CS535: Interactive Computer Graphics

Instructor: Daniel G. Aliaga (aliaga@cs.purdue.edu, www.cs.purdue.edu/homes/aliaga)
Classroom: LWSN 1106
Time: TTh @ 12pm-1:15pm
Office hours: by appointment (LWSN 3177)
TA: TBD
TA Office hours: TBD

Interested in computer graphics? Does modeling objects interest you? Do you like rendering photorealistic imagery? Is doing animations fun to you? Are you amazing by machine learning and deep visual computing abilities? All this is part of computer graphics. This course teaches the fundamentals, at a graduate school level, for such activities and research projects. Major tools and applications include:

- *Origin: graphics beginning and today*
- *Pipeline: fixed, programmable, and AI-enhanced*
- *Vectors and matrices: ray-tracing, point rendering, and more*
- *Shading and Illumination: diffuse, specular, colors & perception, radiosity, AO, AI based*
- *Images: plenoptic function, lightfields, NERFs*
- *Modeling: procedural modeling, 2D and 3D generative AI-based modeling*

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, a written homework assignment, exams, and interactive class participation. The programming assignments consist of a warm-up assignment, three minor programming assignments, a homework, and a final project. The exams consist of a midterm and a final exam. The final project consists of a fast-forward presentation at mid-project time and a final project presentation at end of semester. 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. Books

There is no mandatory book but here are some recommendations.

- Computer Graphics: Principles and Practice, 3rd Edition by Hughes, van Dam, McGuire, Sklar, Foley, Feiner, and Akeley: this is a good overview of fundamentals of graphics.
- OpenGL Programming Guide ("red book"): you can buy the latest version or an old version free at www.glprogramming.com/red, for example.
- OpenGL Reference Manual ("blue book"): you can buy the latest version or an old version free at www.glprogramming.com/blue or https://www.khronos.org/registry/OpenGL-Refpages/gl4/, for example.

There are plenty of other good books and I will bring some to class so you can peek at them.

4. Grading

Attendance:	5%
Programming Assignments:	30% (assignments 0-4: 1%, 4%, 7%, 10% and 8%)
	35% (5% fast forward, 30% final presentation)
Exams:	15% (midterm)
	15% (final)

	100% TOTAL

5. Tentative Lecture Schedule (most current one is on webpage)

August 20, 22
(1 week) HISTORY: Why does it matter?
The beginning: custom hardware
The revolution: mass produced hardware
Programmability: programmable hardware
Visual computing: AI and more
Toolbox Background

August 27, 29, Sept 3, 5
(2 weeks) GRAPHICS PIPELINE: What's the big picture?
Vector/Matrix Matrix
3D to 2D: Cameras
Fixed pipeline
Programmable pipeline (GPUs)
Deep learning augmented pipeline
End-to-end generative graphics modeling
Hierarchical data structures

September 10, 12, 17
(1.5 weeks) RAY-TRACING AND POINT RENDERING: Lets draw rays and points…
Vector math
Intersections
Points only
Splatting

September 19, 24, 26
(1.5 weeks) POLYGON RENDERING: Drawing polygons because most machines do…
Triangulation
Interpolation of parameters
Curved surface basics

October 1, 3
(1 week) SHADING and ILLUMINATION I: Lets color, shade and more.

Diffuse
Specular
Ambient Occlusion
Colors and perception

October 10, 15
MIDTERM
10th: Review
15th: Midterm (in class)

October 17, 22
(1 weeks) SHADING and ILLUMINATION II: Lets color, shade and more.
Radiosity
Differential rendering/Neural networks

October 24, 29, 31
(1.5 weeks) IMAGE-BASED RENDERING: Lets make pictures…
Plenoptic function
Lightfields
NERFs
Generative models

Nov 5, 7, 14
(1.5 weeks) GENERATIVE MODELING: How to make stuff…
Procedural modeling
Generative Modeling
LLM modeling

Nov 12
Final Project Mid-project Review

Nov 19, 21
(1 week) STYLE-and-APPEARANCE: I like your style…
Non-photorealistic Rendering (NPR)
Sketching
Deep learning style transfer and more

Nov 26, 28 (Thanksgiving)

Dec 3, 5: TBD, Course Review

6. Assignments

You may use CS lab computers or home computers. Assignments must be written in C/C++ on a Windows computer or a Mac computer. However, TA support for Mac-related issues are less supported. Assignments are due before class time on the due date and must be uploaded to Brightspace including all source code, data files, and *an already compiled program*. The 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 – Cook it! (1 week). The objective is to cook up your first compiling, though simple, graphics project. You can either use (i) OpenGL+FreeGLUT or (ii) OpenGL+Qt to implement a simple screen-saver style program. The program will open up a window, display a GUI to choose screen-saver options, and draw a simple 2D screen saver in the main window.

Assignment #1 – Compute It! (1 weeks): The objective is to ensure you understand well camera models and perspective projection. The logic will be described in class and in the assignment. You can use the GLM library to do the linear algebra – or implement your own necessary math libraries.

Assignment #2 – Squash It! (2 weeks). TBD

Assignment #3 – GPU It! (3 weeks). TBD.

Assignment #4 – Write It! (1 week). This objective is a written exam-like homework on some of the relatively more advanced topics.

Final Project (6 weeks). Projects will be presented on a publicly attended "demo day" at the end of the semester (last week of classes, details TBD based on enrollment). You may choose a project that builds upon suggested ones or you may provide a written proposal for an independent project. Team projects (of up to 2 students) are permitted. Grading: the final assignment must be a polished product, with a well designed user interface and clean, reliable functionality.

7. Mid-Project Fast-Forward Presentation

In the middle of the final project time period, each project (individual or group), will give a short GRADED "fast forward" presentation about a background literature search of their proposed project and initial progress. The presentation should include mostly a summary of the state of the art and a preview of what you have already done for your project.

8. 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.

9. Administrative Issues

Late policy
Assignments are due before class on due date. One late pass is given to each student for use in one of assignment 0 to 4. It provides no penalty for up to one week – it will automatically be applied to the FIRST assignment that is given in late. Second and subsequent times -- grade reduction of 33% per day (e.g., turning in 0.001 to 23.999 hours late implies a grade of 100/100 will become 77/100; turning in 24.000 to 47.999 hours late will convert a 100/100 to 34/100; turning in 48.000 or more hours late is a 0/100). Final project fast forward and final project have no late pass.

Collaboration
All assignments, exams, and review presentations must be done individually. Final projects may be done in teams upon approval by the instructor. 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/project. Exceptions allowed only under explicit instructor written approval.