CMSI 370

INTERACTION DESIGN

Fall 2006

Assignment 1019

This assignment exercises a little bit of everything: Swing layout, programmatic construction of user interfaces, some MVC separation, and event handling.

For Submission

- Create a customizable text data entry panel called *DataEntryPanel* (duh) that behaves in the following way: you initialize the panel with a sequence of Strings that represents the names of the text fields that the panel should present. For example, the code fragment below should result in a Swing panel with 2 columns and 4 rows: the first column displays the provided prompts, and the second column displays JTextFields for each requested field.

  ```
  DataEntryPanel dep =
    new DataEntryPanel("Last Name",
      "First Name", "Phone", "Birthdate");
  ```

  Tip: You might want to look up the *SpringLayout* layout manager for an easy way to put together the components of this panel.
- Write a demonstration program for this panel. Give it the following buttons: one or more "Set Data" buttons that programmatically set the data that the panel should display, and one "Display Data" button that reads the data that is currently in the panel, then shows the current contents of the data entry panel. Console output is OK for "Display Data," but if you can do it with additional Swing components, then why not.

Commit this program to CVS, under */homework/cmsi370/dataentry*. As usual, include an Ant *build.xml* file, and tag the submission as *hw-1019*.

Extra Credit

Do the regular assignment above, and add the following features to your DataEntryPanel class:

- Add a method that will allow the data entry panel to change the fields that it displays *dynamically*. Specifically, implement this method:

```
public void setFields(String... fieldNames)
```

- Callling this method should rebuild the panel from the ground up, so that it now shows the new set of data entry fields. Look up *java.awt.Container* (a superclass of JPanel) for some hints on how to do this, and you will also want to read up on how to use *revalidate()* and *repaint()* to tell a panel to refresh itself.
- Make sure to modify your demonstration program to add buttons that show *setFields()* at work.

If your *hw-1019* files in */homework/cmsi370/dataentry* include this functionality, then you will get the extra credit.