CMSI 371

COMPUTER GRAPHICS

http://myweb.lmu.edu/dondi/spring2008/cmsi371

Spring 2008 — Doolan 222
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 master the principles of the art and science of computer graphics and become proficient in the design and programming of interactive graphics applications. The emphasis is on learning how to architect and write graphics software, rather than on learning how to use graphics software that has already been developed. Students will be exposed to basic computational geometry and OpenGL programming, while gaining valuable exposure to other technologies such as graphics in Java.

Course Requirements

Mastery of a programming language such as Java, C, or C++; expert knowledge of data structure and algorithm design; some familiarity with object-oriented programming, computer hardware, and operating systems.

Materials and Texts

- Edward Angel, *Interactive Computer Graphics: A Top-Down Approach with OpenGL*, Fourth Edition, Addison Wesley, 2006.
- Dave Shreiner, Mason Woo, Jackie Neider, and Tom Davis. *OpenGL Programming Guide*, Sixth Edition, Addison Wesley, 2007.
- Assorted handouts, articles, and sample code to be distributed throughout the semester.

The following texts are recommended and not required — but they *will* fill in a lot of details in case you're interested:

- Randi J. Rost. *OpenGL Shading Language*, Addison Wesley, 2004.
- Andrew S. Glassner. *Graphics Gems I.* Morgan Kaufmann, 1990.

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

Graded coursework consists of homework (25%), 1 midterm (25%), 1 graphics project (25%), and 1 final exam (25%). Letter grades are determined as follows: ≥ 90% gets an A– or better; ≥ 80% gets a B– or better; ≥ 70% gets a C– or better. The instructor 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 graphics project. 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 25% allocation of homework to your final grade.

Tests

The midterm is initially scheduled for February 21. The final exam is scheduled for May 6. 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.

Graphics Project

To demonstrate your proficiency in the covered material, you will design and implement some graphics software, chosen from these categories:

- *Portable, interactive graphics application.* This project uses an existing graphics engine to display or model something, well, "interesting." *Portable* means that your project will need, at most, a recompile to run on multiple platforms. *Interactive* means that the user can make dynamic changes to the application's model or display.
- *Graphics algorithm implementation.* This project performs something "under the hood" — either a vertex or fragment shader, or a 3D graphics engine, using your own implementations of the algorithms covered in class.
- *Graphical user interface (GUI) toolkit.* This project synthesizes material from CMSI 370 and 371 by building a GUI toolkit (buttons, menus, other controls) on top of a "raw" graphics library such as OpenGL.

The graphics project will be graded according to the following 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? Does it perform its tasks in a reasonable amount of time? 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. *Version control (5%):* Is the code committed at reasonable intervals? Are milestones appropriately tagged? Are adequate descriptions of provided in the commit logs?

Graphics project deliverables are due on May 6. Late projects will not be accepted.

Attendance

I am not a stickler for attendance, but I do like having a full class. Remember that the university add/drop with 100% refund deadline is January 18. The deadline for withdrawal or credit/no-credit status is March 14.

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.

January	How to use OpenGL
January 18	*Add/drop deadline for full refund*
February	Graphics and memory; transforms; object modeling
February 21	Midterm
March	Viewing and projection; clipping; hidden surface removal; shading
March 3–7	*Spring break; no class*
March 14	*Withdraw/credit/no-credit deadline*
April	Graphics primitives; miscellaneous topics (time permitting)
May 6	*Final exam, 11am; graphics projects due*

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.