005 Sarah E. Smith

WindowAdapter

- To use the WindowAdapter use
addWindowListener(WindowAdapter a)
- windowActivated(WindowEvent e)
 - invoked when a window is activated
- windowClosed(WindowEvent e)
 - invoked when a window has been closed
- windowClosing(WindowEvent e) {
 - invoked when a window is in the process of closing

© 2005 Sarah E. Smith

WindowAdapter (2)

- windowDeactivated(WindowEvent e)
 - invoked when a window is deactivated
- windowDeiconified(WindowEvent e)
 - invoked when a window is deiconified
- windowIconified(WindowEvent e)
 - invoked when a window is iconified
- windowOpened(WindowEvent e)
 - invoked when a window is opened

© 2005 Sarah E. Smith

MouseEvent

- MouseEvent class contains information about a particular mouse event in the GUI
- MouseEvent.getX()
 - returns the x value of a selected coordinate in the GUI
- MouseEvent.getY()
 - returns the y value of a selected coordinate in the GUI

© 2005 Sarah E. Smith

MouseAdapter Interface

- Way to interact with the mouse in a GUI
- Set an object to use a MouseAdapter by addMouseListener(MouseListener m)
- mousePressed(MouseEvent e)
 - invoked when a mouse button is pressed on a component
- mouseClicked(MouseEvent e)
 - invoked when the mouse has been clicked on a component

© 2005 Sarah E. Smith

MouseAdapter (2)

- `mouseReleased(MouseEvent e)`
 - invoked when a mouse button has been clicked on a component
- `mouseEntered(MouseEvent e)`
 - invoked when the mouse enters a component
- `mouseExited(MouseEvent e)`
 - invoked when the mouse exits a component

© 2005 Sarah E. Smith

MouseMotionAdapter Interface

- Way to interact with mouse movements in a GUI
- Set an object to use a MouseAdapter by `addMouseListener(MouseListener m)`
- `mouseDragged(MouseEvent e)`
 - invoked when a mouse button is pressed on a component and dragged
- `mouseMoved(MouseEvent e)`
 - invoked when the mouse has been moved on a component (with no buttons on the mouse being pressed)

© 2005 Sarah E. Smith

References

- Jason Schwarz's Lecture 27 slides:
<http://courses.ncsu.edu/csc116/>