

**CMSC 23700
Winter 2012**

Introduction to Computer Graphics

**Handout 1
January 3**

Course information

(v2.1 Feb 14: removed “individual” qualification from 24-hour extension on projects)

- Instructor:** Gordon Kindlmann (glk@uchicago.edu)
- TA:** Ross Girshick (rbg@cs.uchicago.edu)
- Lectures:** Tuesday and Thursday 1:30pm – 2:50pm
Ryerson 251
- Labs:** **Thursday** (not Wednesday) 4:30pm-5:50pm
MacLab (the Macs) in Regenstein Library
<http://csmacslab-www.uchicago.edu/>
- Office hours:** Kindlmann (Ryerson 161-B): Tuesday 3pm-4pm, Friday TBA;
Girshick (Ryerson 275-A): Tuesday 4pm-5pm
- Mailing list:** cmssc23700@mailman.cs.uchicago.edu
<https://mailman.cs.uchicago.edu/mailman/listinfo/cmssc23700>
- Web page:** <http://people.cs.uchicago.edu/~glk/23700>
- Phoenixforge:** web: <https://phoenixforge.cs.uchicago.edu/projects/cmssc23700-win-2012>
svn: <https://phoenixforge.cs.uchicago.edu/svn/cmssc23700-win-2012>

Overview

This course aims to provide an introduction to the basic concepts and techniques of 3D computer graphics, with a focus on interactive rendering techniques, such as those found in computer games. These include: coordinate systems and transformations, surface appearance and shading, geometric modeling, level-of-detail, and efficiently using graphics hardware.

The course covers mathematical and programming aspects. The lectures, homework assignments and exam will focus on the mathematical and algorithmic foundations of computer graphics, while the lab sessions and programming projects focus on implementing these with modern (shader-based) OpenGL. The lab section is a **required** part of the course. You are responsible for the material presented in lab, and lab time will sometimes be used for project demos.

Note: Information in this handout is subject to change; the web page will have the latest information.

Prerequisites

Trigonometry, linear algebra, and matrix operations (on 3x3 and 4x4 matrices) will be reviewed briskly; some familiarity with these will be important. You should be able to create and debug non-trivial C programs, as this skill will not be taught in lectures or labs.

Texts

There are two required texts:

Real-time Rendering (3rd Edition)

by Tomas Akenine-Möller, Eric Haines, and Naty Hoffman
A.K. Peters Ltd., 2008.

OpenGL SuperBible: Comprehensive Tutorial and Reference (5th Edition)

Richard S. Wright, Nicholas Haemel, Graham Sellers, Benjamin Lipchak
Addison Wesley, 2011

The programming assignments will be written using the C programming language. If you do not have a good C manual, we recommend the following:

C – A Reference Manual (5th Edition)

by Samuel P. Harbison and Guy L. Steele Jr.
Prentice Hall, 2002

We have requested that a copy of the Real-time Rendering text be placed on reserve in the Eckhart Library.

Grading

There will be homework assignments, programming projects, and a midterm exam (middle of February). The final project will be a group project (up to two-people per group). Grading will be based on following approximate weights:

Homework	20%
Midterm exam	30%
Projects	50%

The extent of your thoughtful class participation (asking and answering questions) will also be a factor.

Homeworks and programming projects will be posted on the class website and announced via email to the class list. Both will be handed in by you via pheonixforge.

Late programming projects will be docked 12% every 6 hours, in increments of 6 hours, based on the clock on the pheonixforge server. After 72 hours the project will receive a zero. There is no similar leniency with homework assignments. However, each student is allotted one 24-hour extension on one homework assignment **or** one programming project. You must request the extension at least 24 hours before the deadline.

Programming project grades will consist of a **correctness** portion (worth 70%) and a **style** portion (worth 30%). Your code **must** compile. Failure to compile will result in a 0 for the correctness portion of the grade.

Syllabus

The lectures are expected to cover the following topics:

- Overview of the graphics pipeline [Ch. 2]
- Graphics processors (GPUs) [Ch. 3]
- Linear algebra, trigonometry, model/view transformations and projections[Ch. 4; Appendices]
- Basic lighting and shading[Ch. 5]
- Texturing and bump mapping[Ch. 6]
- Vertex and Fragment shaders
- Shadows [Ch. 9.1]
- Object representations: meshes, scene graphs, level-of-detail refinements [Ch. 12]
- Spatial data structures [Ch. 14]
- Intersection testing [Ch 16]

Though most topics covered in class are also described in the main text, class lectures and additional readings will also present required information. Slides from class will be made available on [Chalk](#), but these will *not* be a substitute for taking notes in class (*i.e.* you need to come to class).

Academic Honesty

The University of Chicago is a scholarly academic community. You need to both understand and internalize the ethics of our community. A good place to start is with the Cadet's Honor Code of the US Military Academy: "A Cadet will not lie, cheat, or steal, or tolerate those who do." It is important to understand that the notion of property that matters most to academics is ideas, and that to pass someone else's ideas off as your own is to lie, cheat, and steal.

The University has a two-paragraph formal policy on [Academic Honesty and Plagiarism](#), which you should read and understand.

We believe that student interactions are an important and useful means to mastery of the material. We recommend that you discuss the material in this class with other students, and that includes the homework assignments. So what is the boundary between acceptable collaboration and academic misconduct? First, while it is acceptable to *discuss* homework, it is not acceptable to turn in someone else's work as your own. When the time comes to write down your answer, you should write it down yourself from your own memory. Moreover, you should cite any material discussions, or written sources, for example,

Note: I discussed this exercise with Jane Smith.

The University's policy says less than it should regarding the culpability of those who know of misconduct by others, but do not report it. An all too common case has been where one student has decided to "help" another student by giving them a copy of their assignment, only to have that other student copy it and turn it in. In such cases, we view both students as culpable and pursue disciplinary sanctions against both.

For the student collaborations, it can be a slippery slope that leads from sanctioned collaboration to outright misconduct. But for all the slipperiness, there is a clear line: present only your ideas as yours and attribute all others. If you have *any* questions about what is or is not proper academic conduct, please ask your instructors.

Thanks to Stuart Kurtz and John Reppy for the above exposition of this essential information.