CS 274 Computer Graphics Fall 2008
Course Syllabus

Instructor: Robert R. Snapp, email: `snapp@cs.uvm.edu`, office: 353 Votey, phone: 802–656–0735.

Office Hours: T 1:30–3:00 PM, W 1:30–3:00 PM, F 9:30–11:00 AM, and by appointment.

Lectures: MWF, 3:35 – 4:25 PM in 367 Votey.

Web Page: `www.cs.uvm.edu/~snapp/CS274/`

Description: Graphical representations of two and three dimensional objects on color raster displays. Line generation, region filling, geometric transformations, hidden line and surface removal, and rendering techniques.

Prerequisites: CS 104 or 124, Math 124 (or Math 271), recommended.

Textbooks:

1. Samuel R. Buss, *3-D Computer Graphics: A Mathematical Introduction with OpenGL*, Cambridge University Press, Cambridge, UK, 2003, ISBN 0–521–82103–7.
2. Dave Shreiner, Mason Woo, Jackie Neider, and Tom Davis, *OpenGL Programming Guide*, 5th (2006) or 6th (2008) edition, Addison-Wesley, Reading, MA. (*The Red Book*.) Earlier editions should suffice.
3. A reference on C or C++ programming is recommended.

Grading Policy: The course grade will be based on approximately six homework assignments (33%), two in-class midterm exams (33%), and a term project (33%). In accordance with the *Graduate Catalogue*, students taking this course for graduate credit are required to achieve higher quantitative scores than undergraduates to receive corresponding grades. Thus, letter grades will be assigned as follows:

Undergraduate	Graduate	Letter Grade
80 – 100	90 – 100	A– to A+
70 – 79	80 – 89	B– to B+
60 – 69	70 – 79	C– to C+
50 – 59		D– to D+
0 – 49	0 – 69	F

Homework: Assignment will often include one or more programming tasks using OpenGL, and a few short exercises. Programs can be written in either C or C++. Programs will be graded on correctness and style, i.e., clarity, robustness, and efficiency.

Midterm Exams: Two midterm exams will be given in class: the first, on Friday, ***October 24***; and the second, on Friday, ***December 5***. The material covered on each exam will be announced at least one week in advance.

Course Projects: Each student is required to complete a term project that consists of independent investigation of an approved topic in computer graphics. Each project contains three parts: a proposal, a 15–30 page term paper (double spaced), and an oral presentation.

PROJECT MILESTONES

1. On Friday, ***October 10***, a 2–3 page, printed proposal is due. (Documents generated by LaTeX are preferred, especially if they contain mathematical expressions.) The proposal should describe your topic and its relation to computer graphics; summarize what you have accomplished to date; describe what you intend to accomplish during the remainder of the semester; and cite at least three relevant *scholarly* publications. The latter should be archival references: books, journal articles, or conference proceedings. (Wikipedia is *not* a scholarly publication.) (10%);
2. On, Friday, ***November 21*** the first draft of the term paper is due. The paper should contain a review the graphics method that you are studying, and should cite relevant publications. The graded first drafts will be returned by Monday, December 1. (10%)
3. During the final exam period (Friday, ***December 19***, 11:45 AM – 2:45 PM) each student is required to deliver a 20 minute oral presentation to the class, describing the outcome of the project. (30%)
4. Also, on ***December 19***, a 10–25 page typeset report is due. It should describe your project in detail. (The suggested page range is exclusive of code: if you wrote some original software, please include it in a separate appendix.) The report should be clearly written, and will be graded on originality, effort, correctness (including spelling and grammar), and clarity. (50%)

A broad range of topics are acceptable. An original computer simulation or a detailed analysis of an algorithm published elsewhere is a desirable goal. Here are a few ideas to get you started:

1. Study and describe photon mapping algorithms (see Jensen, 2001).
2. Describe how computer graphics can be applied to visualize, four and higher dimensional mathematical objects (cf., Banchoff, 1996).
3. Study and simulate one or more model-based, hidden surface removal algorithms (cf., Foley et al., 1996).
4. Study and simulate algorithms that use fractals to generate realistic imagery, such as clouds, trees, and mountain ranges, (cf., Ebert et al., 2003, and Mandelbrot 1983).
5. Study, describe, and simulate simple algorithms that render a three-dimensional model non-photo-realistically, for example, as pencil sketch, or watercolor painting.
6. Study, describe, and simulate algorithms that generate three-dimensional textures.
7. Study, describe, and simulate how one can model crumpled paper, and draped fabrics.
8. Study how "levels of detail" can be used to render complex geometric models efficiently. (See Luebke et al., 2002.)

Late Assignments: Any work turned in late, without a valid excuse, will be penalized 10% credit each *calendar* day. (Note: this includes Saturdays and Sundays.)

Students entitled to special accommodation must notify the instructor by the second week of the semester.

Computer Accounts: Each student should have an EMCF computer account Programming assignments should be written in C or C++, and should run on either a Linux workstation (e.g., Votey 229), or on Windows. Complete source code, data, makefiles, etc., should be e-mailed as a tarball attachment to the instructor before the due date. Assigned programs will be untarred, recompiled and tested as part of the grading. Thus, include a `makefile` (and/or a `read.me` file that describes how to compile and link your programs).

Collaboration: You are *encouraged* to share your knowledge, discoveries, and ideas with other students outside of class. However, all work (e.g., ideas, opinions, analyses, algorithms, data, and source code) generated by others should be properly cited, preferably with an archival source (e.g., a printed book or a peer-reviewed article). In general, sources that appear only on the internet (such as *Wikipedia*) are not sufficient. All non-original text passages should either be placed in quotation marks, or typeset as single-space text, and the original source cited following the conventions of either the *MLA Style Manual and Guide to Scholarly Publishing*, or the *Chicago Manual of Style*.

N.B. Strong similarity between the solutions or programs of two or more students will be treated as a major violation of academic honesty.

Do not plagiarize. Do not cheat. Do not collude. Do not fabricate.

These remarks apply to graded work in this course, including written homework assignments, computer programs, mathematical solutions, course projects and term papers.

Absolutely no collaboration or unauthorized material is allowed during any quiz or exam. All violations will be forwarded to the University Coordinator of Academic Honesty, following the *new* policy of Academic Integrity posted at

`www.uvm.edu/~uvmppg/ppg/student/acadintegrity.pdf`

The first deliberate violation of academic integrity by an undergraduate normally results in a course grade of XF; the second, with a second XF and expulsion.

Etiquette: If you own a laptop computer, you may find it useful on occasion to bring it to class. However, your computer activity must be relevant to the class or lab activity. The the use of cell phones or other personal electronic devices (MP3s, iPods, radios, etc.) is not allowed. Absolutely no calculators, laptops, phones, or other electronic devices are allowed during quizzes or exams.

Religious Holidays: An official policy of the University of Vermont states:

> Students have the right to practice the religion of their choice. Each semester students should submit in writing to their instructors by the end of the second full week of classes their documented religious holiday schedule for the semester. Faculty must permit students who miss work for the purpose of religious observance to make up this work.

TOPICS

(*The following topics are subject to change.*)

1. Introduction.
2. The OpenGL API.
3. Scanline conversion algorithms, and polygon filling.
4. Antialiasing.
5. Affine transformations: translation, rotation, scaling, shear, and reflection.
6. View volumes, parallel and perspective projections, clipping.
7. Hidden surface removal.
8. Lighting and shading models.
9. Texture mapping.
10. Ray tracing.
11. Color models.
12. Polyhedral models.
13. Bézier Splines.
14. B-Splines.

References: There are many useful reference books dedicated to computer graphics. This number is justified by the rapid evolution of this subject, and its interdisciplinary nature. Many books, based on specific platforms, are quickly dated. The following is my personal list of favorites, and is certainly not complete. Many of them are in the Bailey-Howe Library. The rest are available through interlibrary loan.

Also note that there are several important journals and conferences, most notably *Computer Graphics* (which include the proceedings of the annual ACM SIGGRAPH Conferences), the *ACM Transactions on Graphics*, and *IEEE Computer Graphics and Applications*. (Information is also available via internet news and web servers.)

General textbooks on computer graphics.

1. Edward Angel, *Interactive Computer Graphics*, Second Edition, Addison-Wesley, Reading, MA, 2000.
2. Donald Hearn & M. Pauline Baker, *Computer Graphics: C Version*, 2nd edition, Prentice-Hall, Upper Saddle River, New Jersey, 1997. A comprehensive introduction to computer graphics, that includes many important advanced topics as well.
3. James D. Foley, Andries van Dam, Steven K. Feiner, & John F. Hughes, *Computer Graphics: Principles and Practice*, 2nd Edition in C, Addison-Wesley, Reading, MA, 1996 (ISBN 0–201–84840–6). The most comprehensive book on computer graphics available; more advanced than Hearn & Baker.
4. F. S. Hill, Jr., *Computer Graphics using OpenGL*, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 2000, ISBN 0–02–354856–8.
5. Tomas Möller and Eric Haines, *Real-Time Rendering*, Second Edition, A. K. Peters, Natick, MA, 2002.
6. David F. Rogers, *Procedural Elements for Computer Graphics*, Second Edition, McGraw-Hill, New York, NY, 1997.
7. David F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics*, McGraw-Hill, New York, NY, 1990.
8. Peter Shirley, *Fundamentals of Computer Graphics*, second edition, A. K. Peters, Wellesley, MA, 2005.

9. Alan Watt, *3D Computer Graphics*, Addison-Wesley, Wokingham, England, 1993 (ISBN 0–201–63186–5).
10. Alan Watt and Fabio Policarpo, *The Computer Image*, Addison-Wesley, Reading, MA 1998 (ISBN 0–201–42298–0). Explores the relationships between computer graphics, image processing, and computer vision. An up-to-date survey, and a good source for term project ideas.

The *Graphics Gems* series. Books in this series contain submitted articles that describe various tricks of the trade for the "working graphics programmer." (Maybe you'll contribute to a future edition.)

1. Andrew S. Glassner, ed., *Graphics Gems*, Academic Press, 1990 (ISBN 0–12–286166–3).
2. James Arvo, ed., *Graphics Gems II*, Academic Press, 1991 (ISBN 0–12–0644–80–0).
3. David Kirk, ed., *Graphics Gems III*, Academic Press, 1992 (ISBN 0–12–409673–5).
4. Paul S. Heckbert, *Graphics Gems IV*, Academic Press, 1994 (ISBN 0–12–336156–7).
5. Alan W. Paeth, *Graphics Gems V*, Academic Press, 1995 (ISBN 0–12–543455–3).

Some useful programming references

1. Adobe Systems Incorporated, *The Postsript Language Reference*, Third Edition, Addison-Wesley, Reading, MA, 1999.
2. Mark Kilgard, *OpenGL Programming for the X Window System*, Addison-Wesley, Reading, MA, 1996.
3. Stanley B. Lippman and Josée Lajoie,*C++ Primer*, Third Edition, Addison-Wesley, Reading, MA, 1998.
4. Paul Martz, *OpenGL Distilled*, Addison-Wesley, Reading, MA, 2006 (ISBN 0–321–33679–8).
5. Glenn C. Reid, *Thinking in Postscript*, Addison-Wesley, Reading, MA, 1990. (Also available on-line.)
6. Bjarne Stroustrup, *The C++ Programming Language*, 3rd edition, Addison-Wesley, Reading, MA, 1997 (ISBN 0–201–88954–4).
7. Dave Shreiner, Mason Woo, Jackie Neider, and Tom Davis, *OpenGL Programming Guide*, 5th edition, Addison-Wesley, Reading, MA, 2006 (ISBN 0–321–33573–2).

More specialized books.

1. Thomas Banchoff, *Beyond the Third Dimension: Geometry, Computer Graphics, and Higher Dimensions*, W. H. Freeman, NY, 1996.
2. Michael F. Barnsley, *Fractals Everywhere*, Second Edition, Academic Press, Boston, MA, 1993.
3. Richard H. Bartels, John C. Beatty, and Brian A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan-Kaufmann, San Francisco, 1987.
4. Jim Blinn, *Jim Blinn's Corner: A Trip Down the Graphics Pipeline*, Morgan Kaufmann, San Francisco, 1996 (ISBN 1–55860–387–5). A collection of entertaining and informative articles that originally appeared in the journal *IEEE Computer Graphics and Applications*.
5. David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley, *Texturing and Modeling: A Procedural Approach*, Third Edition, Morgan Kaufmann, San Francisco, 2003.
6. Jean Gallier, *Curves and Surfaces in Geometric Modeling*, Morgan-Kaufmann, San Francisco, CA, 2000.
7. Andrew S. Glassner, ed., *An Introduction to Ray Tracing*, Academic Press, 1989.
8. Andrew S. Glassner, *Principles of Digital Image Synthesis*, 2 volumes, Morgan-Kaufmann, San Francisco, 1995 (ISBN 1–55860–276–3).
9. Henrik Wann Jensen, *Realistic Image Synthesis using Photon Mapping*, A. K. Peters, Natick, MA, 2001.
10. D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Huebner, *Level of Detail for 3D Graphics*, Morgain-Kaufmann, San Francisco, CA, 2002.
11. Benoit B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman, New York, NY, 1983.
12. Matt Pharr and Greg Humphreys, *Physical Based Rendering*, Morgan-Kaufmann, San Francisco, 2004.
13. Peter Shirley, *Realistic Ray Tracing*, A. K. Peters, Natick, MA, 2000.
14. Alan Watt & Mark Watt, *Advanced Animation and Rendering Techniques: Theory and Practice*, Addison-Wesley, Wokingham, England, 1992 (ISBN 0–201–54412–1).
15. Rosalee Wolfe, ed., *Seminal Graphics: Pioneering Efforts that Shaped the Field*, ACM Sigraph, 1998.

Writing references.

1. Joseph Gibaldi, *MLA Handbook for Writers of Research Papers*, 6th edition, Modern Language Association of America, New York, 2003.
2. Helmut Kopka and Patrick W. Daly, *Guide to LaTeX*, 4th edition, Addison-Wesley, Boston, 2004.
3. William Strunk Jr. and E. B. White, *The Elements of Style*, 4th edition, Allyn & Bacon, Boston, 2000.
4. University of Chicago Press Staff, *The Chicago Manual of Style*, 15th edition, University of Chicago Press, Chicago, 2003.