

## CIS 277 Introduction to Computer Graphics Techniques

<b>Course Number &amp; Title (A.1)</b>	<b>CIS 277 Introduction to Computer Graphics Techniques</b>
<b>Credit Units (A.2)</b>	1 CU (3 hours of lecture per week)
<b>Instructor (A.3)</b>	Norman I. Badler, Professor, badler@seas.upenn.edu, 304 Levine Hall, 215-898-5862
<b>Text(s)/Required Materials (A.4)</b>	Interactive Computer Graphics, by Edward Angel, 5th Edition, Addison-Wesley, 2009. Course notes made available via Blackboard
<b>Catalog Description (A.5a)</b>	The course is intended as an intensive programming course meeting three times a week. Students will create a computational graphics and geometry software library over the course of the semester. Software design, debugging, unit testing, code re-use, and code sharing will be important characteristics of the programming assignments. There will be (nominally) one assignment due each week. The individual assignments will be small enough to ensure completion, but they are designed to grow into a library of reusable C++ code for other graphics projects or purposes. All code will be based on OpenGL and/or Visual Studio, with QT used as the GUI development environment.
<b>Prerequisites (A.5b)</b>	CIS 120 or equivalent experience
<b>Course Satisfies (A.5c)</b>	[ ] Math [ ] Science [ x ] Engineering [ ] Technical Elective [ ] TBS <b>(check only one, UG curric impact only)</b>  <b>Elective</b>
<b>Course Web</b>	<a href="http://www.cis.upenn.edu/~badler/courses/CIS277.htm">http://www.cis.upenn.edu/~badler/courses/CIS277.htm</a>
<b>Course Outcomes (A.6a)</b>	<ul style="list-style-type: none"> <li>• Understand and implement 2D and 3D vector and matrix representations as a geometry library in C++.</li> <li>• Understand and implement a scene graph for graphics object representation and display traversal.</li> <li>• Understand and implement a variety of 2D and 3D computational geometry algorithms.</li> <li>• Understand and implement 3D camera positioning and viewing parameters.</li> <li>• Understand and implement a 3D polygon mesh data structure (the Half-Edge).</li> <li>• Understand surface modeling methods including polygons, subdivision surfaces, Bezier curves and surfaces, fractals, height fields and deformation methods.</li> <li>• Understand and implement quaternions for rotations and camera views in 3D.</li> <li>• Ability to design and successfully implement a relatively large (~5000 line) program in C++ .</li> <li>• Ability to take a textual specification of some desired functionality, implement a program in C++ that will perform the task and generate test inputs and visual graphics outputs to verify the operation of the program.</li> <li>• Ability to work in a three-person team to adapt and integrate software components authored by self or others into a complete graphics 3D modeling system with on-screen inputs and user interfaces.</li> <li>• Ability to use code sharing tools (Subversion) to coordinate a large team project.</li> <li>• Understand how to program in C++ using software libraries and code debugging.</li> <li>• Understand how to program user interfaces in QT and QT Designer.</li> <li>• Understand how to use immediate graphics mode in OpenGL.</li> <li>• Understand how to use modern Vertex Buffer Objects and shaders.</li> <li>• The ability to select and implement algorithms and data structures for graphics models, displays and user interactions that facilitate application development and execution.</li> <li>• Students design and program assignments that build on one another, so that they must be debugged and working before continuing on to the next.</li> <li>• Students design and program software components that must be integrated and extended in specific ways as part of a final team project.</li> <li>• Students wrap all software developed in graphical user interfaces of their own design.</li> </ul>
<b>Contribution towards Program Outcomes (A.6b)</b>	a,b,c,d,i,j,k
<b>Topics Covered (A.7)</b>	<ul style="list-style-type: none"> <li>• Overview of C++ and software library engineering and code debugging.</li> <li>• Introduction to 2D and 3D graphics</li> <li>• 2D and 3D transformations</li> <li>• Matrix data structure for polylines and polygons</li> <li>• Basic OpenGL for 2D drawing</li> <li>• Scene Graph construction and traversal</li> <li>• QT and QT Designer</li> <li>• Viewing transformations</li> <li>• 2D polygon manipulation</li> <li>• 2D geometry algorithms</li> <li>• Introduction to 3D graphics programming</li> <li>• 3D viewing transformations and cameras</li> <li>• Normals, height fields, and meshes</li> <li>• Sub-division surfaces</li> </ul>

	<ul style="list-style-type: none"><li>• Fractal surfaces</li><li>• Bezier curves and surfaces</li><li>• Polygon meshes</li><li>• Vertex Buffer Objects and simple shaders</li><li>• Shape deformation: global transformations, free-form deformations, image morphing</li><li>• Quaternions</li></ul>
<b>Grading Details</b>	70% Homework 10% Midterms 20% Final
<b>Prepared By/Date</b>	Norman I. Badler, May 2011