CS535: Interactive Computer Graphics

Instructor: Daniel G. Aliaga
Classroom: CS G066
Time: TTh @ 4:30-5:45pm

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
- Cartoons
- 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.

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)
	20% (final assignment)
Class Participation:	30% (15% mini-review, 15% active participation)
Exams:	10% (midterm)
	20% (final)

	100% TOTAL

4. Lecture Schedule

August 23 – Introduction
August 25 – Review of linear algebra and geometric operations
August 30 – Representations of graphical objects
Assignment #0 out
September 1 – Projections and transformations
September 6 – Cameras
Assignment #0 due
Assignment #1 out
September 8 – Omnidirectional cameras
September 13 – Transformation, rasterization, shading and lighting
Assignment #1 due
Assignment #2 out
September 15 – Colors and displays
September 20 – TBA
September 22 – TBA
September 27 – Spatial hierarchies
September 29 – Culling
Assignment #2 due
Assignment #3 out
October 4 – Simplification
October 6 – Review
October 11 – Fall Break – no classes
October 13 – Midterm
October 18 - Shading and lighting
October 20 - Ray tracing
Assignment #3 due
Final Assignment out
October 25 – TBA
October 27 – TBA
November 1 – Texture Mapping I
November 2 – Texture Mapping II
November 8 – Image Operations
November 10 – Review by Students I
November 15 – Review by Students II
November 17 – Review by Students III
November 22 Review by Students IV
November 24 – Thanksgiving – no classes
November 29 – Advanced Topics I
December 1 – Advanced Topics II
December 6 – Demo Day
Final Assignments due
December 8 – Review
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.

Assignment #0 – Warm up (1 week). Implement a short program that bounces a (2D) wireframe square within the window. The main purpose of this program is to get a working project setup. The only OpenGL command you may use is glDrawPixels() (i.e., you must implement all projection and rasterization code).

Assignment #1 – Basic Wireframe Renderer (1 week)). Implement a program that draws basic 3D objects in wireframe and allows the user to press key commands to translate and rotate the object (6-degrees of freedom). Again, you may only use glDrawPixels() – you must implement all matrix operations, projections, and rasterization yourself.

Assignment #2 – Basic Shaded Renderer (~2.5 weeks). Implement a program that renders solid 3D objects using a single light source and Gouraud shading. Use a painter's algorithm to draw objects (render from back to front), rasterize each triangle, compute pixel colors and draw. The objects should be inside a wireframe cube, move in a random initial direction and then bounce of the inside walls of the cube. Extra credit: implement other shading techniques.

Assignment #3 – Hierarchical Spatial Subdivision (3 weeks). Implement a program that computes the octree of a given 3D model. You are now free to use OpenGL matrix stacks and polygonal rendering abilities. The scene should have at least one light source and be shaded. Using GLUI, allow the user to walk through the octtree and render triangles in the current node in red. Make the object and its octtree bounce interactively. Extra credit: TBA.

Final Assignment. Choose from one of the following:

a) Simplification: extend the octtree data structure 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.
b) Ray-tracer: use the octree data structure to aid in implementing a ray-tracer for basic 3D objects and at least first-order reflections.
c) Occlusion culling: implement an occlusion culling algorithm using the octtree (e.g., compute estimates of which part of the scene is occluded by other parts of the model and cull that portion).

d) Collision detection: allow multiple objects to bounce around the inside of a cube and use the octtree to compute estimated 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 -- up to one week, but instructor must be notified via email in advance -- no penalty (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 15% 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 standard departmental and University policies. Code obtained from the Internet, books, or other sources may *not* be used for any assignment. Exceptions may be allowed but must be given explicit instructor approval.

Teaching Assistant
Denny Wong is the graduate TA (wang124@purdue.edu) . He will hold weekly office hours. Time and location: TBA.