CSC 470 Introduction to Computer Graphics  
Fall 2008  
Course Syllabus

Instructor: Dr. Natacha Gueorguieva, Professor  
Lectures: Mon 3:35 – 5:30 – 3N 113  
Wed 3:35 – 5:30 – 1N 111  
Office Hours: Mon 2:30 pm – 3:15 pm  Wed 12:30 pm – 2:30 pm or by appointment  
E-mail: natachag@mail.csi.cuny.edu

Web site for the course:  
http://www.cs.csi.cuny.edu/~natacha/TeachFall_2008/CSC470/title.htm

Prerequisite: CSC 326  
Number of credits: 4

Pre-Requisite by Topic:  
1. Good programming skills in C++ language.  
2. Knowledge of data structures (stacks, queues and lists).  
3. Basic knowledge of linear algebra such as matrix and vector space.

Current Catalog Description  
Introduction to the basic concepts and techniques of interactive computer graphics including the hardware and software components of computer graphics systems and mathematical handling of graphical objects. Algorithms for two- and three- dimensional graphics: windowing, clipping and transformations. Viewing with parallel and perspective projections. Possible additional topics include: curves and surface modeling; realistic rendering (shading with illumination and material, shadowing, reflection and surface texturing).

Required texts:  

Course Objectives:  
Students are required:  

a) to learn the fundamentals of computer graphics hardware systems and organization of graphics software systems, to use mathematical transformations and vector techniques in the production of computer graphics, understand the fundamental processes of the 3D graphics rendering pipeline;  
b) to understand the algorithms and fundamental techniques for generating and modifying pictures with a digital computer, including handling of color, and generation of visible-surface projections of three dimensional scenes as well as the advanced features of rendering such as shading and texture mapping;  
c) to learn how to define the underlying geometric and mathematical representations of the objects involved and then translate the representations into algorithms and graphics programs using OpenGL and GLUT to represent 2D and 3D interactive data models.

Learning Outcomes:  
1. Ability to explain the theory, fundamental concepts, and practical concerns involved in representing, modelling, and interacting with graphical scenes in 2D and 3D spaces and define the purpose of each component in the graphics pipeline.
2. Ability to apply mathematical transformations and vector techniques used by linear affine transformations such as scaling, translation, and rotation to points in two- and three-dimensional space and analyze the effects of such transformations.

3. Define and perform orthographic and perspective projections on points and scenes in three-dimensional space.

4. Understand the issues relevant to computer animation and develop interactive applications using 3D graphics.

5. Understand the roles of the eye point, look point, and up vector parameters in the synthetic camera's view of a three-dimensional scene and to perform the computations necessary to illustrate how these parameters affect the model-view transformation matrix.

6. Understand the roles played by color, lighting, and material parameters in shading models (flat, smooth, Gouraud, Phong, ray-tracing, radiosity) and be able to analyze their advantages and disadvantages.

7. Using a graphics library such as OpenGL, implement three-dimensional animations rendered in real-time using an appropriate lighting model and develop experience in graphics programming using a modern API (OpenGL and GLUT).

8. Define and compare the variety of transformations used in texture mapping to identify a point in texture space with a point in world coordinate space.

9. Demonstrate proper documentation of software including internal comments, design reports, and user manuals.

10. An ability to function as part of a multi-disciplinary team.

11. An ability for effective verbal and written communication.

Course Outline
1. Introduction to computer graphics: elements of pictures, graphics display devices, scan conversion.
2. Basic hardware and graphics primitives. Hardware: raster display system and some basic terminologies. Introduction to OpenGL and related libraries (glut, glu, etc.), buffering.
3. Interactive graphics, user input, animation.
4. 2D graphics: graphics primitives, windows, viewports, clipping, 2D viewing in OpenGL, primitive attributes, polygonal fill algorithms, animation. Color theory, RGB color model, basic concepts.
5. 2-D transformations: vector tools for graphics, geometric transformations, homogeneous coordinates, composition of transformation.
6. 3-D transformations: 3D translate, scale, rotate, 3D composition of transformation. OpenGL implementation, examples.
8. 3D models: modeling shapes with polygonal meshes, hidden-surface removal, front and back surfaces, order of visibility, hidden line determination. High realism displays.
9. Illumination and reflectance: shading models, light source models.
10. Rendering: shading models (flat, Gouraud, and Phong), adding shadows.

Methods of Assessment
1. Work on individual homeworks and design projects.
2. Work on team power-point presentations.
3. Mid-term and final exams.
4. Participation in class through discussions and in-class lab exercises.

Assignments
There will be two kinds of assignments: individual assignments (homeworks and two projects) and a group assignment (PPT). All assignments will be posted on the class web page http://www.cs.csi.cuny.edu/~natacha/TeachFall_2008/CSC470/title.htm. Assignments turned in late will be penalized 10% for each day or partial day of lateness for up to five days. After five days, no submission will be accepted unless other arrangements have been made in advance or unless unusual circumstances warrant an exception. All assignments except the PPT must be done individually. While you may discuss the assignment in general terms with others, your solutions should be composed, designed, written and tested by you alone. Topics for PPT are posted on the course web site as well as criteria for PPT evaluation. All PPT will be presented in class in early December. You will get extra points for being creative and going beyond the specifications of your programming assignments.

Exams
There will be one mid-term exam during the semester and one final exam. The final exam will be comprehensive. The mid-term exam consists of theoretical (home-taken) and practical (during the regular class session) parts and will be 80 minutes in length. The final exam will take place during the week specified by the university and will include theoretical and practical part as well. The material for both exams will come from either a material covered in class, homework problems, and/or assignment reading.

Discussions and Lab Exercises
You are encouraged to actively participate in the learning process through in-class activities: discussions and in-class lab exercises. You will receive up to 6 additional bonus points which will be added to your final grade if you regularly and successfully take part in these activities.

Attendance Policy
Attendance will be taken each class. Attendance is not a certain percentage of the grade, but might bring a grade down to WU if you miss more than two weeks of classes. It is the students own responsibility to acquire the material covered in classes she / he missed.

Oral and Written Communications
Each homework assignment should be accompanied by a short report (1-2 pages) and the two projects require 2-3 pages written report as well. The second project requires a written one page user guide Each team member will participate in presentations of up to 20 minutes each.

Academic honesty
You are encouraged to discuss assigned problems with other people but you must individually design and write your own solutions / code for all assignments. It is also acceptable to cut-and-paste code snippets from in-class demos or examples you find on the internet provided that you cite the sources of these code snippets in your report as well as in your program as comments.
Grading

Programming Projects ~ 30 %
Presentation ~ 15 %
Homework assignments ~ 15 %
Midterm ~ 20 %
Final: ~ 20 %

ABET Computer Science Program Outcomes

a) An ability to apply knowledge of computing and mathematics appropriate to the discipline
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
(d) An ability to function effectively on teams to accomplish a common goal
(e) An understanding of professional, ethical, legal, security and social issues and responsibilities
(f) An ability to communicate effectively with a range of audiences
(g) An ability to analyze the local and global impact of computing on individuals, organizations, and society
(h) Recognition of the need for and an ability to engage in continuing professional development
(i) An ability to use current techniques, skills, and tools necessary for computing practice.
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
(k) An ability to apply design and development principles in the construction of software systems of varying complexity.

Assessment of Program Outcomes

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Level</th>
<th>Course Outcomes Met</th>
<th>Method of Proficiency Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Ability to explain the theory, fundamental concepts, and practical concerns involved in representing, modelling, and interacting with graphical scenes in 2D and 3D spaces and define the purpose of each component in the graphics pipeline.</td>
<td>S</td>
<td>a, b</td>
<td>discussions, PPT</td>
</tr>
<tr>
<td>2) Ability to apply mathematical transformations and vector techniques used by linear affine transformations such as scaling, translation, and rotation to points in two- and three-dimensional space and analyze the effects of such transformations.</td>
<td>H</td>
<td>a, b</td>
<td>PPT, homeworks, projects, lab exercises, exams</td>
</tr>
<tr>
<td>3) Define and perform orthographic and perspective projections on points and scenes in three-dimensional space</td>
<td>H</td>
<td>a, b, i</td>
<td>homeworks, projects, exams</td>
</tr>
<tr>
<td>4) Understand the issues relevant to computer animation and develop interactive applications using 3D graphics.</td>
<td>H</td>
<td>b, i</td>
<td>homeworks, project, exams</td>
</tr>
<tr>
<td>5) Understand the roles of the eye point, look point, and up vector parameters in the synthetic camera's view of a three-dimensional scene and to perform the computations necessary to illustrate how these parameters affect the model-view transformation matrix.</td>
<td>S</td>
<td>a, b</td>
<td>discussions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>6) Understand the roles played by color, lighting, and material parameters in shading models (flat, smooth, Gouraud, Phong, ray-tracing, radiosity) and be able to analyze their advantages and disadvantages.</td>
<td>H</td>
<td>a, b, c, i</td>
<td>homeworks, lab exercises, projects, exams</td>
</tr>
<tr>
<td>7) Using a graphics library such as OpenGL, implement 3-dimensional animations rendered in real-time using an appropriate lighting model and develop experience in graphics programming using a modern API (OpenGL and GLUT).</td>
<td>S</td>
<td>a, b, c</td>
<td>project2</td>
</tr>
<tr>
<td>8) 3D models: modeling shapes with polygonal meshes, hidden-surface removal, front and back surfaces, order of visibility, hidden line determination. High realism displays.</td>
<td>S</td>
<td>b, c</td>
<td>discussions</td>
</tr>
<tr>
<td>9) Demonstrate proper documentation of software including internal comments, design reports, and user manuals.</td>
<td>S</td>
<td>c</td>
<td>homeworks, projects</td>
</tr>
<tr>
<td>10) An ability to function as part of a multi-disciplinary team.</td>
<td>S</td>
<td>b, c, i</td>
<td>PPT</td>
</tr>
<tr>
<td>11) An ability for effective verbal and written communication</td>
<td>H</td>
<td>b, c, i</td>
<td>discussions, projects, PPT</td>
</tr>
</tbody>
</table>

**Level:** S = Somewhat supported  H = Highly supported