GUI Programming: Components

Slides derived from the work of Dr. Amy McGovern and Dr. Deborah Trytten
Basic JComponents

• JRadioButton: Radio buttons to select one item from a set of possibilities
• JCheckBox: Check box to select an item. Can have multiple checked at the same time
• JLabel: Display text
• JTextField: Enter/display a single line of text
Other GUI Components

• JTextArea:
  • Multiple lines of text
  • Can embed within JScrollPane to enable scrolling

• JComboBox
  • Dropdown list of items to choose from
  • Pre-validates input!
  • ActionEvent for new selections
  • ItemEvent for unselect/new select
JList

• Like a combo box but enables multiple selection
  • Single item
  • Single interval selection
  • Multiple interval selection
  • Also combines with JScrollPane

• Fires ListSelectionEvent
  • valueChanged() handler
Range Selection

JSlider

• Select a value from a range
• Sliding selection with a knob
Multiple Windows

• Use multiple JFrames

• Example:
  • Create a JFrame with a button on it
  • When button is clicked, a new window appears
  • (like a progress bar and ‘OK’ button)
Progress Bars

JProgressBar

• Provide feedback to a user about the progress of a file operation or a long computation

• Important for indicating that the program is busy (as opposed to frozen)
Creating Menus

Object hierarchy (has-a):

• JFrame
  • JMenuBar: single drop-down menu
    • JMenuItem: selectable item in a MenuBar

• JMenuItem actions:
  • Extend ActionListener
    • actionPerformed method with ActionEvent as a parameter
Pop-Up Dialog Boxes

Short conversations with users:
• Inform the user that an error has happened
• Prompt the user to answer a yes/no question
• Prompt the user for a file name

These windows are modal: they take the focus away from other windows
New Class: JOptionPane

One method:
showConfirmDialog(Component parentComponent, Object message)

• parentComponent: Use for placement of the dialog box. Can be null
• Message: Often a String containing the user prompt
New Class: JOptionPane

One method:

```
showConfirmDialog(Component parentComponent,
                   Object message,
                   String title,
                   int optionType,
                   int messageType)
```

• `optionType`: YES_NO_OPTION, YES_NO_CANCEL_OPTION, or OK_CANCEL_OPTION

• `messageType`: ERROR_MESSAGE, INFORMATION_MESSAGE, WARNING_MESSAGE, QUESTION_MESSAGE, or PLAIN_MESSAGE

• Return: Index number of selected option
New Class: JOptionPane

One method:

`showConfirmDialog(Component parentComponent, Object message, String title, int optionType, int messageType)`

• `optionType`: YES_NO_OPTION, YES_NO_CANCEL_OPTION, or OK_CANCEL_OPTION

• `messageType`: ERROR_MESSAGE, INFORMATION_MESSAGE, WARNING_MESSAGE, QUESTION_MESSAGE, or PLAIN_MESSAGE

• Return: Index number of selected option
File Selection

JFileChooser:

• Create instance of file chooser:
  ```java
  choose = new JFileChooser(new File("./data"))
  ```

• Sets default directory (here, the default is relative to the execution context)

• Note: does not automatically open dialog box
File Selection

Invoke the file chooser:

```java
int ret = choose.showOpenDialog();
```

- Select an existing file to open (there are similar method calls for other operations)
- Return is one of CANCEL_OPTION, APPROVE_OPTION, or ERROR_OPTION
- If APPROVE_OPTION, then the file can be retrieved by:

  ```java
  File file = choose.getSelectedFile();
  ```
Data Models

ListModel: List of objects to be displayed by a selection component (e.g., a JList)

• On construction, a JList will automatically create a ListModel from the array of objects that it is given

```java
JList jl = new JList({"A", "B"});
```
Data Models

• In many cases, the content of a JComponent, such as a JList stays constant: here, we use the default

• In other cases, the contents of the DataModel may change with time

• In this case, we must explicitly create a DataModel
For JList: DefaultListModel

• Model creation:

```java
DefaultListModel model =
    new DefaultListModel<String>();
```

• Can operate on this model as a list. For example:

```java
model.clear();
model.addElement("Foo");

JList jl = new JList(model);
```
DefaultListModel

• Every update to the list model will cause the JList to be repainted automatically
• Note that selections can change, too
More swing components


• Java documentation:
  • [http://docs.oracle.com/javase/tutorial/uiswing/components/componentlist.html](http://docs.oracle.com/javase/tutorial/uiswing/components/componentlist.html)