

# CMSI 671

## COMPUTER GRAPHICS

<http://myweb.lmu.edu/dondi/spring2005/cmsi671>

**Graduate Level, Spring 2005** — Doolan 222

Tuesdays, 6:30–9:30pm, 3 semester hours

Office Hours: T 4–6pm, TR 11am–12pm

John David N. Dionisio, PhD

e-mail: [dondi@lmu.edu](mailto: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 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.

Alternatively, much of the content in the above materials is available in various forms on the Worldwide Web; starter links are available on the class Web site. Do not hesitate to search for and find additional sources of information regarding the techniques, tools, and paradigms that we will discuss.

### **Course Work and Grading**

Your graded coursework will consist of accumulated homework (10%), 1 midterm (30%), 1 graphics project and paper (30%), and 1 final exam (30%). Ungraded coursework includes discussion of current topics and your own work in front of the class.

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. Fractions of a percent are handled with the usual rule:  $\geq 0.5$  rounds up to the next integral value. The instructor may curve your grade upward based on qualitative considerations such as degree of difficulty, effort, 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

submit your homework 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 exams and the project.

Homework is due at the beginning of the next class. Late homework, whether 10 minutes late or 10 days late, will receive half credit. Occasionally, “extra credit” homework may be assigned. Fulfilling this extra credit work is counted *on top of* the 10% allocation of homework to your final grade.

## Research Paper and Presentation

You are asked to research, study, model, and render a real-world entity using OpenGL. Sample ideas: working models of real-world devices (clocks, vehicles, cell phones, calculators); a physics simulation (sports, fireworks, heavenly bodies); natural phenomena (fire, fluids, refractive surfaces, fabrics, fibers) — you will be surprised at how much thought and analysis may be required of the simplest real-world entities. You may use any modern high-level language for which a portable implementation of OpenGL exists — most likely C, C++, or Java. The work involved includes:

1. Reviewing current computer graphics and modeling literature,
2. Downloading, testing, or implementing software for the real-world entity that you have chosen,
3. Packaging the code into an interactive graphics application,
4. Writing a near-publication-quality paper that documents your work, and
5. Presenting the final results to the class at the end of the semester.

Items 4 and 5 comprise the final deliverables: the *near-publication-quality paper* as well as a *10–20 minute presentation* of your work. There are no hard limits on paper length, but 10–20 pages of 1.5-spaced, 12-point text with 1-inch margins, not including the list of references cited, is typical. Your research will be evaluated along the following criteria:

1. *Content* (40%): What is the quality of the work? Is the model elegant, thorough, and innovative? Is the implementation efficient? How realistic is the final rendering?
2. *Organization* (30%): Are the paper and presentation well-structured? Are concepts and the flow of ideas easy to follow? Are all required sections present and clearly identified?
3. *Writing* (20%): Are statements clear and easy to follow? Is the language precise, unambiguous, and grammatically correct?
4. *Polish* (10%): Is the content properly proofread? Are there many misspellings, typos, or other formatting faux pas?

To compel you to focus on the actual work and content of the project (as opposed to busy work such as formatting and reference management), the prospectus and paper must be written using LaTeX. We will talk about LaTeX in class.

Programming style plays a part in the grading of any code that you write, according to these criteria:

1. *Design*: How good is the overall structure of the program? Is it clear, easy to understand, flexible, and easy to maintain? Is it elegant or innovative? How closely does it follow the “one change, one place” rule of thumb?
2. *Functionality*: How well does the program work? Does it accomplish the project’s goals? Are its results accurate or correct? Does it perform its tasks in a reasonable amount of time?
3. *Naming*: Are program entities — classes, subroutines, variables, etc. — clearly and consistently named? Do their names correspond to their functions and roles?

4. *Comments*: Are comments provided where appropriate? Are they clear and well-written? Do they use special features provided by the project's platform (e.g. Javadoc in Java)?

Prior to launching full-bore into the project, you will first need to submit a prospectus that we will refine together until we agree on the scope and subject matter of the work. The prospectus is due at the beginning of our February 1 class, and will be finalized by February 15 at the latest.

The paper and presentation are due at the beginning of our last class, April 26.

## Exams

The midterm is initially scheduled for February 22. The final exam is scheduled for May 3. All tests are open-*paper-everything*; no sharing. Electronic lookups may also be allowed depending on the scope or subject matter. You may neither solicit nor give help while an 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 March 18.

## 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 Topics and 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	Overview; introduction to graphics APIs; basic graphics techniques
February	The math behind computer graphics; object and environment modeling
<i>February 1</i>	<i>Project prospectus due</i>
<i>February 15</i>	<i>Project prospectus finalized (at the latest)</i>
<i>February 22</i>	<i>Midterm</i>
March	Computer graphics algorithms: 3D viewing, rendering
<i>March 18</i>	<i>University add/drop deadline</i>
<i>March 22</i>	<i>Easter break; no class</i>
April	Advanced graphics topics: specifics TBA
<i>April 26</i>	<i>Research paper due; final project presentations</i>
<i>May 3</i>	<i>Final Exam (6:30 PM)</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.