

CS 334 00 – Fundamentals of Computer Graphics, Fall 2019

Christoph M. Hoffmann, 3183 Lawson, cmh@purdue.edu

Lectures MWF 9:30, Haas G066

August 1, 2019

periodic updates thereafter, accounting for class progress

General Points

The course teaches basic tools of computer graphics, material that also finds application in computer games, animation, scientific visualization, etc. There will be both programming assignments as well as paper and pencil homework.

In order to make good on wider relevance, mathematical tools and facts will be considered on par with the program assignments. Too often do we hear from industry that good math skills are needed for the work they do.

Programming assignments must be done using Microsoft Visual Studio 2015 or up, C++, and libraries such as `freeglut.lib`. If you are worried about compatibility issues, it may be a good idea to do the work on the lab PCs. When you do make sure to save your work on the non-volatile file system. You risk losing your work if you neglect this matter.

The role of a textbook for this course is to give you a starting point of some topic you feel unsure about. The two books I would recommend, about the basics, are

1. S. Marschner and P. Shirley, *Fundamentals of Computer Graphics*. CRC Press, 4th edition, 2015.
2. J. Hughes, A. Van Dam, M. McClure, D. Sklar, J. Foley, S. Feiner, K. Akeley; *Computer Graphics, Principles and Practice*. Addison-Wesley, 3rd edition, 2014.

Both books are excellent. You won't go wrong with either one. (2) is the current version of the pioneering "Foley and Van Dam."

There is also a Primer, just for OpenGL. It is quite good.

1. E. Angel, *OpenGL: A Primer*, 3rd Edition, Addison-Wesley, 2007.

Trade Press books are often outdated, so the value of the sample codes they contain is often limited. Note that there are always search engines where you can ask your questions and get answers. Sometime the answers are really good.

Below is a list of the topics I plan to discuss and explain, not necessarily in that order. The learning outcomes are straightforward: understand the topics taught, in depth. The lineup is aggressive: there is a lot of material here ☐☐☐ we may not get to all of it. Then again, perhaps we are all in the zone and wind up with extra time; who knows?

Course Mechanics; August

We begin by discussing the rules of the road: assignments, exams, and what-if scenarios. The course will be challenging, but it should also be fun. We will work to get everyone on the same page.

OpenGL Pipeline; August

If you understand how OpenGL works you will avoid those pesky occasions where the rendered image does not agree with your expectation and your program appears to ignore your code. So we need to spend some time on this topic. There is no greater disappointment than to render, after hours of hard work, a black rectangle.

Basic Viewer, Linear Algebra, Quaternion, 4x4 Transforms; August

Exercising your understanding of the OpenGL pipeline, you will implement a basic rendering program, for files of polygons, and you will read and write such files describing the objects. The program you construct for this will support rendering and interacting with the rendered 3D object(s). Interaction includes rotation and scaling, executed by mouse gestures. This part makes use of linear algebra and exercises projective geometry, 4-by-4 transforms, and quaternions. Those math concepts are fundamental and important to your future career. Nobody does graphics without knowing this technology.

Color and Perception of Color; September

The eye reports *percepts*., the brain integrates it into perceptions. The two are not necessarily the same. Consequently, we need to understand how visual perception works. More than that, different display monitors may have different color spectra, another aspect that you need to be aware of. We will learn about color spaces, RGB vs HSV conversion, the RGB cube, etc. Some applications work with restricted number of color combinations, and for those we will learn about color quantization.

Lighting and Basic Rendering; September

The basic viewer will be enhanced by adding lights and switching between flat and smooth shading. This addition adds strong depth cues. As before, the main interaction is through mouse and menu, but we will later add to it an interface with buttons, sliders, etc., to augment interaction. Phong and Gouraud shading will be discussed, as will reflectance and material properties impacting rendering. The archival format will be extended suitably. A color dialogue will add the ability to change colors.

Perspective, Cameras, Ray Tracing; October

If you only use a function that paints a specific pixel with an RGB color, you could, in principle, draw anything that can be shown on a raster. One way to do it is by ray tracing (and a generous budget of time). It is good to know the basics of ray tracing. Then you can later on experiment with camera models. You could also mix ray tracing and OpenGL. That requires a precise understanding of the rendering pipeline.

Motion; October

When you simulate dynamic systems such as a roller coaster, the motion has to be realistic. Your senses are quite capable of judging motion. A realistic approach uses differential equations, that is, equations describing how quantities in moving systems change. For instance, when an object drops in free fall the acceleration is constant. Pick a simple system, for instance a fountain, and simulate how the water droplets move. CPU speeds are quite fast enough to simulate such systems using the simplest of algorithms to evaluate the equations and so produce a realistic scenario.

Bézier Curves and Surfaces; November

Bézier curves and surfaces are a technology for creating shapes that is widely used in industry. These algorithms are fundamental and were included into GL and OpenGL right from the get-go. Some of the simpler methods include the DeCasteljau algorithm, properties such as convex hull and variation diminishing, degree raising, and building complex curves by fitting pieces together. A few years ago the CS 434 students put together a simulated roller coaster. The tracks were Bézier curves.

Raster Algorithms and Image Processing; December

Short of drawing fancy pictures (POVray style for instance), it might interest you how the lower layers of OpenGL draw lines and circles. More than that, how would we draw a polygon, outline or filled, and what would we need to do when the polygon sticks out beyond the view port? You'd be surprised what comes up when looking into these apparently simple tasks. Matters such as anti-aliasing crop up. Basic image processing needs to know this, whether you want to construct high resolution or instead compress the image.

Homework and Exams

Depending on class progress, there will be 6 programming assignments and 6 written homework assignments. The two categories may be weighted differently, accounting for the time demands of each. The midterm is October 1, the final is December 9. Typical grade compositing is programming assignments 35%, written homework 20%, midterm 20%, final 25%.

The Course Team

Instructor: Christoph Hoffmann; office hours M, F 2:30-3:00 and by appointment.
Teaching Assistant: Chengyuan Lin; office hours T, W 10:30 – 12:00.
Grader: Yizhi Song.