CSC 406 – Computer Graphics
Geometry, Modeling, and Animation

Lectures
Tyler Hall Rm 226
Tuesday, 6:00 – 7:15 PM
Tyler Hal, Envision Lab
Tuesday, 7:30 – 8:45 PM

Instructor
Jean-Yves Hervé
Tyler Hall, Room 252
Tel.: 874 4400 or 874 2701
email: jyh@cs.uri.edu

Office hours:
Monday, 1:30 – 3:00 PM,
Wednesday, 3:00 – 4:30 PM

Teaching Assistant
Clifford Manzanillo
Tyler Hall, Room 141
Tel.: 874-2575
email: cman4471@postoffice.uri.edu

Office hours:
Monday, 3:00 – 4:00 PM
Wed., 2:00 – 3:00 PM

Textbook
Our textbook only covers the mathematical aspects of the course:

Eric Lengyel
Mathematics for 3D Game Programming & Computer Graphics,

The following booklet
Edward Angel
gives a good introduction to OpenGL and glut. In fact, I think that it does a better job of it than the classical and official “Red Book” (OpenGL Programming Guide) published by Addison-Wesley. There are many good books on OpenGL programming, but if you want to learn OpenGL in depth, nothing will replace the “Blue Book” (OpenGL Reference Guide) also published by Addison-Wesley.

Prerequisites
CSC 212, MTH 215 and 243

I know that the URI catalog lists CSC305 as a prerequisite, but CSC212 is the real CS prerequisite for this course.

It might come to you as a shock that linear algebra is used a lot in computer graphics. It would be a good idea for you to refresh your (fond) memories of matrices in the next few days.

MTH243 material will be needed when we start talking about parametric models of curves and surfaces, and kinematics (animation).
Objectives
This course will concentrate exclusively on the geometric and kinematic aspects of computer graphics. At the end of this course, you should be capable of

- Constructing a geometric model of a simple scene;
- Implementing the objects of this scene, using OpenGL, glut, and glu primitives;
- Controlling, interactively or automatically, the displacements of objects, including the “camera(s)” that capture the scene;
- Building a simple graphical user interface for your application, using the glut library.

Other aspects of computer graphics are covered in the following courses
- ART304 and ART306: 3D Graphics design (using Maya 6)
- CSC492M-592R: scene graphs, scene rendering, GUI.

Graphic API
In your future careers as computer scientists and programmers, you will often have to learn a new API (Application Programming Interface) and use it in your projects. This course offers us a good opportunity to develop this important skill.

We will be using OpenGL + glut. This API developed by SGI is the most standard cross-platform API used both for professional 3D editing applications (Maya, SoftImage) and computer games (Quake II & III, Descent 3, etc.). On some Unix platforms, what is found is not OpenGL per se but MESA, which is an open-source project (encouraged by SGI) that offers source-level compatibility with OpenGL.

IDE & Language
All assignments that you return must be under the form of a complete and working Metrowerks CodeWarrior project. CodeWarrior Pro 8.3 is installed on all computers of the enVision Lab.

You can use either C/C++ or Java as your programming language. Things might be a bit easier in C/C++, because C/C++ is really the “native” language of OpenGL, but if you choose to program in Java, you will be able to use the GL4Java library and package, which maps most of the OpenGL and glut functions.

Finally, you should try as much as possible to avoid using platform-specific calls in your code. For example, use glut rather than MFC (on Windows) or Motif (on a Unix) for creating and managing windows and handling events.

Grading
The following coefficients will be used to compute the final grade:

- Homework assignments 15%
- Programming assignments 35%
- Project 25%
- Midterm 25%
Examinations

One midterm examination will be administered on Tuesday, October 26 at the usual class hours. All notes and printed documentation will be allowed during this examination.

Lab Assignments

This semester, the class will be divided into a first half of formal teaching (on the theoretical aspects of the material) and a second half to be held in the Envision Lab. This second half will deal with all practical aspects of the course. We will hand out a programming assignment a few days ahead of the lab. You are not expected to hand out anything at the end of the lab, only to read the assignment, start working on it, and ask questions during the session to make sure that you are on the right track and understand what you should be doing. Complete assignments (code and report) are due at the beginning of the next class, unless explicitly specified in the assignment or on the course’s web page.

Other Assignments

I will hand out regularly homework assignments as well.

Final Project

You will be expected to propose a topic for a final project by Nov. 09. Examples of past projects and topic suggestions will be provided during the semester. The purpose of this project is for you to demonstrate that you can use the concepts and tools seen during the semester to develop a complete application.

Late Submissions

All assignments should be turned in on the day and time they are due. If an assignment is not turned in on time, a penalty will be applied for each late day according to the following table:

- One day late: 10%
- One week late: 40%
- Two weeks late: 60%

Assignments more than 2 weeks late will not be graded.

Cheating

All assignments will state explicitly the maximum group size allowed. You are definitely encouraged to discuss the assignments with other students (from other groups), and try to figure them out together, but when comes the time of writing a report or developing code, you are expected to do it by yourself or with other students from your group. Cheating includes copying all or part of the code or report of another group. It is still copying and cheating if you perform some local and/or global substitutions and formatting and will result at the very list in a 0 grade for all parties involved, and possibly in expulsion from the class with a failing grade (and will be brought to the attention of the Dean for disciplinary action).

Copying software from a book or from a web site without identifying the original author(s) of the work is plagiarism, which is a form of cheating and will be dealt with accordingly.
Graphics Lab  Access to the computer graphics lab will be available for all students registered in the course during the usual hours of the EnVision Lab

Laboratory use  Any abuse of computer or software equipment will be brought to the attention of the appropriate authority for disciplinary action.
Software piracy (the use, appropriation, or storage of illegal copies of software) is a form of abuse of the equipment and will be treated accordingly.

Class Schedule

1  Overview
   • classes of graphical systems
   • display architecture
   • OpenGL and glut

2.  Basic 2D and 3D geometry
   • linear algebra
   • rotation, translations, scaling
   • matrix models
   • homogeneous representation
   • 3D transforms

3.  Viewing
   • projections
   • multiple viewpoints
   • hidden surfaces

3.  Animation of simple rigid objects
   • path vs. trajectory
   • point kinematics
   • kinematics of solid objects
   • composition of movements

4  3D curves and surfaces
   • continuous vs. discrete models
   • parametric models of curves and surfaces
   • mesh surfaces
   • normal vectors

6  Object models
   • hierarchies
   • mathematical models
   • splines and NURBs

10  Animation
   • key framing
   • forward kinematics
   • inverse kinematics