

# CMSI 370

## INTERACTION DESIGN

Fall 2006

### Assignment 0914

This assignment is meant to get your feet wet with Swing and for you to learn CVS hands-on (if you don't already know it).

#### Not for Submission

1. Read Chapter 2 from Shneiderman/Plaisant.
2. Read Chapter 5 from Nielsen.

#### For Submission

Putting together Swing user interfaces takes a lot of practice, particularly in the pure layout/positioning area. The kind of programming specified in this assignment should eventually become second nature for you, because in the long run, layout/positioning will be the least of your interaction design problems!

#### What to Do

Choose at least three (3) user interface displays of sufficient complexity from existing software, and replicate their look and layout with Swing. While there are no hard rules for “sufficient complexity,” these characteristics can serve as a guide:

- At least four (4) distinct types of atomic components (i.e., labels, buttons, text fields, check boxes, radio buttons, sliders, menu items, etc.)
- At least ten (10) actual components (e.g., 3 buttons, 4 labels, 2 text fields, and 2 check boxes)
- Genuine 2-dimensional layout — so, no toolbars
- Multipanel interface (e.g., tabs, master-detail, previous-next)

Good candidates include: preference, configuration, or setup dialogs; non-trivial data entry windows; control panels; instrumentation displays. When in doubt, show me the interface and I can tell you if it's complex enough.

Submit each of your “facsimiles” as self-contained source code trees with Java source starting at *src/* and an Ant *build.xml* script at the top-level directory. Your *build.xml* must have at least these targets: *compile*, for (duh) compiling the code; *run*, to invoke your main method; and *clean*, for restoring the directory to its “pristine” state.

#### How to Turn it In

Since I will access your work electronically, you must follow the instructions below *to the letter, down to the capitalization*. Many of these steps are for initial setup only, and subsequent submissions won't be quite so involved:

1. Your Keck lab account already comes with two CVS subdirectories, *projects* and *homework*.
2. Caskey has prepared a script for setting up your CVS directories so that I can read/write them — ask him how to get to it, and run it.
3. On any computer that you will use for your school work, check out *homework*.
4. Under your local copy of *homework*, create a *cmsi370* subdirectory.

The above steps should be one-time only; now for the stuff that's specific to this assignment.

5. Under the checked-out *homework/cmsi370* directory, create a *facsimiles* subdirectory.
6. Place each of your facsimile programs in its own subdirectory under *homework/cmsi370/facsimiles*. For best results, stick with all-lower case directory names, without spaces.
7. Add then commit the files to CVS. When in doubt,  *cvs update* to verify file statuses.
8. Tag the files in CVS as *hw-0914*.

For example, if you decided to name your 3 facsimiles *explorer*, *thunderbird*, and *gaim*, respectively, then your final CVS depot tree will look like this:

```
.cvs/homework/cmsi370/facsimiles
  explorer/
    build.xml
    src/
  thunderbird/
    build.xml
    src/
  gaim/
    build.xml
    src/
```