Global Learning Semesters
Course Syllabus
Course: COMP-620 Computer Graphics and Animation
Department: Computer Science
Host Institution: University of Nicosia, Nicosia, Cyprus

Course Summary		
Course Code	**Course Title**	**Recommended Credit Hours**
COMP-620	Computer Graphics and Animation	4
Semester Offered	**Contact Hours**	**Prerequisites**
Please contact us	42-45	Students are expected to have undergraduate-level knowledge of programming, Data Structures, Algorithms, Calculus and Computer Graphics. Referring to the Computer Science undergraduate-level courses offered at Intercollege, the students are expected to have covered the syllabus of the following: COMP-301 Data Structures or COMP-516 Data Structures and Algorithms, COMP-255 C++ Language Programming or COMP-511 Initial Programming, MATH-190 Calculus and Analytic Geometry, COMP-320 Computer Graphics, COMP-513 Mathematics for Computer Science and COMP-517 Object-Oriented Software Design and Development.
Department	**Level of Course**	**Language of Instruction**
Computer Science	Upper Division	English

Course Description

This course is designed for graduate students pursuing a Master's degree in Computer Science. The aim of this course is to provide a critical study of theory and research related to advanced topic areas of computer graphics. Topic areas include input techniques and devices, display devices, display files, interactive graphic techniques, two- and three-dimensional computer graphics, and transformations. Advanced work in computer graphics, including surface description methods, 3D animation, color perception and images synthesis, dynamic vectors and raster displays. This course also introduces the 3D-raster approach to graphics. Representation schemes for volumes are described with their corresponding rendering and manipulation algorithms, with an emphasis on modeling with volumes

Prerequisites

Students are expected to have undergraduate-level knowledge of programming, Data Structures, Algorithms, Calculus and Computer Graphics. Referring to the Computer Science undergraduate-level courses offered at Intercollege, the students are expected to have covered the syllabus of the following: COMP-301 Data Structures or COMP-516 Data Structures and Algorithms, COMP-255 C++ Language Programming or COMP-511 Initial Programming, MATH-190 Calculus and Analytic Geometry, COMP-320 Computer Graphics, COMP-513 Mathematics for Computer Science and COMP-517 Object-Oriented Software Design and Development.

Topic Areas

1. Introduction to Interactive Graphics
    - display hardware, frame buffer configuration, color look-up tables, aspect ratio
    - 2D line/circle drawing algorithms
    - 2D transformations, homogeneous coordinates, hierarchical data
    - 2D edge clipping, polygon clipping, polygon scan conversion
    - palette animation, sprites, use of overlay planes
    - 3D transformations, edge clipping, perspective projection
    - introduction to visible surface calculation and light source modeling
    - introduction to ray tracing
2. Introduction to 3D Image Generation
    - frame buffers, video technology
    - survey of visible surface algorithms
    - light source modeling: diffuse and specular reflection
    - smooth shading, transparency, shadows
    - basic texture mapping
3. Advanced 3D Image Rendering
    - into to object representations schemes
    - anti-aliasing
    - advanced texture mapping techniques
    - radiosity, advanced ray tracing
    - dithering, rendering systems
4. Geometric Modelling
    - curves and curved surfaces
    - solids of revolution, extrusion, sweep operators
    - Boolean operators, Euler operators
    - lofting techniques,
    - solids modeling: CSG, b-rep, cellular decomposition
5. Computer Animation
    - principles of conventional animation and history of computer animation animation languages
    - key-frame systems, track-based systems shape transformation algorithms
    - morphing interpolation
    - ease-in/ease-out object orientation representations
    - rigid-body animation
    - flexible-body animation
    - character animation
    - particle systems
    - autonomous behavior
    - interactive key-frame systems
6. Geometric Reasoning
    - Survey of 3D object representation methods
    - Skeletonization, collision detection, visibility computation, motion planning, grasp planning
    - conversion algorithms between models, hierarchical data structures
    - collision detection between objects
    - path planning for 3DOF robotic systems
    - representation of space-time
7. Volumetric Graphics
    - survey of volume graphics
    - surface extraction, surface tracking, marching cubes
    - feed-forward viewing
    - continuous and discrete ray-casting
    - template-based methods
    - splatting, hybrid viewing
    - volumetric shading, intro. to 3D discrete topology
    - voxelization algorithms, volume modeling (volume textures, morphing, sculpting, CSG)

- parallel algorithms for volume graph

Course Assessment
Students will be assessed through a series of weekly assignments, semester project and presentation, a midterm and a final exam. The percentages contributing to the final grade are as follows:

Weekly Assignments:	10%
Semester Project and Presentation:	25%
Midterm Exam:	25%
Final Exam :	40%

Description of course assessment:

Weekly assignments: These will consist of non-programming questions/exercises based on the chapter covered during that week.

Project work: These will be 2 or 3 programming projects where the student will put into practice what is been taught in the course.

Some examples:

1) The student will develop a program using a graphical language (e.g. OpenGL) to display wire-frame and shaded images of 3D objects. The program should have the option of having more than one light-sources. The input of the 3D objects will be provided by the lecturer. The program will have a user friendly graphical environment interface to allow the user to apply 3D transformations.

2) The student will develop a program for geometric modeling methods (spline methods) that takes a number of control points defining a curve/surface and generate the curve/surface. The student will explode different modeling methods, such as B-splines, Bezier splines, NURBS, etc, as well as recursive splines such as the Catmull and Clark recursive spline method. The program will be developed in a graphical language (e.g. OpenGL) with a graphical environment interface to allow the user to apply different transformations to the surfaces.

Midterm exam: It will contain the material covered up to the time of the midterm.
Final Examination: This examination will be comprehensive and it will include all the material covered throughout the semester.

Readings and Resources

Required Textbook

- Foley, van Dam, Feiner, Hughes. Computer Graphics Principles and Practice, Second Edition, Addison Wesley, 1996.
- M. Slater, A.Steed, G. Chrisanthou. Computer Graphics and Virtual Environments: From Realism to Real-Time, Addison Wesley, 2001.
- F. S. Hill. Computer Graphics using OpenGL, Second Ed, Prentice Hall, 2000.
- E. Angel, Interactive Computer Graphics: A top-down approach with OpenGL, Addison Wesley, 2001.