`BreezySwing`: A Lightweight Toolkit for the Construction of Easy, Realistic GUIs in CS1

Contact Information
Kenneth A. Lambert, Washington and Lee University, klambert@wlu.edu
Martin Osborne, Western Washington University, martin@cs.wwu.edu

Problem Statement

Beginning students find it difficult to write realistic, event-driven programs with GUIs using just `java.awt` and `javax.swing`. One challenge is to find a solution that avoids distortions of GUI-based programming, such as the use of a series of simple modal dialogs for inputs. Another challenge is to provide an API that is lightweight enough that it both resembles "real Java" and can be easily trimmed away when students reach maturity.

Solution Overview

`BreezySwing`, like its predecessor, `BreezyGUI` [Lambert99, Lambert00], is a package of classes that allows students to construct easy, realistic GUIs. The package includes container classes that are extensions of `JFrame`, `JDialog`, and `JApplet`. Each container class encapsulates the details of a gridbag layout, allowing the student to place many types of standard components by specifying simple row and column positions, a width in columns, and a height in rows. In addition, each container class encapsulates the details of Java's event handling mechanism by including single methods for responding to button clicks, menu item selections, and list box selections. The classes `DoubleField` and `IntegerField` are extensions of `JTextField` that hide the details of numeric I/O. Finally, the class `GBPanel` extends `JPanel` by including methods that simplify the handing of mouse events in graphics applications. The use of `BreezySwing` focuses students' attention immediately on the idea that a user interface is built from a collection of objects that cooperate by sending messages. Students design an interface by choosing widget/objects of the appropriate types and quickly laying them out in a container. The rest of their time can then be focused on designing and implementing the algorithms that respond to events in the user interface and the code that manages the data in the application's model.

Experience with the Solution

We have used `BreezySwing` in CS1 and CS2 classes for four years. We have published two editions of a CS1 college textbook that uses `BreezySwing` and has been adopted by many institutions around the world. We have also published a widely adopted AP textbook that uses `BreezySwing`. We have heard that the package has been used in classes where the textbooks were not adopted. Users' reports of `BreezySwing` in the first course have been uniformly positive. We have heard some complaints from cognoscenti who are not users that the details of event handling (the listeners) should not be encapsulated, but no actual users in the first course have agreed with this criticism. We also included `BreezySwing` in the first edition of a CS2 book. Several users of this book complained that they had trouble weaning their students away from the package by the end of the course. To address this problem, we included a lighter weight version of `BreezySwing`, called `ioutil`, in the second edition of the CS2 book.

API Documentation

Here is an example of an interactive counter applet with `BreezySwing`:

```
import BreezySwing.*;
import javax.swing.*;

public class CounterApp extends GBApplet{

   private int counter = 0;

   private IntegerField counterField = addIntegerField(0, 1,1,2,1);
   private JButton incButton = addButton("Increment",     2,1,1,1);
   private JButton resetButton = addButton("Reset",       2,2,1,1);

   public void buttonClicked(JButton buttonObj){
      if (buttonObj == incButton)
         counter++;
      else
         counter = 0;
      counterField.setNumber(counter);
   }
}
```

To convert this code to a Java application, one just replaces `GBApplet` with `GBFrame` and includes an appropriate `main` method.

Method calls to add components to a container have the following form:

```
add<type>(<initial value>, <row>, <col>, <width>, <height>)
```

Menu items are added and returned by the following type of method call:

```
addMenuItem(<string name of menu>,  <string name of item>)
```

In addition to `buttonClicked`, the following single methods can be overridden to respond to user events:

```
public void menuItemSelected(JMenuItem mi)
public void listitemSelected(JList listObj)
public void listDoubleClicked(JList listObj, String itemClicked)
```

All of the standard `MouseListener` and `MouseMotionListener` methods are encapsulated in the `GBPanel` class, which essentially wraps their behavior in the more easily used methods

```
public void mouseClicked(int x, int y)
public void mouseMoved(int x, int y)
```

etc.

Complete documentation and downloads of `BreezySwing` are available at www.cs.wwu.edu/martin, scroll down to “Open source software packages.”