CS535: Interactive Computer Graphics

Instructor: Daniel G. Aliaga
Classroom: LWSN B134
Time: TTh @ 9-10:15am
Office hours: TBA

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 a graduate 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 and are recommended to have previous computer graphics experience, such as OpenGL programming experience (although OpenGL will be reviewed at the beginning 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 incremental 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) and the presentation of a mini-review to your classmates. The mini-review will cover material previously covered in class and will serve to help prepare you for the final exam. 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:	20% (assignments 0-3)
	30% (final assignment)
Class Participation:	20% (15% mini-review, 5% active participation)
Exams:	10% (midterm)
	20% (final)

	100% TOTAL

4. Lecture Schedule

August 21 – Introduction
August 23 – Programming at Purdue CS, OpenGL, GLUT, GLUI basics
Assignment #0 out
August 28 – Projections and transformations
August 30 – Cameras and optical aberrations
Assignment #0 due
Assignment #1 out
September 4 – Panoramic and omnidirectional cameras
September 6 – Transformation, rasterization, shading, and lighting
September 11 – Computations with triangles
September 13 – Texture mapping and projective texture mapping
Assignment #1 due
Assignment #2 out
September 18 – Color space and dynamic range
September 20 – Spatial hierarchies
September 25 – Culling
September 27 – TBA
October 2 – Review
Assignment #2 due
Assignment #3 out
October 4 – Midterm
October 9 – Fall break – no classes
October 11 – Midterm Review/Final-Project Preview
October 16 – Ray tracing I: reflections, intersections, efficiency
October 18 – Ray tracing II: higher-order visual effects, anti-aliasing,
Assignment #3 due
Final Assignment out
October 23 – Procedural Modeling I: grammas, plants
October 25 – Procedural Modeling II: cities, buildings
November 8 – Image-based Rendering: image warping and lightfields in a jiffy
October 30 – Level of Detail, Simplification, Cost/Benefit: meshes, polygon soup
November 6 – Level of Detail, Simplification, Cost/Benefit: quality metrics, adaptation
November 13 – Visualization/TBA
November 15 – Detailed review by Students I
November 20 – Detailed review by Students II
November 22 – Thanksgiving – no classes
November 27 – Detailed review by Students III
November 29 – Thanksgiving – no classes
December 4 – TBA
December 6 – Demo Day
Final Assignments due
Final Exam – see University website

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 to the course TA as a single .zip file containing all source code, data files, and an already compiled program. Email time-stamp will be used to verify on time submission. The grading for the assignment will consider *functionality* and *form*. All assignments must be polished products, with a well designed user interface and clean, reliable functionality. A program that does not compile obtains 0 points.

Assignment #0 – Warm up (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 – Camera Projections (2 weeks). Implement a program that draws 2D objects bouncing within a plane but observed from a movable 3D viewpoint. The objects are drawn as simple 2D wireframe objects (e.g., rectangle, triangle, etc.) but they must bounce of the edges of the plane they lie in. The user should be able to specify via the GLUI interface a 3D viewpoint and 3D viewing orientation. Again, you may only use glDrawPixels() – you must implement all matrix operations, projections, and drawing yourself. Extra credit: implement a virtual trackball interface controlled by the mouse.

Assignment #2 – Scene Modeling (2.5 weeks). Implement a program that uses OpenGL to rasterize, shade, and light a 3D scene. We will provide a simple scene file describing several object primitives in a simple text file format. You will load the primitives and shade/illuminate them using several standard methods (e.g., Gouraud, Phong). The scene is static but the viewpoint is dynamic. Extra credit: implement other shading techniques, such as shadow-mapping, and/or support dynamic objects

Assignment #3 – Spatial Subdivision and Culling (2.5 weeks). Implement a program that computes two types of spatial hierarchies of a given 3D model and that uses them to perform efficient culling operations. Using GLUI, allow the user to walk through the spatial hierarchies, visualize them, and render subsets of the triangles in the data structure. .Extra credit/competition: the person that is able to draw the largest model at the fastest frame-rate will win the competition and earn extra credit. The gain of efficiency can be obtained by careful programming and/or by implementing additional techniques/tricks to reduce the rendering load. For the competition, you will be provided with a program that yields an arbitrary size model.

Final Assignment (7 weeks). Projects will be presented on a publicly attended "demo day" at the end of the semester (December 6th). Choose from one of the following:

a) Ray-tracer: use the spatial hierarchies and culling methods to aid in implementing a ray-tracer for 3D objects and at least first-order reflections.

b) Procedural Modeling: implement a 3D procedural modeling method for a particular class of objects.
c) Simplification: extend assignment #3 to some form of geometric simplification (you can be creative), e.g., replace each node with a "box", collapse nodes, use images, etc. I will supply models of varying sizes.
d) Collision detection: allow multiple objects to bounce around the inside of a cube and use a spatial hierarchy and culling to efficiently estimate *accurate* collisions and perform an approximate collision response (i.e. a bounce).
e) Your own project but must be approved beforehand. You must submit it in writing and get it approved *before* the start of the Final Assignment.

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

6. Mini-Review and Class Participation

Students will be called upon during class to explain concepts and solve problems (sometimes in small teams). Towards the end of the semester, each student will also give a ~20 minute review session to the class of a topic covered during the semester. You will be graded on this presentation. It will serve as review and to improve the students understanding of the material. The distribution of topics to students will be performed in a democratic fashion.

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, exams, and review presentations 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 Andrysco is the graduate TA (nandrysc@cs.purdue.edu) . He will hold weekly office hours. Time and location: TBA.