

Example programs showing interactor usage

File: InteractiveDrawFace.java

```
/*
 * File: InteractiveDrawFace.java
 * -----
 * This program draws GFaces on the screen, but allows the
 * use to modify their size and color.
 */

import acm.program.*;
import acm.graphics.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class InteractiveDrawFace extends GraphicsProgram {

    public void init() {
        // Button to clear display
        add(new JButton("Clear"), SOUTH);

        // Check box to display front or back of face
        checkbox = new JCheckBox("Front");
        checkbox.setSelected(true);
        add(checkbox, SOUTH);
        initRadioButtons();
        initColorChooser();
        // Must call this method to be able to get mouse events
        addMouseListeners();
        // Must call this method to get button press events
        addActionListeners();
    }

    private void initRadioButtons() {
        // Radio button group for size
        ButtonGroup sizeBG = new ButtonGroup();
        smallRB = new JRadioButton("Small");
        medRB = new JRadioButton("Medium");
        largeRB = new JRadioButton("Large");
        // Add all radio buttons to button group
        sizeBG.add(smallRB);
        sizeBG.add(medRB);
        sizeBG.add(largeRB);
        // Set initial radio button selection
        medRB.setSelected(true);
        // Add all radio buttons to control bar
        add(smallRB, SOUTH);
        add(medRB, SOUTH);
        add(largeRB, SOUTH);
    }
}
```

```

private void initColorChooser() {
    // Create label (with separating spaces) for combo box
    add(new JLabel("  Color:"), SOUTH);
    // Create combo box with color choices
    pickColor = new JComboBox();
    pickColor.addItem("Black");
    pickColor.addItem("Blue");
    pickColor.addItem("Green");
    pickColor.addItem("Red");
    // Don't allow user to type in a color
    pickColor.setEditable(false);
    // Set initial color selection
    pickColor.setSelectedItem("Black");
    // Add combo box to control bar
    add(pickColor, SOUTH);
}

// Returns diameter size corresponding to radio button choice
private double getDiamSize() {
    double size = 0;
    if (smallRB.isSelected()) {
        size = SMALL_DIAM;
    } else if (medRB.isSelected()) {
        size = MED_DIAM;
    } else if (largerRB.isSelected()) {
        size = LARGE_DIAM;
    }
    return size;
}

// Returns Color object corresponding to combo box choice
private Color getCurrentColor() {
    String name = (String) pickColor.getSelectedItem();
    if (name.equals("Blue")) {
        return Color.BLUE;
    } else if (name.equals("Green")) {
        return Color.GREEN;
    } else if (name.equals("Red")) {
        return Color.RED;
    } else return Color.BLACK;
}

// Called every time user clicks mouse
public void mouseClicked(MouseEvent e) {
    GObject obj;
    double diam = getDiamSize();
    if (checkbox.isSelected()) {
        obj = new GFace(diam, diam);
    } else {
        obj = new GOval(diam, diam);
    }
    obj.setColor(getCurrentColor());
    add(obj, e.getX(), e.getY());
}

```

```

// Called whenever an action event occurs
public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals("Clear")) {
        removeAll(); // Clears the canvas
    }
}

/* Private constants */
private static final double SMALL_DIAM = 20;
private static final double MED_DIAM = 40;
private static final double LARGE_DIAM = 60;

/* Private instance variables */
// Use instance variables to keep track of interactors whose
// "state" you need to check as your program runs
private JCheckBox checkbox;
private JRadioButton smallRB;
private JRadioButton medRB;
private JRadioButton largeRB;
private JComboBox pickColor;
}

```

File: GFace.java

Although you have previously seen the code for `GFace.java`, we include it again here for completeness of the program above.

```

/*
 * File: GFace.java
 * -----
 * This class implements a face as a GCompound.
 */

// Note: only need acm.graphics since this is not
// actually a program, but just a class using graphics.
import acm.graphics.*;

/** Defines a compound GFace class */
public class GFace extends GCompound {

    /* Constants specifying feature size as a fraction of head size */
    private static final double EYE_WIDTH    = 0.15;
    private static final double EYE_HEIGHT   = 0.15;
    private static final double NOSE_WIDTH   = 0.15;
    private static final double NOSE_HEIGHT  = 0.10;
    private static final double MOUTH_WIDTH  = 0.50;
    private static final double MOUTH_HEIGHT = 0.03;

    /* Private instance variables */
    private GOval head;
    private GOval leftEye, rightEye;
    private GPolygon nose;
    private GRect mouth;
}

```

```

/** Creates a new GFace object with the specified dimensions */
public GFace(double width, double height) {
    head = new GOval(width, height);
    leftEye = new GOval(EYE_WIDTH * width, EYE_HEIGHT * height);
    rightEye = new GOval(EYE_WIDTH * width, EYE_HEIGHT * height);
    nose = createNose(NOSE_WIDTH * width, NOSE_HEIGHT * height);
    mouth = new GRect(MOUTH_WIDTH * width, MOUTH_HEIGHT * height);

    add(head, 0, 0);
    add(leftEye, (0.25 * width) - (EYE_WIDTH * width) / 2,
        (0.25 * height) - (EYE_HEIGHT * height) / 2);
    add(rightEye, (0.75 * width) - (EYE_WIDTH * width) / 2,
        (0.25 * height) - (EYE_HEIGHT * height) / 2);
    add(nose, 0.50 * width, 0.50 * height);
    add(mouth, (0.50 * width) - (MOUTH_WIDTH * width) / 2,
        (0.75 * height) - (MOUTH_HEIGHT * height) / 2);
}

/* Creates a triangle for the nose */
private GPolygon createNose(double width, double height) {
    GPolygon poly = new GPolygon();
    poly.addVertex(0, -height / 2);
    poly.addVertex(width / 2, height / 2);
    poly.addVertex(-width / 2, height / 2);
    return poly;
}
}

```

File: TextFieldExample.java

```

/* File: TextFieldExample.java
 * -----
 * This class displays a greeting whenever a name is entered
 */

import acm.program.*;
import java.awt.event.*;
import javax.swing.*;

public class TextFieldExample extends ConsoleProgram {

    public void init() {
        nameField = new JTextField(10);
        add(new JLabel("Name"), SOUTH);
        add(nameField, SOUTH);
        nameField.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == nameField) {
            println("Hello, " + nameField.getText());
        }
    }

    /* Private instance variables */
    private JTextField nameField;
}

```