CMSI 3710

COMPUTER GRAPHICS

http://dondi.lmu.build/spring2024/cmsi3710

Spring 2024—Seaver 309, MW 1:45–3:25pm; 4 semester hours
Office Hours MTW 4–6pm,
or by appointment *(don't hesitate to ask!)*

John David N. Dionisio, Ph.D.
dondi@lmu.edu
Doolan 102; (310) 338-5782

Objectives and Outcomes

This course explores the computer science subfield of *computer graphics*—the study and development of algorithms for synthesizing, manipulating, and displaying visual information. Long after the course concludes, my hope is that you will be able to:

1. **Represent, model, and create visual information digitally.**
2. **Manipulate and display visual information both computationally and mathematically.**
3. **Use and develop computer graphics APIs at different levels of abstraction.**

In addition to the course-specific content, you are also expected to:

4. **Follow disciplinary best practices throughout the course.**

Prerequisites/Prior Background

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

Materials and Texts

No single book is required, but many are recommended, for you to use as needed. They fall into these categories:

Theoretical/Mathematical Foundation

- Peter Shirley and Steve Marschner et al, *Fundamentals of Computer Graphics*, 5th Edition, A K Peters/CRC Press, 2021

Practical/Programming

- Farhad Ghayour and Diego Cantor, *Real-Time 3D Graphics with WebGL 2*, 2nd Edition, Packt Publishing, 2018
- Jacobo Rodriguez, *GLSL Essentials*, Packt Publishing, 2013

Reference Works

- John Kessenich, Graham Sellers, and Dave Shreiner, *OpenGL Programming Guide*, 9th Edition, Pearson Education, 2016—*a.k.a. "the Red Book"*
- Randi J. Rost, Bill Licea-Kane, et al., *OpenGL Shading Language*, 3rd Edition, Pearson Addison Wesley, 2021—*a.k.a. "the Orange Book"*

Assorted handouts, articles, and sample code will also be distributed throughout the semester. In addition, do not hesitate to look for further information regarding the concepts, techniques, tools, and paradigms that we will discuss.

Course Work and Grading

Your final grade will be based on the percentage of the points you get for the following deliverables against the total number of possible points:

Fun with *their* 3D library	100
Pixel-level primitives and filters	100
Static scene based on *our* polygon mesh library	100
Our own matrix library	100
Our lights, *our* camera…	100
…*Your* action!	200
Total	700 points

Percentages ≥ 90% get an A– or better; ≥ 80% get a B– or better; ≥ 70% get a C– or better. I may nudge grades upward based on qualitative considerations such as degree of difficulty, effort, class participation, time constraints, and overall attitude toward the course.

Term Portfolio

Your accumulated assignments for the semester comprise the *term portfolio*—the final, definitive artifact that demonstrates the course's outcomes. It is how you show whether you have, indeed, accomplished the objectives of this course. Your portfolio for this course consists of the following:

- An interactive 3D scene based on a high(er)-level graphics library (*1b*, *1c*, *2a*, *2b*, *3a*, *3c*)
- Simple image processing filters (*1a*, *2c*, *3b*)
- A 3D object library with polygon meshes, object composition, and instance transformations (*1a–1c*, *2a*, *2b*, *3a*, *3d*)
- Vertex and fragment shaders implementing commonly-used computer graphics algorithms and techniques (*2a–2d*, *3a*, *3b*, *3d*)
- An interactive 3D scene based on these libraries (*1a–1c*, *2a–2d*, *3a*, *3b*, *3d*)

The full range of *4a–4f* applies to all assignments.

An assignment's number is its due date in *mmdd* format, and it is always due by 11:59:59.999pm of that date. Point values are based on the state of your assignments at that moment.

Version Control

Version control is an indispensable part of today's computer science landscape in industry, the academe, and the open source community. We use version control heavily in this course: make sure that you get the hang of it.

None of the assignments can be completed (well) overnight; they should be the result of steady progress from the moment they are assigned to the date they are due. "One and done" submissions will negatively affect the final score.

Workload Expectations

In line with the *LMU Credit Hour Policy*, the workload expectation for this four-credit-hour course is a minimum of $4 \times 3 = 12$ hours of work by an average student per week, including the time that we spend together in the classroom.

Attendance

Attendance at all sessions is expected, but not absolutely required. If you must miss class, it is your responsibility to notify me about this and keep up with the course.

The last day to add or drop a class without a grade of W is January 12. The withdrawal or credit/no-credit deadline is April 5.

Academic Honesty

Loyola Marymount University is a community dedicated to academic excellence, student-centered education, and the Jesuit and Marymount traditions. As such, the University expects all members of its community to act with honesty and integrity at all times, especially in their academic work. Academic honesty requires that all members of the LMU community act with integrity, respect their own intellectual and creative work as well as that of others, acknowledge sources consistently and completely, act honestly during exams and on assignments, and report results accurately. As an LMU Lion, by the Lion's Code, you are pledged to join the discourse of the academy with honesty of voice and integrity of scholarship.

Academic dishonesty will be treated as an extremely serious matter, with serious consequences that can range from receiving no credit for assignments/tests to expulsion. It is never permissible to turn in any work that has been copied from another student or copied from any source (including the Internet) without properly acknowledging/citing the source. It is never permissible to work on an assignment, exam, quiz, or any project with another person unless your instructor has indicated so in the written instructions/guidelines. It is your responsibility to make sure that your work meets the standard of academic honesty set forth in the "Academic Honesty Policy" found at:

https://academics.lmu.edu/honesty

Technology Use and Academic Honesty

In this course, academic honesty includes the *appropriate* use of technology as an aid for learning and productivity. This includes but is not limited to generative artificial intelligence tools such as ChatGPT and GitHub Copilot.

Modern Generative AI provides fast solutions to a variety of computing problems, but should be used responsibly in the classroom setting to get the most out of your education. Generative AI will not always be helpful nor available, like in job interviews, technologies it hasn't seen, or extraordinary/unique scenarios. It cannot independently validate the code that it produces, which may contain bugs or security vulnerabilities. As such, you will still need to deeply understand the foundations of computing and should use it in your coursework sparingly, especially when its use might rob you of the same learning and desired difficulty of problem solving that comes from programming on your own.

That said, here are some acceptable use cases of generative AI on programmatic assignments:

- Generating docstrings and type hints
- Getting examples/explanations for language mechanics/syntax (e.g., how to use the spread operator; how to assign variables directly from an array or object; how to implement an error boundary component)
- Interpreting error messages
- Generating unit tests for edge cases
- Brainstorming new ideas for apps or app names

TL;DR *Never use technology (including but not limited to AI) to avoid learning, but do use it to augment, amplify, or accelerate (your) learning.*

Special Accommodations

The Disability Support Services (DSS) Office offers resources to enable students with ADD/ADHD; physical, learning, and psychiatric disabilities; and those on the autism spectrum to achieve maximum independence while pursuing their educational goals. Staff specialists interact with all areas of the University to eliminate physical and attitudinal barriers. Students must provide documentation for their disability from an appropriate licensed professional. Services are offered to students who have established disabilities under state and federal laws. DSS personnel also advise students, faculty, and staff regarding disability issues. Students who need reasonable modifications, special assistance, academic accommodations or housing accommodations should direct their request to the DSS Office as soon as possible. All discussions will remain confidential. The DSS Office is located on the 2nd floor of Daum Hall and may be reached by email at *dsslmu@lmu.edu* or phone at (310) 338-4216. Please visit *http://www.lmu.edu/dss* for additional information.

Course Evaluations

Student feedback provides valuable information for continued improvement. All students are expected to fairly and thoughtfully complete a course evaluation for this course. This semester, course evaluations will be administered online through the Blue™ evaluation system. You will receive an email notification at your Lion email address when the evaluation form is available. You may also access the evaluation form on Brightspace (*https://brightspace.lmu.edu*) dashboard during the evaluation period. Your responses will be anonymous and will not be linked to you in any way.

Topics and Important Dates

Correlated outcomes are shown for each topic. Specifics may change as the course progresses. University dates (italicized) are less likely to change.

January	3D graphics with a JavaScript high-level library *(1b, 1c, 2a, 2b, 3a, 3c)*
January 12	*Last day to add or drop a class without a grade of W*
February	Graphics and memory *(1a, 2c, 3b)*; 2D graphics primitives *(1a, 3b)*; graphics with JavaScript and foundational libraries *(1b, 1c, 2a, 2b, 2c)*; introduction to programmable shaders *(3d)*
February 26–March 1	*Spring break; no class*
March	Object modeling *(1b, 1c)*; transforms *(2a, 2b, 3d)*; viewing and projection *(2b, 3a, 3d)*
March 27–29	*Easter break; no class*
April	Lighting and shading *(2c, 3d)*; clipping and hidden surface removal *(2d)*; 3D scene workshop/lab sessions *(1a–3b, 3d)*
April 1	*Cesar Chavez Day; no class*
April 5	*Last day to withdraw from classes or apply for Credit/No Credit grading*
April 29	*Last set of term portfolio assignments due*

You can view my class calendar and office hour schedule in any iCalendar-savvy client. Its subscription link can be found on the course web site (it's too long to provide in writing).

Tentative Nature of the Syllabus

If necessary, this syllabus and its contents are subject to revision. Students are responsible for any changes or modifications announced or distributed in class, emailed to students' LMU Lion accounts, or posted on LMU's course management system, Brightspace. If you are absent from a synchronous class session, it is the student's responsibility to check Brightspace and with the professor to see if you missed any important class announcements. Students should not rely on word-of-mouth from classmates.

Course Outcomes

1 Represent, model, and create visual information digitally.

1a	*…in terms of pixels and geometric primitives.*	With a few exceptions, these outcomes will be demonstrated within a single, cumulative "scene" program to be developed throughout the semester. Assignments incrementally build on previous ones. It will thus be more important than usual that you keep up and keep current.
1b	*…in terms of polygon meshes: vertices, edges, and faces.*	
1c	*…as a composition of multiple discrete objects (scenes).*	

2 Manipulate and display visual information both computationally and mathematically.

2a	*Apply and implement transforms.*	In the same way that the study of general-purpose data structures starts with the structures themselves, then goes into algorithms and operations on those structures, so goes the study of computer graphics. Learning objective 1 looks at structure; learning objective 2 looks at algorithms and computations.
2b	*Project 3D objects onto a 2D viewport.*	
2c	*Perform color and light computations.*	
2d	*Be familiar with established algorithms such as clipping and hidden surface removal (HSR).*	

3 Use and develop computer graphics APIs at different levels of abstraction.

3a	*Develop a library of 3D objects.*	There is some overlap between these outcomes and the ones for learning objective 2. This is by design—the outcomes in objective 2 focus on understanding these computations conceptually; the outcomes in learning objective 3 look at your ability to implement them concretely. "Different levels of abstraction" pertain to *where* your implementation starts—high-level libraries allow you to write less code, but using someone else's assumptions and designs; low-level libraries expose full flexibility and power, but you need to write more code to fully tap that power.
3b	*Perform bit-level color manipulation.*	
3c	*Render a 3D scene using a high-level library.*	
3d	*Render a 3D scene using programmable shaders.*	

4 Follow disciplinary best practices throughout the course.

4a	*Write syntactically correct, functional code.*	Code has to compile. Code has to work. No errors, no bugs. Use unit tests as much as possible.
4b	*Use programming best practices, demonstrating principles such as DRY, proper separation of concerns, correct scoping of variables and functions, etc.*	This is the basis of good software design. It makes software easier to maintain, improve, and extend. Heed feedback well. *What you learn here will apply to future work in this field, in school and beyond.*
4c	*Write code that is easily understood by programmers other than yourself.*	This outcome involves all aspects of code readability and clarity for human beings, including but not limited to spacing & indentation, proper naming, presenting code in a manner that is consistent with its structure, documentation & comments when appropriate, and adherence to conventions or standards.
4d	*Use available resources and documentation to find required information.*	The need to look things up never goes away. Remember also that the course instructor counts as an "available resource," so this outcome includes asking questions and using office hours.
4e	*Use version control effectively.*	In addition to simply using version control correctly, effective use also involves appropriate time management, commit frequency, and descriptive commit messages.
4f	*Meet all designated deadlines.*	