

CS334: Fundamentals of Computer Graphics

Instructor: Daniel G. Aliaga

Classroom: Lecture – HAAS G066, PSO – HAAS G056

Time: Lecture Tue/Thur @ 9:00-10:15am, PSO Mon @ 9:30-11:20am, Tue @ 1:30-3:20pm

Office hours: by appt

TA: Nathan Andrysc (nandrysc@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. In-class participation will consist of presenting a preview of your final assignment (schedule arranged later). 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:	35% (assignments 0-3)
	30% (final assignment)
In Class Presentations:	5% (preview of final assignment)
Exams:	15% (midterm)
	15% (final)

	100% TOTAL

4. Lecture Schedule

Basics I

Aug 26 – Overview/Example Research in Computer Graphics

Aug 28 – Vectors, Points, Matrices

Assignment #0 out

Basics II

Sept 2 – Vectors, Points, Matrices

Sept 4 – Perspective and Other Projections

Assignment #1 out

2D Processing and Graphics Pipeline

Sept 9 – 2D Image Processing

Assignment #0 due (extension from Sept 4th because Sept 1st PSO on holiday)

Sept 11 – Graphics Pipeline, Rasterization

3D Processing and Shading

Sept 16 – Shading and Lighting

Sept 18 – Shading and Lighting, Texture Mapping

Assignment #1 due, Assignment #2 out

3D Processing and Shading

Sept 23 – Animation and Collisions

Sept 25 – Spatial Hierarchies

Ray Tracing I

Sept 30 – Ray Tracing

Oct 2 – Review

Assignment #2 due, Assignment #3 out

Midterm

Oct 7 – Midterm Exam

Oct 9 – Midterm Solutions

Ray Tracing II

Oct 14 – Fall Break

Oct 16 – Ray Tracing

Procedural Modeling

Oct 21 – Procedural Modeling

Oct 23 – Procedural Modeling

Assignment #3 due, Final Assignment out

Simplification

Oct 28 – Large-scale Modeling/Urban Modeling

Oct 30 – Simplification

Non-photorealistic Rendering

Nov 4 – Simplification, Impostors

Nov 6 – Non-photorealistic Rendering

Object Representations

Nov 11 – Meshing and Other Object Representations

Nov 13 – TBA

Final Project Preview

Nov 18 – Final Project Preview

Nov 20 – Final Project Preview

Nov 25 – Final Project Preview/TBA

Nov 27 – Thanksgiving Break

Final Project

Dec 2 – Review

Dec 4 – Demo Day!

Final Assignment due

Special Topics

Dec 9 – Special Topics I: Visualization (tentative)

Dec 11 – Special Topics II: Computational Geometry (tentative)

Final Exam

Dec 15-20 – 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.

NOTE: The assignments will build-up a software platform for 3D graphics and you should keep this in mind from the beginning – you will be re-using your software so design it well!

Assignment #0 – Warm-up to OpenGL/GLUT/GLUI (1 week). The purpose of this assignment is to get you able to compile ASAP. Implement a short OpenGL+GLUT+GLUI program that implements a simple screen-saver style program. The program will open up a window, display a GLUT 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

drawing command you need to use is that to draw lines (e.g. `glBegin(GL_LINES)`). A short and working OpenGL program will be provided to you as a starting point. *If you are already well versed in Windows GUI programming, you may use that instead of GLUT 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) and implementing a simple 3D navigation tool (i.e., a virtual trackball). The only OpenGL matrix command you can use is `glLoadMatrixf()`, in other words your code must do all the vector/point/matrix operations.

Assignment #2 – Basic Shading and Lighting (2 weeks). Implement a program that uses OpenGL to rasterize, shade, and light a few objects that bounce up-and-down off a flat ceiling and ground plane. The objects are drawn using 3D 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 using the tool of previous assignment.

Assignment #3 – Collision Detection and Texture Mapping (3 weeks plus Fall Break). Extend the previous assignment to a program that draws several object primitives bouncing off each other and within a box-shape volume. Movement is done using a constant-velocity integration as explained in class. Also, enable specifying textures to be mapped onto the objects.

Final Assignment (4 weeks). Projects will be presented on a publicly attended “demo day” near the end of the semester (Dec 4th). The project options are:

- a) Ray Tracer: Generate images using ray tracing.
- b) Procedural Modeler: Create models using a procedural modeling method.
- c) Simplification Engine: Develop a simplification algorithm for fast rendering.
- d) Self-Guided: A project of your choosing but approved by the instructor.

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

Nathan Andrysc is the graduate TA (nandrysc@cs.purdue.edu).