Master Course Syllabus
School of Engineering and Computer Science
Washington State University Vancouver
CS 442
Computer Graphics
3 Semester Hours
(3 lecture hours)

Catalog Description
Raster operations; transformations and viewing; geometric modeling; visibility and shading; color.

Prerequisite Courses
- CS 223 with a C or better
- CS 320 with a C or better
- Math 220 with a C or better

Prerequisite Topics
- Linear systems of equations, matrix operations, linear spaces, vector operations.
- Proficient in the C programming language. Able to develop, build, and debug moderately sized programs.
- Implementation of common data structures for lists, trees, graphs, hash tables, heaps.

Measured Course Outcomes
Students taking this course will:
1. Develop and build an interactive 3D graphics program using the OpenGL application programming interface. (Contributes to performance criteria I-5)
2. Create a polygonal mesh representing the boundary surface of a three dimensional object. (Contributes to performance criteria A-1 and K-2)
3. Use affine transformations to build a hierarchical model composed of multiple interconnected components. (Contributes to performance criteria A-1 and K-2)
4. Describe the graphics pipeline and stream-processing model used in modern GPU architectures. (Contributes to performance criteria A-3)

Required Textbooks
One of the following:
- *Computer Graphics Using Open GL* by Francis S. Hill, Prentice Hall, or
- *Interactive Computer Graphics* by Edward Angel, Addison-Wesley, or
- *Interactive Computer Graphics: A Top-Down Approach with WebGL* by Edward Angel and Dave Shreiner, Pearson.

Reference Material
OpenGL Reference Manual, Addison-Wesley. OpenGL Shading Language, Addison-Wesley. OpenGL Distilled by Dave Schreiner, Addison-Wesley. OpenGL Programming Guide by Woo, et al, Addison-Wesley.

Major Topics Covered in the Course
1. OpenGL graphics pipeline.
2. Event driven programming.
3. Rasterization of lines, polygons.
4. Modeling 3D surfaces with polygonal meshes.
5. Composition of affine modeling transformations to alter the geometry of 3D geometry; Homogeneous coordinates; Hierarchical modeling and scene graphs.
6. Clipping.
7. Shading surfaces using a local illumination model.
8. Viewing transformations; Camera model.
9. Orthographic and perspective projections.
10. Visible surface determination.
11. 2D texture maps.
12. Color theory.
13. Curves and surfaces; parametric and implicit representations.
14. GPU architecture and programming.

Laboratory Projects

Programming projects and/or assignments for this course should include at least the tasks listed in the following table:

Programming Project Area	Weeks
Construct a hierarchical model of an object that consists of interconnected components. Use object instancing and modeling transformations to replicate similar components.	2
Construct a polygonal mesh, augmented with surface normal vectors, representing the boundary surface of a 3D shape. Specify view, projection, lighting and material parameters and render the shape, using the Phong local illumination model, with hidden surfaces removed.	2
Augment a polygonal mesh with texture coordinates and use 2D textures to map an image onto an object's surface.	2

CSAB Category Content

Topic	Fundamental	Advanced
Data Structures	0	1
Computer Organization and Architecture	0	1
Algorithm & Software Design	0	1
Concepts of Programming Languages	0	0

Oral and Written Communications
No significant oral or written communications are specified for this course.

Social and Ethical Issues
No significant social or ethical issues are covered in this course.

Theoretical Content

Topic	Hours
Geometric and topological descriptions of 3D shapes.	3
Homogeneous coordinates and affine transformations.	3
Viewing and projecting 3D shapes.	4
Visible surface determination.	4
Local illumination.	4
Color theory.	1

Problem Analysis
This course explores various computer graphics problems and how they have been solved using a "pipeline" process model as implemented by the OpenGL programming library. Example problems that are analyzed:
- Mathematical representations for shapes;
- Affine modeling and viewing transformations;
- Local illumination and shading of surfaces;
- Orthographic and perspective projections of 3D shapes for 2D displays;
- Hidden surface removal;
- Rendering of geometric primitives (e.g., lines and polygons);
- Mapping 2D image data onto 3D surfaces.
- Color modeling.

Solution Design
The student is presented with a variety of computer graphics problems that are solved with the graphics pipeline model. In the process, the student learns the OpenGL programmer's interface as well as the underlying theory and concepts used in its implementation. The student is required to solve a variety of problems in written assignments and programming projects, for example:
- Construct polygonal meshes representing the boundary of 3D shapes that include surface normals and/or texture coordinates.
- Compose affine modeling transformations to relocate geometry defined in one coordinate system to another;
- Specify camera, lighting and material parameters appropriate for image synthesis.

CC2001
This course provides coverage of topics in the following areas (hours listed are minimums):

Area	Hours
HC6. Graphical user-interface programming [elective]	1
AL10. Geometric algorithms [elective]	2
HC2. Building a simple graphical user interface [core]	1
SE2. Using APIs [core]	1
GV1. Fundamental techniques in graphics [core]	8
GV2. Graphic systems [core]	2
GV3. Graphic communication [elective]	1
GV4. Geometric modeling [elective]	2
GV5. Basic rendering [elective]	6
GV7. Advanced techniques [elective]	5

Course Coordinator:	Scott Wallace
Last Updated:	September 22, 2017
Syllabus Version Number:	2.3