Instructor: Ichiro Suzuki, EMS 1219, 229-3718, suzuki AT uwm.edu

Office Hours: MW 12:30 PM – 1:30 PM (till 12/12/18), or by appointment

TA: Reihaneh Rostami, rostami AT uwm.edu
Office Hours: TBA

Victor Williamson, vlw AT uwm.edu
Office Hours: TBA

Course Homepage: Login to D2L.

Prerequisite: Junior Standing; Math 232(P); CompSci 251(P). Additionally, interested students should be familiar with basic linear algebra, trigonometry, and 2D and 3D coordinate geometry, mostly that related to lines, planes, circles, and spheres. This material would be from high school or beginning undergraduate math courses. (P) indicates “A course in which a student is required to earn credit prior to being allowed to enroll in a subsequent higher-level course. A prerequisite course may not be taken for credit subsequent to the earning of credit in the higher-level course.” (UWM Schedule of Classes)

Level: U/G.

Textbooks (required):

- Course Pack for CS459, available in D2L.

Recommended Supplementary Reading:

  http://www.opengl.org/documentation/red_book/
  http://www.opengl.org/documentation/blue_book/
Useful Resources:

- SGI reference site for OpenGL: http://www.sgi.com/software/opengl
- More OpenGL information: http://www.opengl.org
- Older editions of The Blue Book and The Red Book:
  http://www.glprogramming.com/blue/
  http://www.glprogramming.com/red/

Description/Outline: We use Visual Studio 2017 C/C++ and an industry-standard graphics package OpenGL under Windows 10 running on virtual PCs (iMacs) in EMS 942. The approach will be top-down — students will right from the start use OpenGL to create 3D scenes. As we proceed to learn and use OpenGL we shall pick up the underlying theory, mostly the mathematics of geometric transformations, viewing, illumination, splines, and certain implementation algorithms.

Thus we will concurrently discuss OpenGL programming and the underlying “theory” of computer graphics. Tentatively, the “theory” part covers the topics from the following chapters and sections of Hearn, Baker and Carithers (HBC). The sections of the Course Pack (CP) in which they are discussed are also shown.

- CP Ch 4:
- HBC Ch 1 A Survey of Computer Graphics
- HBC Ch 2 Computer Graphics Hardware (mainly 2.1 and 2.2, raster-scan displays and systems)
- CP Ch 6:
- HBC 6.1 Line-Drawing Algorithms
- HBC 6.4 Circle-Generating Algorithms
- HBC 6.10 General Scan-Line Polygon-Fill Algorithm
- HBC 6.11 Scan-Line Fill of Convex Polygons
- CP Ch 7:
- HBC 7.1–7.5 Two-Dimensional Geometric Transformations
- HBC 9.1–9.5 Three-Dimensional Geometric Transformations
- HBC 10.8 Three-Dimensional Perspective Projections
- CP Ch 10:
- HBC 8.7 Two-Dimensional Line Clipping
- HBC 10.11 Three-Dimensional Clipping Algorithms (mainly line clipping)
- CP Ch 11:
- HBC 17.1 Light Sources
- HBC 17.2 Surface Lighting Effects
- HBC 17.3 Basic Illumination Models
- HBC 17.11 OpenGL Illumination and Surface-Rendering Functions
- CP Ch 12:
• HBC 5.3 OpenGL Color Functions
• HBC 6.15 Implementation Methods for Antialiasing
• HBC 16.3 Depth-Buffer Method
• HBC 16.14 OpenGL Visibility-Detection Functions
• HBC 17.11 OpenGL Illumination and Surface-Rendering Functions (OpenGL Transparency Functions)
• CP Ch 13:
  • HBC 20.7 OpenGL Menu Functions
• CP Ch 14:
  • HBC 20.6 OpenGL Interactive Input-Device Functions (mainly OpenGL Picking Operation)
• CP Ch 15:
  • HBC 13.4 Quadric Surfaces
  • HBC 13.6 OpenGL Quadric-Surfaces and Cubic-Surface Functions
  • HBC 18.2 Texture Mapping
  • HBC 18.5 OpenGL Texture Functions
• CP Ch 16:
  • HBC 14.1–14.3 Spline Representations and Continuity
  • HBC 14.8 Bezier Spline Curves
  • HBC 14.9 Bezier Surfaces
  • HBC 14.16 OpenGL Approximation-Spline Functions

In addition, you have to read most of the sections of HBC listed below that discuss OpenGL programming, and look up, in the suggested references or on-line documentations, the details of the OpenGL functions not covered in HBC. (You may skip those marked with (*) first time you read.)

• HBC 3.5 Introduction to OpenGL
• HBC 4.2 Specifying a Two-Dimensional World-Coordinate Reference Frame in OpenGL
• HBC 4.3 OpenGL Point Functions
• HBC 4.4 OpenGL Line Functions
• HBC 4.5 OpenGL Curve Functions
• HBC 4.8 OpenGL Polygon Fill-Area Functions
• HBC 4.9 OpenGL Vertex Arrays (*)
• HBC 4.10 OpenGL Pixel-Array Functions (*)
• HBC 4.12 OpenGL Character Functions (*)
• HBC 4.15 OpenGL Display Lists
• HBC 4.16 OpenGL Display-Window Reshape Functions
Additional chapters/sections of the textbook and topics from other sources might also be discussed.

**Grading:**

- Attendance in LAB 5%, Programming assignments (and possibly homework) 50%, Exam I (Mon 10/29/18) 18%, Final Exam (Wed 12/19/18, 10:00 AM – 12:00 Noon) 27%.
• Grade scale: A for a total score (out of 100) of 93 or higher, A- for 90 or higher, B+ for 87 or higher, B for 83 or higher, B- for 80 or higher, C+ for 77 or higher, C for 73 or higher, C- for 70 or higher, D+ for 67 or higher, D for 63 or higher, D- for 60 or higher, and F for below 60. The instructor reserves the right to increase a grade that, in his judgment, would not reflect the student’s mastery of the material. A grade may be decreased for academic dishonesty.

• Unless otherwise stated, all assignments are due before midnight on their due dates, and marked down as follows: 10% for up to 24 hours late, 20% for up to 48 hours late, and 30% for up to 72 hours late. Submissions more than 72 hours late will not be accepted. If you submit your work for an assignment multiple times, then only the last submission will be graded.

• Your programs should be well-designed, use good coding, use explanatory names for functions and variables, and contain enough comments for a competent programmer to be able to understand what your code does and why.

• Important: Your programs must be written in C or C++, and compile and run, without any changes, under Microsoft Visual Studio 2017 and Windows 10 on the virtual PCs (iMacs) in EMS 942.

Undergraduate/Graduate Differentiation: By a Graduate School requirement, graduate students taking this U/G level course must meet a higher level of expectation. Consequently, assignments and exams for graduate students are different in general from those for undergraduate students. Examples of the differences include (i) extra and alternative assignments/problems that require deeper understanding and/or higher problem solving skills, and (ii) extra reading/research.