COURSE DESCRIPTION

Department and Course Number	CS350	**Course Coordinator**	Eun-Young (Elaine) Kang
Course Title	Foundations of Computer Graphics	**Total Credits**	4

Current Catalog Description:

Programming in an object oriented graphics environment implementing primitive operations in two and three dimensions. Emphasis on image modeling using affine transformations, polygonal meshes and other topics.

Textbook:

F.S. Hill Jr., *Computer Graphics using Open GL, 2nd Ed.,* Prentice Hall, 2001.

References:

- Edward Angel, *Interactive Computer Graphics – A Top-Down Approach Using OpenGL, 3rd Ed.*, Addison Wesley, 2003.
- D. Hearn, M. Baker, *Computer Graphics with OpenGL, 3rd Ed.,* Prentice Hall, 2004
- J. Foley, et al, *Computer Graphics-Principles and Practice, 2nd Ed.*, Addison Wesley, 1996.

Course Goals:

At the end of the course, students understand and can work with

- graphics display devices and graphics primitives.
- Coordinate spaces, coordinate conversion, and transformations of graphics objects.
- The 2D rasterization process and 2D rasterization algorithms.
- The graphics modeling process with polygonal meshes.
- The 3D graphics rendering pipeline.
- OpenGL.

These course goals contribute to the success of **Student Learning Outcomes 1.a, 2, 3, 5, and 6**.

Prerequisites by Topic:

- Good programming skills in Java or C++
- Basic knowledge of data structures
- Basic knowledge of linear algebra such as matrix and vector space

Major Topics Covered in the Course:

- Graphics display devices and input/output primitives
- Linear algebra
- Rasterization
    1. Drawing graphics primitives (lines, polygons, arcs, circles, and ellipse).
    2. Filling polygons
    3. Mapping from the window to the viewport
    4. Intersection of lines
    5. Clipping
- Representation and transformation of geometric objects
    1. Graphics primitives
    2. 2D and 3D transformation groups
    3. Changing coordinate systems
- Modeling shapes with polygonal meshes
    1. Solid modeling with polygonal meshes
    2. Polygonal meshes for a curved surfaces
- Three dimensional viewing and graphics rendering pipeline
    1. Camera position and view
    2. Taxonomy of projections
- OpenGL

Laboratory Projects (specify number of weeks on each):

Through out the quarter, students are required to work on homework or a project.

- Week 1-2: Be familiar with the graphics input/output devices and implement an image input/output program.
- Week 3-4: Implement 2D rasterization algorithms (line drawing and polygon filling.)
- Week 4-7: Implement a 3D wireframe renderer based on the graphics rendering pipeline.

- Week 7-10: Implement an interactive 3D wireframe renderer using OpenGL.

Estimate Curriculum Category Content (Quarter Hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms		0.5	Data Structures		0.5
Software Design		1.0	Prog. Languages		2.0
Comp. Arch.					

Oral and Written Communications:

Written documentation of software built in labs and homework assignments.

Social and Ethical Issues:

No significant component.

Theoretical Content:

Vectors, Matrix Theory.

Problem Analysis:

Students are required (a) to understand the processes of the 3D graphics rendering pipeline in theory, (b) to define the underlying geometric and mathematical representations of the objects involved, and then (c) to translate the representations into algorithms and program codes.

Solution Design:

Solution design adopts a progressive approach. The processes of the rendering pipeline are decomposed into independent steps with a well defined API that connects each to the next. Students complete one step, extend the codes in the next step, and produce the integrated code at the end. Solution design also provides an introduction to OpenGL for all essential data structures (e.g., vector, matrix, and matrix stack). By understanding OpenGL, students can develop efficient programs and can make a smooth transition from renderer programs written in a conventional language to rendererings implemented using OpenGL.