Programming Structures and Abstractions (CS 2334)
Lab 7: Model View Controller Paradigm

November 3, 2009

Due: Tuesday, November 10th, 2009, 5:00 pm

Group members (same as for your project):

Objectives

The objectives of this lab are to:

1. use Events and EventListeners to connect a viewer to a model, and
2. update the display of a window in response to an Event.

Problem Context

For this lab, we will be completing an implementation of a Finch sensor monitoring program that uses the Model-View-Controller framework. The model will represent both the most recently sensed light levels and temperature. Any time either of these values are updated, the model will inform any viewer that the update has occurred. The viewers will respond by grabbing the current sensor values (stored in the model) and updating their displays.

The sample code provides a basic skeleton for creating a FinchModel and a set of windows that can display sensor values and respond to user events.

Specifically, you will:

- add missing functionality to the FinchModel class that will result in a properly-formed model (in the MVC sense),
- add missing functionality to the LightWindow class that will result in a properly-formed viewer,
- add missing functionality of the TemperatureWindow class, and
- use the information stored in the ActionEvent to update only the window for which the information has changed.

Milestones

Milestone 1: Make FinchModel a Proper Model

There are two missing components from the FinchModel class. Add them.

Milestone 2: Make LightWindow a Proper Viewer

A LightWindow instance is a JFrame that contains the current sensory information for both the left and right light sensors. There are three missing components that are required to make this class a proper viewer in the MVC sense and to display the light sensor information. Add these components.

Milestone 3: Make TemperatureWindow a Proper Viewer

This JFrame displays the current state of the temperature sensor. Complete the implementation.

At this stage, you will have a complete and working program.

Milestone 4: Focused Updates

In the above implementation, the Light and Temperature Windows will update any time some part of the model changes, even if it does not correspond to the information that they are displaying. Alter your implementation so that the windows are updated (and the println is called) **ONLY** when their respective part of the model is updated.

Hints

- You may have to modify how the ActionEvents are created.
- One possible approach is to alter the implementations of updateLights() and updateTemperature().

What to Hand In

All materials are due: Tuesday, November 10th, 2009, 5:00pm.
Note: getting this lab done before the exam will help you study for it.

Hand in the following:

- an electronic copy of your modified code (to D2L), and
- include a note at time of hand-in as to which group members participated in the lab.

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.