CS334: Fundamentals of Computer Graphics

Instructor: Daniel G. Aliaga
Classroom: Lecture - LWSN 1106, PSO – HAAS G056
Time: Lecture MWF @ 10:30-11:20am, PSO W @ 1:30-3:20pm, F @ 3:30-5:20pm
Office hours: by appt
TA: Carlos Vanegas (cvanegas@cs.purdue.edu)

Interested in computer graphics? Does modeling objects interest you? Do you like rendering photorealistic imagery? Is doing animations fun to you? All this is part of computer graphics. This course teaches the fundamentals, at an undergraduate school level, for such activities and research projects. Major applications include:

- Virtual Reality
- 3D Scanning
- Video Games
- Film Special Effects
- CAD/CAM
- Simulation
- Medical Imaging
- Image Processing
- Scientific Visualization
- Information Visualization

1. Prerequisites

Students are required to have previous C/C++ programming experience. Knowledge of linear algebra is strongly recommended. Previous computer graphics experience, such as OpenGL programming experience, is beneficial but not mandatory. OpenGL will be implicitly used in the course: a review of OpenGL will be indirectly given during first half of the semester.

2. Course work

The course work is composed of programming assignments, exams, and interactive class participation. The programming assignments consist of a warm-up assignment, three programming assignments and a final assignment. The exams consist of a midterm and a final exam. Class participation will consist of active participation during class (you be called upon). Course work will be easier to manage if you keep a constant pace through the semester. This course is hard work but you will learn a lot and have fun!

3. Grading

Programming Assignments:	30% (assignments 0-3)
	30% (final assignment)
Class Participation:	10% (active participation)
Exams:	15% (midterm)
	15% (final)

	100% TOTAL

4. Lecture Schedule

Basics
Jan 7 – Introduction to Computer Graphics (History)
Jan 9 – CGVLab and Research Summary
Jan 11 – Vectors, points, matrices, coordinate systems, transformations I

Linear Algebra
Jan 14 – Vectors, points, matrices, coordinate systems, transformations II
Jan 16 – 3D->2D Projections I
Assignment #0 out (1 wk: OpenGL/GLUT/GLUI)
Jan 18 – 3D->2D Projections II

2D Processing
Jan 21 – Holiday
Jan 23 – TBA
Jan 25 – 2D Image processing: rotating, translating, scaling
Assignment #0 in
Assignment #1 out (2 wks: 3D point, vector, matrix operations and xforms)

2D Processing
Jan 28 – 2D Image processing: affine and perspective warping, inferring the transformation
Jan 30 – Line Rasterization
Feb 1 – Triangle Rasterization

Shading and Lighting
Feb 4 – Lighting models (Gouraud)
Feb 6 – Lighting models (Phong)
Assignment #1 in
Assignment #2 out (2 wks: Shading and Lighting)
Feb 8 – Other illumination models

Geometric Thinking
Feb 11 – Geometry/Triangle Computations I
Feb 13 – Geometry/Triangle Computations II
Feb 15 – TBA

Collisions
Feb 18 – TBA
Feb 20 – Collision Detection Basics
Assignment #2 in
Assignment #3 out (3 wks: Collision Detection and Animation)
Feb 22 – Animation and Interpolation

Review and Midterm
Feb 25 – Review I

Feb 27 – Midterm
Feb 29 – Solutions

Ray Tracing
March 3 – Ray Tracing Basics
March 5 – Ray Tracing Advanced
March 7 – Ray Acceleration Schemes

Spring Break
March 10 – Spring Break
March 12 – Spring Break
March 14 – Spring Break

Object Representations
March 17 – Points, Polygons, Triangles, Curved Surfaces
March 19 – Subdividing Triangles
Assignment #3 in
Final Project out (4 wks)
March 21 – Manifolds and Meshing

Simplification Basics
March 24 – Vertex Clustering
March 26 – Edge Collapse
March 28 – Simplification Trees

Spatial Hierarchies
March 31 – Top-down Spatial Subdivisions
April 2 – Bottom-up Bounding Volume Hierarchies
April 4 – TBA

GPU Programming
April 7 – Fragment Shaders
April 9 – Pixel Shaders
April 11 – Non-standard uses of the GPU

Special Topics
April 14 – TBA
April 16 – TBA
April 18 – TBA

Review and Demos
April 21 – Review I
April 23 – Review II
April 25 – Demo Day!

Final Exam – TBA

5. Assignments

You may use CS lab computers or home computers. Assignments must be written in C/C++ on a Windows computer. Assignments are due before class time on the due date and must be emailed handed-in via Purdue's Blackboard Vista, including all source code, data files, and an already compiled program. The grading for the assignment will consider *functionality* and *form*. All assignments must be polished products, with a well designed user interface (if appropriate) and clean, reliable functionality. A program that does not compile obtains 0 points.

Assignment #0 – OpenGL/GLUT/GLUI (1 week). Implement a short OpenGL+GLUT+GLUI program that implements a simple screen-saver style program. The program will open up a window, display a GLUI interface to choose the screen-saver option, and draw a simple 2D screen saver in the main window. The purpose of this program is to get a working project setup. The only OpenGL command you may use is glDrawPixels(). *If you are already well versed in Windows GUI programming, you may use that instead of GLUI but only upon approval by the instructor*.

Assignment #1 – Linear Algebra (2 weeks). Implement a 3D point class, a 3D vector class, a 4x4 matrix class, and the transformations of rotate, scale, translate, and perspective projection. Illustrate your classes by performing these operations on simple object primitives (points, squares, lines). The only OpenGL command you may use is glDrawPixels().

Assignment #2 – Shading and Lighting (2 weeks). Implement a program that uses OpenGL to rasterize, shade, and light a simple 3D scene using GLUT primitives (spheres, cylinders, etc). You will shade/illuminate them using several standard methods (e.g., Gouraud, Phong). The scene is static but the viewpoint is dynamic.

Assignment #3 – Collision Detection and Animation (3 weeks plus Spring Break). Implement a program that draws several object primitives bouncing within a box-shape volume. Enable specifying the animation using either constant-velocity integration or constant-acceleration integration.

Final Assignment (4 weeks). Projects will be presented on a publicly attended "demo day" at the end of the semester (April 25th). A competition will be held for each project type with the best two receiving bonus points:

a) Implement the fanciest ray-tracer you can.
b) Implement the fanciest simplification-engine you can to render large models.

Grading: the final assignment must be a polished product, with a well designed user interface and clean, reliable functionality.

7. Exams

The midterm will cover material explained in class, stressing fundamentals. The final exam will cover material of the entire semester and will stress understanding of general interactive

computer graphics and its fundamentals. Both are closed book and will require “understanding and imagination” rather than memorization of formulas.

8. Administrative Issues

Late policy
Assignments are due before class on due date. First time late – no penalty for up to one week, but instructor must be notified via email BEFORE deadline (*if instructor not notified via email before assignment due date, late pass cannot be used and assignment will be late*). Second and subsequent times -- grade reduction of 20% per day. All assignments required by demo day at end of course or failing grade will be issued.

Collaboration
All assignments and exams must be done individually. Copying or plagiarism will give you a failing grade in the course and you will be subject to departmental and University policies. Code obtained from the Internet, books, or other sources may *not* be used for any assignment. Exceptions allowed only under explicit instructor approval.

Teaching Assistant
Carlos Vanegas is the graduate TA (cvanegas@cs.purdue.edu).