

CS 150 — Topics: 3-D Computer Graphics

Time and place: 10:00–11:50, TTh, Porter Hall 3.

Professor: David J. Hunter, Ph.D.

e-mail: dhunter@westmont.edu
 Office Phone: x6075
 Office: New Math/CS Offices
 Office Hours: Friday morning, or by appointment.

Prerequisites: Consent of instructor. The desirable prerequisites for this course are MA 20 (Linear Algebra) and at least one semester of programming in C or C++. Students without this background may have some catching-up to do during the semester.

Course objectives: This course will give a general introduction to three-dimensional computer graphics. The point of view will be “low-level,” that is, we will focus on the general *mathematical fundamentals* of computer graphics. Although we will use a modern API (OpenGL) to see how things work, the main purpose of the course is to give you enough familiarity with the mathematical underpinnings of three-dimensional graphics to be able to use any current (or future) API.

It might be helpful to be up front about what this course is *not*:

- ⊗ This is not a course in using the computer to do graphic design with programs like Photoshop or Gimp. The Art Department teaches these courses.
- ⊗ This is not, strictly speaking, a programming course. Although many of our exercises will involve writing programs, we will not wade too deeply into the minutia of programming (which varies from language to language).
- ⊗ This course is not about teaching you how to use a Graphical User Interface. We will be able to do everything with a console window and a graphics window that can capture mouse clicks and keystrokes. We will not discuss radio buttons, drop-down menus, spinners, dialog boxes, and all that.

This course *is* about the fundamentals of three-dimensional computer graphics, including: the rendering pipeline, transformations, viewing, lighting, textures, splines, animation, and kinematics. In addition to coverage of these topics, I have in mind the following goals:

- You will appreciate the deep interplay between mathematics and graphics programming, and you will become a better programmer by learning more mathematics.
- You will grow as a scientist who understands the foundations of software engineering.
- You will be able to write a reasonably complex program using OpenGL.
- You will consider how the Christian faith relates to the discipline of computer science. In particular, you will develop a discerning view of the way our culture uses visual images and virtual reality.

Textbook: *3-D Computer Graphics: A Mathematical Introduction with OpenGL*, by Samuel Buss, Cambridge University Press, 2003.

Assignments: Regular assignments will be announced in class and on **eureka**. Some of these assignments will involve writing programs, others will be mathematical problem sets, and some will be a combination of the two. I will use a point system to weight assignments according to difficulty.

This is an upper-division computer science course, so I expect (1) that you already know how to program in C++, and (2) that you can figure out how to get OpenGL to work on your favorite computer without a lot of help from me. I will grade the programs on a Linux machine using the Gnu compiler, so make sure anything you turn in works in that environment. Specifically, you must turn in a **.tar.gz** file containing all **.c** or **.cpp** and **.h** files, as well as a **Makefile**. Given these constraints, I strongly encourage you to write your programs using a *nix machine and compile them with **gcc** or **g++**. Late assignments will lose 10% credit for each day late (or fraction thereof).

Grading: Your grade will be calculated as follows:

Assignments: 50%
Midterm #1 15%
Midterm #2: 15%
Final: 20%

I grade on the standard 90/80/70/60 scale, although I reserve the right to decide borderline cases by taking into account attendance and participation.

Other Policies: I expect you to attend class every day. Please inform me in advance of any classes you plan to miss. Work missed (including tests) without a valid excuse will receive a zero.

You will learn more and enjoy class more if you make an effort to participate fully. Good participation involves being attentive in class, asking questions when confused, and being a good sport about exercises and labs that we work on during class. You will find it easier to participate if you read (or at least skim) the assigned pages before we discuss them in class.

I will post additional information and course materials on the `eureka` page for this course. Go to <http://eureka.westmont.edu>

and make sure you are subscribed to this course. I expect you to check this page, along with your email, on a regular basis. If you use a non-Westmont email account, please forward your Westmont email to your preferred account. Instructions for doing so are on the IT Wiki:

<https://eureka.westmont.edu/mod/wiki/view.php?id=2848>

Learning communities function best when students have academic integrity. Cheating is primarily an offense against your classmates because it undermines this community. Therefore, **dishonesty or plagiarism will result in a zero for the work in question, and repeated or major infractions will result in expulsion from the course with a grade of F.**

If you miss more than four classes without a valid excuse, I reserve the right to terminate you from the course with a grade of F.

Tentative Schedule: We will try to abide by the schedule below, although it is subject to revision at the discretion of the instructor. For more detailed information, consult `eureka`.

Reading	Topics
I.1–I.3	Introduction: pixels, points, lines, OpenGL
II.1–II.4	Transformations and viewing
III.1	Lighting, illumination, and shading: The Phong model
	<i>Midterm #1</i>
IV.1–IV.2	Averaging and interpolation
V.1–V.4	Texture mapping
VI.1–VI.2	Color
VII.1–4,10,11	Bézier curves and surfaces
	<i>Midterm #2</i>
VIII.1–VIII.4	B-Splines
XII.1–XII.4	Animation and Kinematics
	<i>Final Exam: Monday, April 28, 12-2pm</i>