CMSI 370
INTERACTION DESIGN
Fall 2007

Final Review Sheet

The final exam will take place on Tuesday, December 11, at 11am. It will be open book, notes, handouts, and yes, this time, open computer — so we'll have it at the Keck lab. You can use one computer, either your own laptop or one of the Keck workstations.

Covered Material

The final covers the entire semester, including all handouts and sample code that have been distributed; in addition to the covered material from the midterm (see the Midterm Review Sheet for that), we also have:

- Chapters 6, 7, and 8 in Shneiderman/Plaisant
- Parts 3 and 4 in Norman
- User interface API topics
  - Higher-order Swing components: JTables, JLists, renderers, editors
  - Externalization and internationalization
  - Swing custom and reusable components
  - Swing menus, toolbars, and Actions
  - Low-level event handling (mouse, keyboard)
  - Drag-and-drop (Swing, plus general principles regardless of the API)
  - Swing threading and memory management
  - Overview knowledge of XHTML, CSS, and JavaScript development
  - Overview knowledge of GLUT

Sample Tasks and Questions

The following represent the types of questions or tasks that you may be asked to accomplish (in addition to those listed in the Midterm Review Sheet):

- Given a proposed application, choose and justify the most appropriate interaction style/paradigm for that application
- Given two alternative user interfaces based on the same interaction style/paradigm for the same application, state which user interface is more effective and why
- Assess the quality of a menu-driven user interface based on the guidelines and principles for creating effective menus and dialogs
- Assess the quality of a direct manipulation user interface based on the guidelines and principles for creating effective direct manipulation designs
- Assess the quality of a command-line user interface based on the guidelines and principles for creating effective commands
- Given some Swing code, evaluate its design based on its reusability and separation of concerns (MVC, etc.)
- Explain or describe some frequently-used Swing technique, such as externalization of resources, creation of custom components, eliminating memory leaks, etc.
- Break down a user action (drag-and-drop, text field manipulation, button tracking, etc.) into a sequence of low-level input events
- Identify bugs or potential issues in some given fragment of Swing code
- Identify, describe, or compare the similarities or differences among Swing, XHTML/CSS/JavaScript, and GLUT user interface frameworks