

CMSI 371

COMPUTER GRAPHICS

Fall 2004

Doolan 219

Tuesdays and Thursdays, 3:00–4:15pm

3 semester hours

Office Hours: M 3–5pm, TR 2–3pm

John David N. Dionisio, PhD

dondi@lmu.edu

AIM: dondi2lmu

(310) 338-5782

Doolan 106

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 existing tools to create images, film, or business presentations. 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 high-level programming language such as Java or C++; expert knowledge of data structure and algorithm design; proficiency with vector and matrix operations; some familiarity with object-oriented programming, computer hardware, and operating systems; ability to document, demonstrate and explain one's own software; willingness to participate actively in class discussions.

Materials and Texts

- Edward Angel, *Interactive Computer Graphics: A Top-Down Approach with OpenGL*, Second Edition, Addison Wesley, 1999.
- Mason Woo, editor, *OpenGL® 1.2 Programming Guide, Third Edition: The Official Guide to Learning OpenGL, Version 1.2*, Addison Wesley, 1999.
- Assorted handouts and sample code to be distributed throughout the semester.

Course Work and Grading

Your graded coursework will consist of accumulated homework (10%), 3 quizzes (30% or 10% each quiz), 1 graphics project (30%), and 1 final exam (30%). Ungraded coursework includes frequent discussion of current topics and your own work in front of the class. Letter grades are determined using the percentage-to-letter map shown below:

93–100%	A	80–82%	B–
90–92%	A–	77–79%	C+
87–89%	B+	73–76%	C
83–86%	B	70–72%	C–

Fractions of a percent are handled with the usual rule: ≥ 0.5 rounds up to the next integral value. Your programming style will play a part in the grading of any code that you write. Poor structure, under- (or non-) commenting, bad naming, inappropriate hard-coding, and difficult maintenance constitute negative qualities that will drag down the grade of an otherwise working program.

Homework

Homework will consist of assorted 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 submit your homework on time, you will get full credit for them, regardless of their correctness. What goes around comes around — the effort you put into your homework pays off in the quizzes, final exam, and project.

Homework is due at the beginning of the next class. We will always “process” the homework on the class that it is due by having one or more of you talk through what you did; it is frequently just as important to be able to communicate your understanding of a subject as it is to understand it in the first place — particularly if you are working in a team and trying to understand each other’s code!

Late homework, whether 10 minutes late or 10 days late, will receive half credit. At the end of the semester, the proportion of homework that I have from you relative to the total homework assigned will answer for 10% of your final numeric grade. For example, if homework is assigned 5 times throughout the semester, 5 out of 5 on-time submissions result in a full 10%; 3 on-time submissions and 2 late submissions add up to 8%; and 2 submissions result in 4% credited.

Yes, technically, you can still get an A if you submit no homework at all but absolutely ace the research paper, quizzes, and final exam. But...do you really want to do that?

Graphics Project

You will design and implement an interactive graphics application or toolkit using OpenGL and/or Java. *Interactive* means that the user can make dynamic changes to the application’s model or display. For a toolkit, provide an application that demonstrates the use of this toolkit. The project consists of five deliverables:

1. “*White paper*” describing what you will do: submit as a text file using CVS (see below); due September 21.
2. *Development journal* — a diary that documents your on-going progress. Because you’ll be using CVS, building this diary will be virtually automatic.
3. *Source code and supporting files* — this should be everything a “customer” will need to compile, run, and use your project. To be submitted via CVS.
4. *Midsemester oral progress report* — a lab session where you show me what you’ve done so far. We will also look at the state of your development journal, source code, and supporting files at this juncture. Currently scheduled for October 28.
5. *Final oral presentation* — time to show off! This is a lab session at the end of the semester to share your labor of love with everyone in the class. Currently scheduled for December 7.

Sample ideas: an interactive 3D weather map; a 3D “working model” of a real-world device (clocks, vehicles, cell phones, calculators); a 3D construction set of some kind (Lego bricks, buildings, landscapes, faces); a physics simulation (sports, fireworks, heavenly bodies). And oh yes, a graphics-oriented game, with emphasis on graphics — not the game.

Use of CVS: CVS stands for Concurrent Versions System — it is a source code management system that many development shops use for archiving, storing, and updating their source code. We will use CVS to manage your project files, and we will use specific CVS features to build your development journals and

mark milestones in your code. Your Keck Lab accounts automatically come with CVS set up in your home directories.

Quizzes and Final Exam

Quiz 1 is initially scheduled for September 30. Quiz 2 is scheduled for October 21. Quiz 3 is scheduled for November 23. The final exam is scheduled for December 14. All tests are open-*paper*-everything; no sharing. You may neither solicit nor give help while the exam is in progress. Late and/or missed tests will be handled on a case-to-case basis; in all instances, talk to me about them.

Attendance

I am not a stickler for attendance, but I do like having a full class. Your submitted work will determine your final grade. Remember that the university add/drop deadline is November 5.

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

September 21	Project white paper due
September 30	Quiz 1
October 21	Quiz 2
October 26	Undergraduate holiday; no class
October 28	Midsemester oral progress report, including review of development journal and source code
November 5	University add/drop deadline
November 23	Quiz 3
November 25	Thanksgiving; no class
December 7	Development journal, source code, and final oral presentation
December 14	Final Exam (2:00 PM)

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.