Assignment 0616

Sorry for not writing this up quite on time; hope one day doesn’t make a huge difference. Anyway, this assignment hopes to exercise simultaneously some of that threading and custom component knowledge that you’ve acquired. Not everyone has access to a decent profiler, so it’s hard to exercise the memory management component of our material — for this, I recommend going through Dr. Toal’s notes and sample program (see below).

Not For Submission

The topics that we covered for this assignment have many supplemental readings available on the Web, and they are all highly recommended if you want further depth on the topics:

1. *Memory leaks in Swing* — Dr. Toal has talk notes and sample code showing a similar type of Swing memory leak at this URL:
   http://www.technocage.com/~ray/talks/swing.html

2. *Threads and Swing* — The Swing Connection has the core set of articles on the subject, including code for SwingWorker:
   http://java.sun.com/products/jfc/tsc/articles/threads/threads2.html

3. *Custom components* — The “official” recipe for creating a custom Swing component can be found in the Swing Tutorial site:
   http://java.sun.com/docs/books/tutorial/uiswing/14painting/index.html

For Submission

Implement a “stopwatch” or “counter” program that consists of the following elements:

1. A custom Swing component that represents the stopwatch or counter
2. A Start button that, when clicked, initiates the stopwatch or counter and starts updating the custom Swing component with the latest time/number/status
3. A Stop button that, when clicked, will stop the count

Be as creative as you wish about the custom component; the only restriction is that you should paint this component yourself, without resorting to the canned component library. Also, a properly operating program will have some degree of threading, or else you won’t see your component update, nor will you be able to click on the Stop button!

Feel free to add any touches that make your program adhere to the various HCI guidelines or principles that we have seen in class. One such behavior comes to mind immediately: I’ll be curious to see if any of you think of it also (and implement it), or if you think of anything else that improves this simple program’s usability.

As usual, e-mail the code to me before class on June 16.