Programming Structures and Abstractions (CS 2334) Lab 5: Graphical User Interface Basics

October 14, 2009

Due: Monday, October 19, 2009, 11:29am

Group members (same as for your project):

Objectives

The objectives of this lab are to:

1. analyze and extend the functionality of a java program,
2. create and manipulate basic graphical user interface components, including JPanel, JLabel, GridLayout and TitledBorder
3. display Finch sensor data in a graphical user interface, and
4. use information entered into the GUI to affect the state of the Finch.

Problem Context

In this project, we will be creating a graphical user interface (GUI) to interface with the Finch. In particular, this interface will display the current light, obstacle, temperature and accelerometer values. In addition, the user will be able to control the color of the nose LED using a set of slider bars.

The sample code provides a basic skeleton for creating a window that can display light sensor values and accept input from slider bars. Specifically, you will:

- analyze the structure of the existing code base,
- use the slider bar data to change the state of the LEDs, and
- add display elements for the acceleration, obstacle, and temperature sensors.

Milestones

Milestone 1: Analyze Existing Program

Compile and run the lab5 program that has been provided. Briefly explain what happens in the GUI when you move the Finch around:

The **FinchDisplay()** constructor is responsible for creating an instance of the class. This class extends the **JFrame** class, and therefore inherits a lot of functionality. Because JFrame is a **container**, it provides an **add()** method that allows one to add graphical child objects to the frame (it adds in the "has-a" sense). Other classes, such as **JPanel**, are also containers.

Given the FinchDisplay() constructor, draw on engineering paper the FinchDisplay instance and all of its children. Do this recursively (so the childrens' children, etc.). Note 1: you only have to show the child instances that are created in the FinchDisplay() constructor. Note 2: although we are using the "parent" and "child" terminology here, it refers to a "has-a" relationship from parent to child, and not to an inheritance relationship.

Milestone 2: Use the Slider Bars to Change LED State

The FinchDisplay class includes a private property called **LED**. This array of integers stores the LED brightness values that are "intended" by the user. Given the implementation of the constructor, briefly explain your intuition about how LED[1] is changed (you should not have to look at Chapter 15 to answer this):

The FinchDisplay class provides an accessor method to this private property, **getLED(),** which is used by the driver class. However, nothing is currently done by the driver with this information. Add the necessary code to the driver that will take the value obtained from getLED() and change the state of the Finch nose.

Note: get this working before you move on to the next milestone.

Milestone 3: Add Other Sensors

Add the necessary code so that the state of the following sensors is also shown in the GUI:

- acceleration,
- obstacle, and
- temperature.

Following the implementation of the light sensors, add the following components for each sensor:

- (FinchDisplay) a separate JPanel with an appropriate TitledBorder and GridLayout that contains JLabels for each of the values to be displayed,
- (FinchDisplay) a mutator method that allows the driver class to update the values to be stored, and

- (driver) a read of the appropriate sensor from the Finch and a write of the sensor value to the FinchDisplay.

What to Hand In

All materials are due: Monday, October 19, 2009, 11:29am

Hand in the following:

- a hard copy of your instance diagram,
- a copy of this handout containing the provided answers, and
- an electronic copy of your modified code (to D2L).

NOTE: ONLY HAND IN ONE COPY PER GROUP.

In addition to handing in a copy of the code, you must do a short demonstration of your working code for the TA or the instructor. Ideally you will do this before the end of the lab period. Otherwise, please make an appointment before the deadline. All group members should be in attendance during the demonstration.