Events (from last time)

- Most conventional programs loop, waiting for input and processing it.
- Event-driven programs have methods triggered by the user.
  - Behind the scenes, it could be a loop.
Java Events (1.1 and after)

- Events are objects, from various classes.
- You create classes which are listeners.
- You instantiate objects of the listener class, then register each listener with the component which fires the event.
- addXXXListener()
- Each listener gets notified when the corresponding event occurs.

Listeners

- A listener must implement the right interface.
  - e.g. ActionListener
- Method in listener object wakes up with the event as its argument.
  - e.g. actionPerformed()

MyButtons.java

- Things to notice:
  - Events
  - Text Area manipulation
  - Layout
- This is a simplified program
  - e.g. Errors while resizing.
  - Separate class for each button.
Creating a JTextArea

JTextArea inText =
    new JTextArea("", 5, 25);
JScrollPane inScrollPane =
    new JScrollPane(inText);
add(inScrollPane);

Linking to a Button and Performing the Action

appendSelectedButton.addActionListener(
    new AppendSelectedListener());

class AppendSelectedListener
    implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        outText.append(
            inText.getSelectedText());
    }
}

[Gets the currently selected text from inText and appends it to outText.]
For all Components, you can:
- setForeground(Color n)
- setBackground(Color n)

An object of class Color specifies a color.

Many constructors
- Color(int red, int green, int blue)
- red, green, blue in the range 0-255.

Rotate Colors Example
(Initial Conditions)

rotateColorsButton.setForeground(new Color(0, 64, 128));
rotateColorsButton.setBackground(new Color(128, 192, 0));
Button Listener

```java
class RotateColorsListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        Color oldColor, newColor;
        // Rotate the foreground color
        oldColor = rotateColorsButton.getForeground();
        newColor = rotateColors(oldColor);
        rotateColorsButton.setForeground(newColor);
        // Rotate the background color
        oldColor = rotateColorsButton.getBackground();
        newColor = rotateColors(oldColor);
        rotateColorsButton.setBackground(newColor);
    }
}
```

rotateColors() Method

```java
Color rotateColors(Color oldColor) {
    Color newColor;
    int red, green, blue; // hold the color components
    red = oldColor.getRed();
    green = oldColor.getGreen();
    blue = oldColor.getBlue();
    red = (red + 64) % 256;
    green = (green + 64) % 256;
    blue = (blue + 64) % 256;
    newColor = new Color(red, green, blue);
    return newColor;
}
```
What Happens If We Resize The Window?

- Default LayoutManager for applets is FlowLayout.
- Components are added left-to-right until you run out of window, then they continue adding on the next line.

Voila!

Next Time

- Multi-Tasking and Threads