22C:151 Introduction to Computer Graphics
Syllabus for Fall 2007

Class Time: TTH 9:30–10:45 am, 221 MacLean Hall.
Class Webpage: http://www.cs.uiowa.edu/~cwyman/classes/fall07-22C151/

Instructor: Chris Wyman
E-mail: cwyman@cs.uiowa.edu *(Preferred contact method)*
Office: 101J MacLean Hall
Office Phone: (319) 353-2549 *(Please e-mail if possible)*
Office Hours: T 10:45 am–12:00 pm, Th 10:45 am–12:15 pm, or by appointment.

Teaching Assistant: Greg Nichols
E-mail: gbnichol@cs.uiowa.edu
Office: 317 MacLean Hall
Office Hours: MW 10:45 am–12:15 pm, or by appointment.

Prerequisite: Grade of C- or better in 22C:031 and 22M:027. Note that linear algebra (particularly matrices, transformations, and point and vector manipulation) is very important in computer graphics. We **will** make heavy use of these concepts, so you may wish to review them now if you feel you need a refresher.

Course Objective:

This course provides an introduction to the area of computer graphics. Over the past 30 years, computer graphics has revolutionized movie and printing techniques, improved human-computer interfaces, and driven new applications such as computerized photography, art, games, simulations, and mechanical design. While graphics has become widespread, few people understand the internal workings of applications like video games, Adobe Photoshop, and the renderers behind Hollywood's special effects and recent CG blockbusters.

The goals of this course will be to: introduce OpenGL, a common graphics programming API; discuss a number of important, low level implementational details hidden by APIs like OpenGL; and survey a number of subfields of graphics.

Warning: This course is very programming intensive! In my opinion, the best way to learn applied areas such as computer graphics is by coding . . . lots. There will be assignments roughly once a week, most (or all) of which will include programming. For students proficient in programming in C or C++, assignments should take 3 to 6 hours each. For those less experienced in coding and debugging, these assignments may take much longer. If you consistantly spend more than 6 hours, you should consider that as a strong hint to ask for help.

Course Requirements:

Course grades will be based on weekly assignments, one midterm exam, and a final exam. Your final grade will be computed as follows:

70 % – *Homework*
15 % – *Midterm*
15 % – *Final Exam*

Note that images showing results of your programming assignments must be posted on a publicly-accessible webpage for full credit! *No credit* will be given for programming assignments without a corresponding webpage. You may, however, choose not to have this webpage linked from the course homepage.

Grades will be based on a plus/minus system determined via a curve, unless overall scores are very high, in which case a typical fixed scale (90+ = A+/A/A-, 80+ = B+/B/B-, 70+ = C+/C/C-, 60+ = D+/D/D-) will be used. Last year only a mild curve was necessary (A's $\in [89-100]$, B's $\in [80-89]$, C's $\in [64-80]$, and D's $\in [54-64]$), but previous years have seen more significant curves. Class participation, effort, and intellectual curiosity *will be* considered for students on the borderline between grades. Questions (including regrade requests) about scoring of assignments and exams must be asked within one week of the return of the assignment/exam.

Textbooks:

- **Required:** *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 2.1*, Sixth Edition. By Dave Shreiner, Mason Woo, Jackie Neider, and Tom Davis. Addison-Wesley, 2007. ISBN #0321481003

The fifth edition will be fine for this course, though there are numerous serious typos on material we might use in the last few weeks. The fourth edition is sufficient until the last few weeks of the semester. (Page numbers for readings can be found on last year's syllabus for the 5^{th} edition and the 2005 syllabus for the 4^{th} edition). I encourage anyone who plans to continue using OpenGL to purchase the current edition, however. Other excellent books you may wish to refer to for additional help include:

- **Further Reading:** *Interactive Computer Graphics: A Top-Down Approach Using OpenGL*, Fourth Edition. By Edward Angel. Addison-Wesley, 2003. ISBN #0321321375.
- **Further Reading:** *Fundamentals of Computer Graphics*, Second Edition. By Peter Shirley (and others). AK Peters, 2005. ISBN #1568812698.

Students may visit my office to peruse my copies of the books. I will place copies of these books on reserve in the Mathematics Library. Note that not all material covered in class (and on exams) is present in these textbooks.

Programming Language, Graphics API, Toolkits, etc.

Example code and lecture material will be presented in C or C++ using the OpenGL API. It is *highly* recommended that assignments be completed in C/C++. You are free to use whatever language you want, with four caveats:

- Setup and debugging help **will not** be offered for languages other than C or C++.
- Code other than C or C++ must run on all of the managed Linux computers in MLH 301.
- Instructions for compilation and running of executables must be extra clear.
- Partial credit may not be assigned for "partially working" code.

OpenGL should be installed on all departmental Linux and Windows machines. If your home computer does not have OpenGL installed, you may instead use Mesa, which is a free OpenGL implementation. Assignments may be done on either Linux (using gcc/g++) or Windows (using Visual Studio 2005 .NET). If you need copies of Windows XP, Windows Vista, or Visual Studio 2005 .NET, you may download them using our department's MSDNAA subscription. Find out how, by reading the instructions here: *http://msdnaa.cs.uiowa.edu/*.

OpenGL does not have calls to deal with user interaction, like mouse clicks or opening windows. To avoid teaching X or Windows interfaces, we will use GLUT (the GL Utility Toolkit), which is a portable API for basic windowing primitives. Documentation and copies of GLUT are available online, including at: *http://www.xmission.com/~nate/glut.html*. Students interested in more complex user interfaces may wish to use GLUI (*http://www.cs.unc.edu/~rademach/glui/*) or FLTK (*http://www.fltk.org/*), which are also portable across multiple windowing systems. Later in the course, we may use the GLEW library (*http://glew.sourceforge.net/*) to allow easy access to advanced OpenGL extensions.

Make sure your code is portable (i.e., I should be able to compile and run under Windows or Linux without modifications to the source code)! This is not difficult, as long as you write clean code. Assignments that do not execute as submitted, do not come with clear instructions detailing how to compile and run the program, or do not include all necessary files for execution or compilation will be given a zero score.

Late Assignments:

Unless otherwise specified programming assignments are due Wednesdays at 11:59 pm via ICON, written assignments are due at the beginning of class on Thursday, and images from programming assignments must be posted on your webpage by Thursday at 11:59 pm. There is a small grace period, after which assignments are penalized at the rate of 25% per day (i.e., 25% penalty by Thursday at 11:59 pm, 50% penalty by Friday, etc.)

Each student has six (6) free "late days" during the semester, for which no penalties will apply to late assignments. These days are meant for medical emergencies, absences due to university sponsored events, etc., so *no* additional "late days" or extensions will be granted to individuals unless you discuss the circumstances with me during the first two weeks of the semester.

Due to the quick turnaround time between assignments, no more than three (3) late days can be used per assignment. Assignments four or more days late will *always* receive no credit, unless you have prior approval.

Academic Honesty:

Academic dishonesty of any kind will not be tolerated. Unless otherwise stated in class, all assignments and exams are to be completed individually. While discussion of ideas and problems with fellow students is encouraged, code and written homeworks must be done individually. In certain circumstances, code fragments may be provided to eliminate tedious coding or to provide a common framework for all students. All other code *must* be original.

Online resources may be used to help you understand the material, but copying online code is grounds for failure. Period. As a continuum exists ranging through reading a textbook-like online resource, examining online code to help find bugs in your own, posting questions in online forums, writing your code while examining online code, and explicit copying, I encourage you to discuss such behaviors with me *before* utilizing them. Discussion after the fact is too late, and may lead to failure. If you find yourself relying on such resources to a greater and greater extent, you need to make an appointment with me so I can help clarify the course material.

For clarification on what constitutes academic dishonesty, contact me or consult the printed policy in the *Schedule of Courses*, the *CLAS Bulliten*, or online at *http://www.clas.uiowa.edu/faculty/teaching/classroom_p&p/acad_fraud_etc.shtml*. Clarification must occur *before* you turn in questionable work.

Further Considerations:

Makeup exams will not be given, except in circumstances allowed under the University of Iowa's policy on absences from examinations (*see: http://www.clas.uiowa.edu/faculty/teaching/classroom_p&p/general_exam_p&p.shtml*). If a makeup exam is necessary, please inform me as far in advance as possible.

I need to hear from any student with a disability that requires modification to seating, testing, or other class requirements. Please talk with me as soon as possible during office hours, so that appropriate arrangements can be made in a timely fashion. For more information on the procedures refer to: *http://www.clas.uiowa.edu/faculty/teaching/classroom_p&p/disabilities.shtml*

Note: As a course offered by the College of Liberal Arts and Sciences, course policies are governed by the CLAS.

Complaints:

If you have complaints, please feel free to discuss them directly with me during office hours or via e-mail. If you have problems with the TA, please attempt to resolve them with her first before contacting me. If you do not feel I have appropriately dealt with your complaint, you should consult the Computer Science DEO/Chair, Professor Jim Cremer, 14D MacLean Hall, (319) 335-1713, cremer@cs.uiowa.edu. If still unresolved, complaints must be submitted in writing to (for undergrads) Helena Dettmer, the CLAS Associate Dean for Academic Programs, or (for grads) to Eric Wurster, Graduate College Associate Dean for Academic Affairs. Further information about this policy is available at: *http://www.clas.uiowa.edu/students/academic_handbook/ix.shtml#4*.

Additional References:

Additional books that may prove helpful or interesting include:

- *Computer Graphics: Principles and Practice*, Second Edition in C. By James Foley, Andries van Dam, Steven Feiner, and John Huges. Addison-Wesley, 1997. ISBN #0201848406
- *Ray Tracing from the Ground Up*. By Kevin Suffern. A K Peters, 2007. ISBN #1568812728.
- *Realistic Ray Tracing*, Second Edition. By Peter Shirley and R. Keith Morley. A K Peters, 2003. ISBN #1568811985.
- *Physically Based Rendering*. By Matt Pharr and Greg Humphreys. Morgan Kaufmann, 2004. ISBN #012553180X.

These books are available for you to peruse in my office, and some of them are available from the library.

Tentative schedule:

	Topic	Assignment Due	Suggested Readings
August 28	Intoduction & Expectations, OpenGL State Machine, PPM files		
August 30	Introduction to OpenGL and GLUT		Red Book: Ch 1, Appendix D Interactive CG: Ch 1
September 4	Line Drawing Algorithms		Interactive CG: 7.8, 7.9 Fundamentals: 3.5
September 6	Line Drawing Algorithms	Asgn # 1 Due	Red Book: p. 27–55
September 11	Triangle Rasterization		Interactive CG: 7.10, 7.11 Fundamentals: 3.6
September 13	Triangle Rasterization, Linear Algebra Review	Asgn # 2 Due	Red Book: p. 55–65 Interactive CG: 4.1, 4.2
September 18	Linear Algebra Review, Homogeneous Coordinates, Transformations		Interactive CG: Appendices B,C Interactive CG: 4.3, 4.4, 4.6–4.8 Fundamentals: 2.3, 2.4, 5, 6
September 20	Transformations, Perspective, Perspective Matrix,	Asgn # 3 Due	Red Book: Appendix F, Ch 3 Interactive CG: 5.1–5.5
September 25	Perspective in OpenGL Matrix Stack		Fundamentals: Ch 7
September 27	Z-Buffer 3D Depth in OpenGL	Asgn # 4 Due	Interactive CG: 5.6, 7.11 Red Book: p. 185–187, 468–471
October 2	Barycentric Coordinates Basic Color Perception Phong Illumination Model		Red Book: Ch 4 Red Book: p. 220–225 Fundamentals: 2.11, 9.2, 20
October 4	Gouraud & Phong Shading OpenGL Lighting	Asgn # 5 Due	Red Book: Ch 5 Interactive CG: Ch 6
October 9	Texture Mapping		Red Book: Ch 9 Fundamentals: Ch 11
October 11	Texture Mapping	Asgn # 6 Due	
October 16	Questions & Review for Midterm		
October 18	**Midterm Exam**		
October 23	Shadows, Shadow Maps		Redbook: p. 459–461, 612–622 Interactive CG: 5.10
October 25	OpenGL Shadow Mapping	Asgn # 7 Due	
October 30	Introduction to Raytracing		Interactive CG: 12.2–12.4
November 1	Raytracing		Fundamentals: Ch 10
November 6	Raytracing		
November 8	Raytracing (and Pathtracing)	Asgn # 8 Due	Fundamentals: Ch 23
November 13	(Advanced Topic)		
November 15	(Advanced Topic)	Asgn # 9 Due	
November 20	**Thanksgiving Break**		
November 22	**Thanksgiving Break**		
November 27	(Advanced Topic)		
November 29	(Advanced Topic)	Asgn # 10 Due	
December 4	(Advanced Topic)		
December 6	(Advanced Topic)		
December 11	(Advanced Topic)		
December 13	Review for Final Examination	Asgn # 11 Due	
December 18	**Final Exam (Tuesday at 2:15-4:15 pm)**		

Some potential topics for the last weeks of class include: modern (programmable) graphics hardware, Bezier curves, radiosity, advanced raytracying or pathtracing, non-photorealistic rendering, and scientific visualization.