

CMSI 370

INTERACTION DESIGN

<http://myweb.lmu.edu/dondi/fall2006/cmsi370>

Fall 2006 — Doolan 219
TR 10:50am–12:05pm, 3 semester hours
Office Hours: TR 3–6pm or by appointment

John David N. Dionisio, PhD
e-mail: dondi@lmu.edu, AIM: dondi2LMU
Doolan 106; (310) 338-5782

Course Objectives

To learn the art and science of interaction design, and to learn how to program user interfaces. Students will be exposed to the first principles and metrics behind human-computer interaction, while gaining working knowledge of user interface technologies such as Java's Swing, OpenGL's GLUT, and XHTML/CSS/JavaScript.

Course Requirements

Mastery of object-oriented programming; expert knowledge of data structures and algorithms; willingness to participate in class discussions. Some familiarity with computer system organization, operating systems, and Web technologies is helpful.

Materials and Texts

- Ben Shneiderman and Catherine Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Fourth Edition, Addison Wesley, 2004.
- Jakob Nielsen, *Usability Engineering*, Morgan Kaufmann, 1994.
- Donald A. Norman, *The Design of Everyday Things*, Basic Books, 2002.
- Assorted handouts, articles, and sample code to be distributed throughout the semester.

Additional information is also available on the Web; do not hesitate to look for further sources of information regarding the concepts, techniques, tools, and paradigms that we will discuss.

Course Work and Grading

Course work consists of homework (20%), 1 midterm (20%), 1 design poster (10%), 1 programming portfolio (25%), and 1 final exam (25%). Letter grades are determined as follows: $\geq 90\%$ gets an A– or better; $\geq 80\%$ gets a B– or better; $\geq 70\%$ gets a C– or better. I may curve grades upward based on qualitative considerations such as degree

of difficulty, effort, class participation, time constraints, and overall attitude throughout the course. Grades are never curved downward.

Homework

Homework consists of questions, exercises, and programming assignments to be given throughout the semester. Homework is where you can learn from your mistakes without grading penalty: if you do the work and submit it on time, you will get full credit, regardless of correctness. What goes around comes around: the effort you put into your homework pays off in the tests and the portfolio. The homework submission deadline is always the beginning of class on the designated due date; the due date is encoded in the homework number. Submissions after the deadline receive half credit, period. Extra credit homework may be assigned; fulfilling this is counted on top of the 20% allocation of homework to your final grade.

Tests

The midterm is initially scheduled for October 5. The final exam is scheduled for December 12. All tests are open-paper-everything; no sharing. “Open computer” might also be allowed depending on the scope, subject matter, or circumstances. You may neither solicit nor give help while the exam is in progress. Late and/or missed tests are handled on a case-to-case basis; in all instances, talk to me about them.

Design Poster

Some of your homework assignments will involve original user interface designs. Near the end of the semester, one of these design exercises will be re-rendered as a poster, for presentation and display somewhere in Doolan Hall. If possible, we will try to arrange for a poster session where you can discuss your designs with faculty, colleagues, and friends. The poster is graded as credit/no-credit (like homework), and is due on December 5. Late posters will not be accepted.

Programming Portfolio

Some assignments involve programming, and when submitted as homework, will be graded as homework (i.e., credit/no credit). However, at the end of the semester, you will be asked to resubmit some of these assignments in a *programming portfolio* — a showcase of sorts for your newfound interaction design skills. *This* will be graded more closely; presumably, by semester's end, you will know this stuff better, and will be able to improve your prior work. The code will be graded along these criteria:

1. *Design (30%)*: How good is the overall structure of the code? Is it clear, flexible, and easy to maintain? Is it elegant or innovative? How well does it apply the principles of “separation of concerns” and “one change, one place?”
2. *Functionality (30%)*: How well does the code work? Does it fulfill requirements? Are its results accurate or correct? How fast does it run? How well do unit tests validate the code?
3. *Naming (20%)*: Are program entities — classes, subroutines, variables, etc. — clearly and consistently named? Do their names correspond to their functions and roles?
4. *Comments (15%)*: Are comments provided where appropriate? Are they clear and well-written? Does the code take advantage of any special support for comments provided by the project language or platform (e.g., JavaDoc)?
5. *CVS use (5%)*: Is the code committed at reasonable intervals? Are milestones appropriately tagged? Are adequate descriptions provided in the commit logs?

The programming portfolio is due on December 12. Late portfolios will not be accepted.

Use of CVS

Version control is an indispensable component of today's computer science landscape in industry, the academe, and the open source community. The Keck Lab provides each user with individual version control depots via CVS (Concurrent Versions System). We will make heavy use of CVS in this course: all programming assignments and your final programming portfolio are to be turned in electronically via CVS. Specific instructions and guidelines on CVS use will be provided. Until then, you are encouraged to read up on CVS on your own, particularly if you've never used it before.

Attendance

I am not a stickler for attendance, but I like having a full class. Note that the add/drop with 100% refund deadline is September 1. The withdrawal or credit/no-credit status deadline is November 3.

University Policy on Academic Honesty

Loyola Marymount University expects high standards of honesty and integrity from all members of its community. Applied to the arena of academic performance, these standards preclude all acts of cheating on assignments or examinations, plagiarism, forgery of signatures or falsification of data, unauthorized access to University computer accounts or files, and removal, mutilation, or deliberate concealment of materials belonging to the University Library.

Course Schedule

This schedule may change based on the actual ebb and flow of the class; deadlines, exams, and university dates (*italicized*) are less likely to change than lecture topics.

September	HCI guidelines, principles, and theories; Java's Swing
<i>September 1</i>	<i>Add/drop deadline for full refund</i>
October	Interaction styles; advanced Swing; OpenGL's GLUT
October 5	Midterm
<i>October 23–24</i>	<i>Undergraduate holidays; no class</i>
November	Usability evaluation; XHTML/CSS/JavaScript
<i>November 3</i>	<i>Withdrawal/credit/no-credit deadline</i>
<i>November 23–24</i>	<i>Thanksgiving; no class</i>
December 5	Design posters due
<i>December 12</i>	<i>Final exam, 11am; programming portfolios due</i>

You can view the class calendar on the Web at <http://ical.mac.com/dondi/LMU>. If you have an iCalendar-savvy client (i.e., Mozilla Calendar, Ximian Evolution, KOrganizer, Apple iCal, etc.), you can subscribe to the class calendar at webcal://ical.mac.com/dondi/LMU.ics. On-the-fly updates and adjustments to the class schedule will be reflected in this calendar.