

Graphics and Events Examples

Based on examples by Eric Roberts.

File: GFace.java

```
/*
 * 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.*;

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;

    /** 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;
    }

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

}

File: DrawFace.java

```
/*
 * File: DrawFace.java
 * -----
 * This program draws a GFace in the middle of the screen.
 */

import acm.program.*;
import acm.graphics.*;

public class DrawFace extends GraphicsProgram {

    /** Width of face */
    private static final int FACE_WIDTH = 100;

    /** Height of face */
    private static final int FACE_HEIGHT = 200;

    public void run() {
        GFace face = new GFace(FACE_WIDTH, FACE_HEIGHT);
        add(face, (getWidth() - FACE_WIDTH) / 2,
                (getHeight() - FACE_HEIGHT) / 2);
    }
}
```

File: ClickForFace.java

```
/*
 * File: ClickForFace.java
 * -----
 * This program displays a face in every location the user
 * clicks on. It is an example of an event-driven program.
 */

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

public class ClickForFace extends GraphicsProgram {

    /** Private constants */
    private static final double FACE_DIAM = 30;

    // Note: no run() method in this program

    // init() method is called when program starts
    public void init() {
        // Must call this method to be able to get mouse events
        addMouseListeners();
    }

    // This method is called everytime user clicks mouse
    public void mouseClicked(MouseEvent e) {
        GFace face = new GFace(FACE_DIAM, FACE_DIAM);
        add(face, e.getX(), e.getY());
    }
}
```

}

File: MouseTracker.java

```
/*
 * File: MouseTracker.java
 * -----
 * This program displays the (x, y) location of the mouse.
 */

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

public class MouseTracker extends GraphicsProgram {

    public void run() {
        label = new GLabel("");
        label.setFont("Times New Roman-36");
        add(label, 50, 50);

        // Must call this method to be able to get mouse events
        addMouseListeners();
    }

    // This method is called everytime user moves mouse
    public void mouseMoved(MouseEvent e) {
        label.setLabel("Mouse: (" + e.getX() + ", " + e.getY() + ")");
    }

    /* Private instance variable */
    private GLabel label;
}
```

File: DragObjects.java

```

/*
 * File: DragObjects.java
 * -----
 * Example program to show mouse and keyboard interactions.
 * This program allows us to move objects on the screen
 * by dragging then with the mouse. We can also change the
 * color of the last object moved to a random color by typing a key.
 */

import acm.graphics.*;
import acm.program.*;
import acm.util.*;
import java.awt.*;
import java.awt.event.*;

/** This class displays a mouse-draggable rectangle and oval */
public class DragObjects extends GraphicsProgram {

    // Initializes the program
    public void init() {
        GRect rect = new GRect(100, 100, 150, 100);
        rect.setFilled(true);
        add(rect);
        GOval oval = new GOval(50, 50, 150, 100);
        oval.setFilled(true);
        add(oval);
        addMouseListeners();
        addKeyListeners();
    }

    // Called on mouse press to record the coordinates of the click */
    public void mousePressed(MouseEvent e) {
        // GPoint has X and Y coordinate
        last = new GPoint(e.getPoint());
        gobj = getElementAt(last);
    }

    // Called on mouse drag to reposition the object
    public void mouseDragged(MouseEvent e) {
        if (gobj != null) {
            gobj.move(e.getX() - last.getX(), e.getY() - last.getY());
            last = new GPoint(e.getPoint());
        }
    }

    // Change color of last object dragged
    public void keyTyped(KeyEvent e) {
        if (gobj != null) {
            gobj.setColor(rgen.nextColor());
        }
    }

    /* Private instance variables */
    private GObject gobj;          /* The object being dragged */
    private GPoint last;           /* The last mouse position */
    private RandomGenerator rgen = RandomGenerator.getInstance();

```

}