Fall 9-1-2000

CS 446.01: Computer Graphics

Alden H. Wright
University of Montana - Missoula, alden.wright@umontana.edu

Computer Science 446
Computer Graphics.
Fall semester, 2000
Course description and Syllabus

Instructor: Alden Wright.

Office:	407 Social Science
Tentative Hours:	1:10 am – 2:30 pm on Mondays, Wednesdays, Fridays.
	Feel free also to drop in to see if I am in my office
	(and not occupied talking to someone else).
E-mail:	*wright@cs.umt.edu*
Phone:	243–4790

Prerequisite: Math 221 and CS 331. (A knowledge of basic C would also be helpful.) If you lack the prerequisites, you should see me to discuss whether or not you should take the course.

Required text:

- **Interactive Computer Graphics: A top-down approach with OpenGl** (Second Edition) by Edward Angel, (Addison Wesley).

Optional supplementary books: (not required)

- If you don't have a reference book on C, you should buy one.
- It would also be helpful to have a reference book on Unix.
- **OpenGL Superbible: The complete guide to OpenGL programming for Windows NT and Windows 95**, by Richard S. Wright and Michael Sweet, (Waite Press). This does not cover the Glut library used in this class.
- **Programming OpenGL for the X Window System**, by Mark Kilgard (Addison Wesley). This book includes both a complete tutorial and documentation for GLUT, as well as a full explanation of using OpenGL with the X Window System's Xlib and Xt/Motif interfaces. This is an excellent book.
- **Open Geometry, OpenGL + Advanced Geometry** by Georg Glaeser and Hellmuth Stachel (Springer).

This course will use a top-down approach to computer graphics.

In a "traditional" bottom-up graphics, one starts with algorithms to draw (rasterize) a line and a circle. Then one does algorithms for clipping and transformations. Maybe at the end of the course the students can draw a wire-frame 3d picture.

Under the top-down approach, we will start with the most commonly used graphics API, namely OpenGL. Using the tools of OpenGL and related libraries, we will be able to do 3D programs with hidden surfaces

and animation by the middle of the semester. Later, we may look at some of the lower-level implementation algorithms (Chapter 7 of Angel).

Objectives:

At the end of the course, students should be able to:

- Write 3D graphics programs using OpenGL.
- Understand the components of common graphics architectures.
- Understand and discuss the strengths and weaknesses of alternative lighting and shading models, such as z-buffer, ray-tracing, and radiosity.
- Understand and do problems on linear and projective transformations as applied to computer graphics.
- Understand and use hierarchical models in graphics programming.
- Understand and use discrete techniques in graphics programming. Discrete techniques include texture mapping, pixel operations, and compositing techniques.

Evaluation and assessment:

- Readings as assigned with the homework.
- Homework exercises and programming assignments.
- Possible group projects.
- Two midterm exams.
- A final examination.

Probable Grading plan:

Assignments and projects:	50%
Exams:	50%

Programming environments: Programs in this course will be done using the OpenGL API and the GLUT library (consistent with the Angel text). OpenGL/GLUT programs are usually portable across Unix, Linux, and Windows32. A separate handout will describe implementatations of OpenGL.

Programming assignments will be done in C or C++. Basic straightforward C without much use of pointers is used, so you should be able to learn C as you do the assignments. Assignments can be done on any of the Solaris, Linux, or Windows32 Operating systems. (MacOS or BeOS may also be possible. See me if you want to use one of these.)

The "definitive" C implementation used for grading assignments will be Mesa on the Solaris workstations.

Class material is one the web at `http://www.cs.umt.edu/u/wright/446/446.html` and in the area: `/class/CS446` on the Solaris cluster. There are a large number of OpenGL demonstration programs at `/class/CS446/Mesa-3.2.1`, especially in `/class/CS446/Mesa-3.2.1/glut-3.7/progs`.

E-mail your programming assignments to: cs446@cs.umt.edu. The subject should be of the form "CS 446, Assignment 1, <your name>" If your program is a single text file, then it can be the entire body of the mail message. If your program includes more than one file, then please either send a gzipped tar file or a zip file as an attachment. Do not include executable files or object files. Do include a `Makefile` if your program contains more than one file. When your gzipped tar file or zip file is extracted, it should create a subdirectory whose name is your name or userid, and which contains your files.

Your programs must include your name, the course, and the assignment number as comments.

Some assignments may be done with assigned teams.

It will be possible to do a project in place of some of the later assignments.

Students will be required to sign the statement on collaboration and cheating.

Incompletes and late drops will be given only for doctor-verified illness, death in the imediate family, and other reasons of similar consequence. The last day to drop/add with a partial refund is Sept. 25, and the last day to drop without a grade penalty is October 16.

Syllabus: (for Fall 2000)

1. Graphics systems and models (Chapter 1 of Angel)
2. Graphics Programming (Chapter 2 of Angel)
3. Input and Interaction (Chapter 3 of Angel)
4. Geometric Objects and Transformations (Chapter 4 of Angel)
5. Viewing (Chapter 5 of Angel)
6. Shading (Chapter 6 of Angel)
7. Discrete Techniques (Chapter 9 of Angel)
8. Heirarchical and Object-Oriented Graphics (Chapter 8 of Angel)
9. Curves and Surfaces (if time, Chapter 10 of Angel)