

# More of the Case Study

CS151

Chris Pollett

Nov. 7, 2005.

# Outline

- Iterations 2-4
- State Pattern

# More on Iteration2

- Recall the goal of iteration 2 was to add menus, dialogs, and to support saving and loading.
- Last day, we showed how menus could be set up with listeners.
- Let's see a couple dialog examples.

# About (message dialog example)

```
class AboutListener implements ActionListener
```

```
{
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        JOptionPane.showMessageDialog(null, "Drawing Pad version 1.0\nXiaoping Jia 2002", "About",  
JOptionPane.INFORMATION_MESSAGE);
```

```
        /*
```

```
            Null in first slot indicates no parent component.
```

```
            Last slot could also have JOptionPane.XXX where XXX is  
            one of
```

```
            ERROR_MESSAGE, INFORMATION_MESSAGE,  
            WARNING_MESSAGE, QUESTION_MESSAGE,  
            PLAIN_MESSAGE
```

```
        */
```

```
    }
```

```
}
```

# Open (JFileChooser example)

```
class OpenFileListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        int retval = chooser.showDialog(null, "Open");
        if(retval == JFileChooser.APPROVE_OPTION)
        {
            File theFile = chooser.getSelectedFile();
            if(theFile != null)
            {
                if(theFile.isFile())
                {
                    String fileName = chooser.getSelectedFile().getAbsolutePath();
                    openFile(fileName);
                }
            }
        }
    }
}
```

# Iteration 3: Refactoring

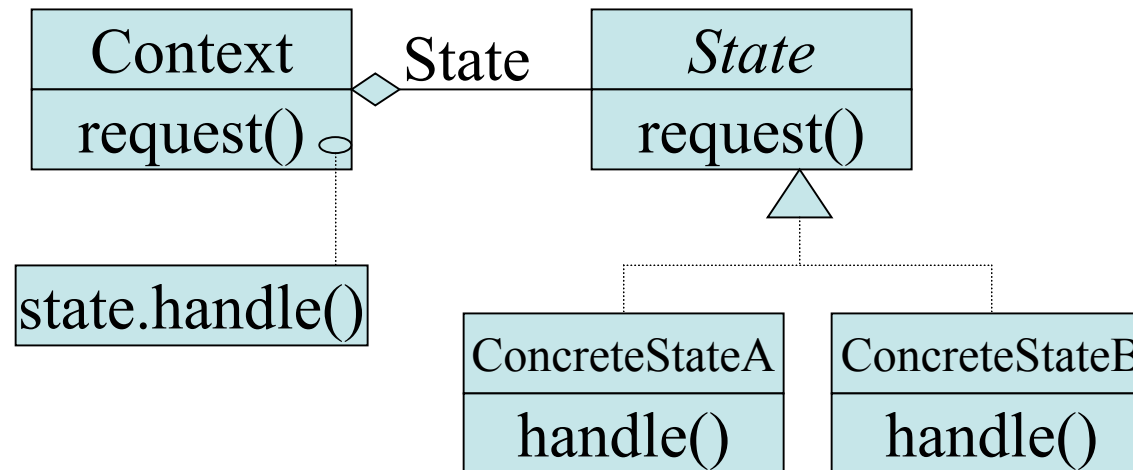
- The aim of this iteration is to refactor the drawing program so it can support other kinds of tools other than just scribbling.
- To do this a new abstract class Shape is created and Stroke is made a subclass of it.
- Shape supports setColor(), getColor(), and has an abstract method draw().
- The ScribbleCanvasListener is now modified so it has on it an object which supports the Tool interface.
- A Tool supports the methods getName(), startShape(Point p), addPointToShape(Point p), and endShape(Point p).
- This iteration makes an abstract class AbstractTool which has a constructor taking a ScribbleCanvas and a name and which implements getName.
- Finally, a concrete class ScribbleTool is implemented based on this.

# Iteration 4

- In this iteration several different kinds of Shape's are created in addition to Stroke, namely, TwoEndsShape which is extended by LineShape, RectangleShape, and OvalShape.
- Our JFrame is now called DrawingPad. It extends Scribble and has on it a Toolkit which can contain one or more tools. It also keeps track of the currently selected tool
- This JFrame has a button on it for each Tool. When this button is clicked. The Toolkit's selected Tool is changed and the DrawingCanvas' (which extends ScribbleCanvas) tool is changed.
- The use of Toolkit is an example of the State Pattern.

# State Pattern

- Category: Behavioral Design Pattern
- Intent: Allow an object to alter its behavior when its internal state changes
- AKA: Object for states
- Applicability: use State when (a) an objects behavior depends on its state and it must change its behavior at run time based on that state or (b) to avoid big switch statements based on object state.



- In Iteration 4, Context is the DrawingPad, which keeps track of a currentTool (current State). State in the above is the Tool interface and ScribbleTool is an example ConcreteState.