CS 480/680 Computer Graphics

Spring 1999

Prerequisites: This course provides an introduction to two dimensional and three dimensional computer graphics. In order to understand the material in this course, you will need to have a good working knowledge of data structures such as linked lists, trees, symbol tables, and dynamically allocated structures. You will also need to have some knowledge of 2D and 3D geometry of vectors and coordinate systems, and of matrix algebra for coordinate transformations (we will review this material in class). To successfully complete this course, you should be familiar with material related to the above topics covered in CS308 (Data Structures) and in MATH182 (Calculus II). If you do not meet the prerequisite requirements for this course, you should see me immediately.

Credit hours: 3.0

Instructor: Dr. George Bebis
Office: 310 LME
Phone: 784 - 6463
E-mail: bebis@cs.unr.edu
URL: http://www.cs.unr.edu/~bebis/CS480
Office Hours: TR: 2:30 - 4:00 pm and by appointment.

Required Texts:

Introduction to Computer Graphics by Foley et al., (Addison-Wesley, 1994).
OpenGL Programming Guide, by M. Woo, J. Neider, and T. Davis (Addison-Wesley, 1997)

Optional Texts:

Computer Graphics by Hearn and Baker (Prentice-Hall, 1997).
Interactive Computer Graphics by Angel (Addison-Wesley, 1997)
Procedural Elements for Computer Graphics by D. Rogers (McGraw-Hills, 1998).

Objectives

The goal of this course is to convey understanding of the process of modeling and generating images of 3D objects. We will start by studying the basic process of drawing primitive objects on a display (lines, circles, polygons). We will then look at the process of building two and three dimensional mathematical models of more complex objects, manipulating and combining these models, and projecting the models onto a two dimensional image space.

Course Outline

- Introduction (Chapter 1)
- Graphics Input-Output (logical input devices, from Chapter 2)
- Line Scan algorithms, Polygon Filling, Clipping (Chapter 3)
- Geometrical Transformations (Chapter 5)

- Projections (Chapter 6)
- Visible-Surface Determination (Chapter 13)
- Illumination models (Chapter 16)
- Representation of Curves and Surfaces (Chapter 9)
- Solid Modeling (Chapter 10)

Exams and Assignments

There will be a midterm and a final exam. The material covered by the exams will be drawn from the lectures and from the homework. Both exams will be closed books, closed notes. Homework problems will be assigned every 2-3 weeks and will be collected at the beginning of the class on the due date. Solution sets will be provided for all the problems assigned. Also, there will 5 programming assignments. Programming may be done on any departmental workstation or on other machines accessible to the student. All programs will involve the use of OpenGL which is available on the SGI machines in Tompson's lab (321 LME). If you prefer to do your programming on the Sun machines, you can use the Mesa graphics library which is a very close emulator of OpenGL. For more information on Mesa, visit the course's homepage at **http://www.cs.unr.edu/~bebis/CS480/**. You can find many useful links in the course's homepage and I am going to add more things during the semester. Make sure that you visit it at least 1-2 times per week for useful announcements and new stuff. Also, download the **tutorial on OpenGL**.

You may work in teams of two to complete the projects, however, all the members of the team is expected to fully understand the structure of the code and the implemented algorithms. Discussion of programming assignments is allowed and encouraged. I view this class as a team whose goal is to learn the basic ideas of 3D graphics, and learn how to implement these ideas. If you have a problem in the lab with OpenGL, **ask your neighbor**. If you find something useful on the network, **tell someone about it**. If you find a better algorithm in the library, **tell someone about it**. However, each team is expected do its own work. **Assignments which are too similar will receive a zero**.

Grading of Programming Assignments

The grading of the programming assignments will be based on the following criteria: (i) results, (ii) knowledge of concepts, (iii) output, and (iv) documentation. Specifically, the program and its resulting output must solve the problem exactly as stated. Also, the source code must reflect an adequate knowledge of the concepts to be discussed in class. The graphical output must be clearly presented and is a fundamental part of the grading of the assignment. Finally, the source code must be well documented. This requirement does not imply that extremely verbose comments are necessary, but it does imply that the people who read your assignments must be able to understand the essential workings of the program from the documentation. Instructions of how to turn in your programs can be found in the homepage of the course.

Course Policies

Late homework or programming assignments will be penalized 10% of the points assigned per day (weekends count as one day). If you are unable to hand in an assignment by the deadline, you must discuss it with me **before** the deadline in order to avoid the late penalty. If you are unable to attend an exam **you must inform me in advance**. A missed exam may be made up only if it was missed due to an extreme emergency.

Useful Tips

Don't get behind in the programming assignments. Probably the main reason for students doing poorly in this course is getting behind in the assignments and never recovering. Design and implement in a top-down, modular fashion. Get something working that has the skeleton structure of what you need and then add features to it. Each time you add a feature, test it and make sure everything is still working. It can be tough to debug graphics programs if all you know is that the output is wrong and you are not sure any one module is working. In addition, partial credit will be given for a program which at least partially works while it is very difficult to give credit for a program which may have many features but is not displaying anything correctly.

Grading Scheme

Midterm Exam: 25%
Final Exam: 25%
Homework: 10%
Programs: 40%

A 90 and above
B 80-89
C 70-79
D 60-69
F < 60

Important dates

3/4/99 - Midterm exam
3/12/99 - last day for dropping classes
3/13/99 - 3/21/98 - no classes (Spring Break)
5/5/99 - no classes (preparation for final exam)
5/11/99 - Final exam (4:30 pm - 6:30 pm)