CSCI 4229/5229 Computer Graphics, Fall 2018

Instructor: Willem A. (Vlakkies) Schreüder

Email: willem@prinmath.com

Course Objectives

The course is targeted at students with a wide range of backgrounds in Computer Graphics, ranging from students with no previous experience to students with undergraduate courses in Computer Graphics. The course teaches both fundamental theory of Computer Graphics as well as practical applications using OpenGL.

Lectures cover fundamental techniques of computer graphics such as 2D and 3D viewing, transformations, drawing lines and polygons, clipping and color to advanced techniques such as lighting, shadows, textures and shaders. OpenGL is used to illustrate implementation of these techniques.

Weekly assignments comprise a sequence of increasingly complex OpenGL programs that seeks to build practical experience using OpenGL. The final assignment is a course project which is an OpenGL program of the students choosing.

Writing portable code that will run on any operating system and any machine with adequate hardware is emphasized.

Grading

There is no midterm or final examination. The grade is entirely based on weekly assignments (50%) and the course project (50%).

The assignments for the graduate and undergraduate students are the same, but more is expected from graduate students, especially for the class project.

Requirements

Students should be comfortable with basic linear algebra, data structures and algorithms. A familiarity with the C programming language is assumed and all example programs will be in C. Students should be comfortable programming in a high level language such as C or C++ for which OpenGL bindings are available. It is assumed that students know how to compile and link programs.

Assignments

There is one assignment per week for the first eight weeks. Assignments generally build on previous assignments in that they become more complex, and code reuse from previous assignments will simplify successive assignments.

Assignments may be completed using a computer language and platform of the student's choice, although C or C++ on a Windows, OS/X or *NIX environment is preferred. CSEL is available to complete the

assignments and for testing.

Assignments will be graded on a GNU/Linux system. Programs should contain *#ifdef* statements to facilitate compilation on this system. Students using exotic programming languages will be expected to aid the instructor in setting up a suitable environment for grading the assignments.

Assignments must be submitted via moodle. Assignments are due at 11:59pm on the due date. The grace period for all assignments is until 08:00 am the next day. Late assignments will not be accepted unless previously arranged. Emailed assignments will **never** be accepted.

BBA students are encouraged to submit assignments on the same schedule as on campus students, but special accomodations will be made on an individual basis.

Course Project

The course project involves writing a significant graphics program. It is intended to be approximately one third of the overall course load and is due by the last day of class.

Potential projects would be a scientific visualization application, a game or any application with a heavy graphical emphasis. Students are encouraged to develop an application that is useful in some other aspect of their studies or work. To accommodate this students are given wide latitude in terms of platform and language of implementation. However, in order to facilitate grading, this should be done in coordination with the instructor.

Students are encouraged to do an oral presentation of their project during class near the end of the semester. The oral presentation is optional, but is highly encouraged, especially for graduate students.

Copying

All assignments including the course project are to be completed individually. Exceptions to this rule may be made by prior arrangement if the scope of the project is particularly ambitious.

Each assignment should reflect each student's individual work. However, code reuse is permitted, including example code from the class as well as code from resources on the web. A "safe harbor" in this regard is simply a comment indicating where code from another source is reused.

Students are responsible for whatever code they turn in. Therefore if you reuse code, make sure that you understand what the code does and why it does it. Errors in borrowed code become your responsibility.

Simply turning in borrowed code is not acceptable. It is expected that if you start with somebody else's code, you should improve on it. You will primarily be graded on the improvements you make.

Tentative Course Outline

Week 1
- Introduction
- Hello World in OpenGL

Week 2
- Drawing in 2D

- Clipping
- Week 3
  - Homogeneous Coordinates
  - Drawing in 3D: Visibility
- Week 4
  - Drawing in 3D: Projections
  - Drawing 3D objects
- Week 5
  - Drawing in 3D: Applications
- Week 6
  - Color and Light
- Week 7
  - Textures
- Week 8
  - Applications of lighting and textures
- Week 9
  - Advanced lighting
  - Transparency
- Week 10
  - Display lists
  - Object models and files
- Week 11
  - Overview of Shaders
  - GLSL examples
- Week 12
  - Special effects: Fog
  - Rasterization algorithms
  - Rendering Lines and Anti-Aliasing
  - Rendering Polygons and Tesselation
- Week 13
  - Parametric Curves and Surfaces
  - Parametric Objects
  - Simple Shadows
- Week 14
  - Advanced Shadows
- Week 15
  - Project presentations

Resources

The OpenGL Programming Guide 8/E The Red (Vermillion) Book
- This is a very thorough introduction to OpenGL and is highly recommended.
- The 8th edition covers OpenGL 4.3.
- The 9th edition covers OpenGL 4.5.
- Older versions of this book is available online and is generally sufficient for the course.

OpenGL: A Primer, 3/E by Edward Angel
- An excellent introduction to the fundamentals of OpenGL. Very readable and a great way to get started.

Inexpensive.

Computer Graphics: Principles & Practice (3/E) Hughes, van Dam, McGuire, Sklar, Foley, Feiner & Akeley

Excellent coverage of principles and applications of computer graphics.

OpenGL.org

OpenGL documentation, code and links.

NeHe OpenGL Tutorials

A series of tutorials and articles covering simple to complex OpenGL operations together with implementation for many different platforms.