MiniDraw--Third Design Iteration

- Objective: To add toolbar, menu, and tools for unfilled geometric shapes and erasing.
- Swing Issues:
  - Will use Box class for ToolBar.
  - Will use JButtons for ToolBar buttons.
  - Will use JMenuBar and JMenu for menus
Problems with this Design Concept

• Which components need explicit knowledge of tools?
  – ToolBar
  – MenuBar
  – MenuBarListener
  – ToolBoxListener
  – DrawingCanvasListener
  – MiniDraw3

• High coupling
• To add a new tool, all of these classes must be modified

Problems with Design Concept--Continued

• ToolBar Listener and MenuBarListener perform nearly the identical function.
• A better design concept would:
  – allow the configuration of tools to be specified in a single place.
  – Combine ToolBar and MenuBar Listener functionality.
Moving Toward an Improved Design

• The Swing Action interface and AbstractAction class provide a neat mechanism for combining menu and toolbar event handling.

• An AbstractAction Object implements the ActionListener interface and also provides:
  – a name that can be used as a text label of a component
  – a set of values that can store various attributes such as an icon to be painted on the component, a “tool tip” string, and a description of the action.

Swing Action Objects--Continued

• When a Swing Action Object is added to a Swing container, the container:
  – creates a component that is appropriate for the container
  – retrieves appropriate attributes from the Action object to customize the component.
  – Sets the initial enabled or disabled state of the action object
  – renders the component
Still More about the Action Interface

- Action interface provides the following methods:
  - `putValue(key, value)` // set an attribute
  - `getValue(key)` // retrieve an attribute
  - `setEnabled(b)` // set enabled state
  - `isEnabled()` // test enabled state

- Key values:
  - `NAME` // e.g. text label
  - `DEFAULT` // e.g. icon
  - `SHORT_DESCRIPTION` // e.g. tool tip
  - `LONG_DESCRIPTION`
  - `SMALL_ICON`

Using Action Objects as Tool Listeners

Note: Action objects can be directly added to a JMenu as menu items.

Note: JButton adds ToolListener object as an ActionListener.
Behavior of a ToolListener

Design Question: What class should be responsible for keeping track of the currently selected tool?

Behavior of the DrawingCanvasListener

Behavior depends on currently selected tool.

Some tools may want to use additional event notifications.
Behavior of the DrawingCanvasListener--continued

DrawingCanvasListener needs to “delegate” behavior to the appropriate tool.

Solution: Use a state pattern:

Use of the State Pattern by DrawingCanvasListener