Name of the Course: Computer Graphics

Course Code: CST 471

Year & Semester: IV year, VII Semester

Branch: Computer Science & Engineering

Course Prerequisites:

1. CST203 Data Structures and Algorithms.

2. CST306 Object Oriented Analysis and Design (Programming in C/C++).

3. Linear Algebra.

There will be a significant amount of programming in this class, and students will be expected to use C/C++ for the assignments. If you are not comfortable programming in C/C++, you should not take this class.

In addition, students should be familiar with *basic* linear algebra (vector and matrix representation and arithmetic). A prior course in linear algebra is more than sufficient, and many students may have had enough previous exposure without such a class. A review of most of the relevant material is given in the textbook's Appendices.

Syllabus

Unit- I
Introduction: History of computer graphics, graphics architectures and software.

Vector and Raster graphics techniques, graphics pipeline, Window, viewport and viewing transformation.

OpenGL: architecture, displaying simple two-dimensional geometric objects, positioning systems, working in a windowed environment.

Unit- II
Rasterization: 2D primitive drawing, including scan conversion of line, circle and ellipse and antialiasing.

Discrete Techniques: buffers, bitblt, reading and writing bitmaps and pixelmaps, texture mapping, compositing. Input: Event driven programming, input measure, event, sample and request input, using callbacks, picking.

Unit- III
Polygon filling: flooodfill and scan line polygon filling.

Clipping: Cohen Sutherland outcode line clipping, parametric, midpoint subdivision, and polygon clipping.

Unit- IV
Geometric transformations: affine transformations (translation, rotation, scaling, shear), homogeneous coordinates, concatenation of transformations, current transformation and matrix stacks.

Three dimensional graphics: classical three dimensional viewing, specifying views, affine transformation in 3D, projective transformations.

Specification of view point and complete classification of parallel and perspective projections.

Unit- V
Introduction to hidden surface removal: z buffer, back face removal, Painters algorithm, area subdivision algorithm, binary space partitioning techniques, ray tracing basics.

Color: Color perception, color models (RGB, CMY, HLS), color transformations. Color in OpenGL. RGB and Indexed color.

Shading: Basic illumination and surface modelling including flat, Gouraud, Phong)

Unit- V
Curves: Basics of geometric modelling, parametric and non-parametric curves, continuity constraints, interpolation and approximation. Hermite curves, Bezier curves, B-Spline curves. Introduction to fractals.

Text Books:

1. Hearn and Baker Computer Graphics with OpenGL, 3e, Prentice Hall, 2004.

2. Foley and Van Dam, Fundamentals of Interactive Computer Graphics, 2e, Pearson.

3. Procedural Elements of Computer Graphics, Rogers Adams, Mc-Graw Hill.

4. Mathematical Elements of Computer Graphics, Rogers Adams, Mc-Graw Hill.

Reference Books:

1. Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL, Sixth Edition , Edward Angel, Dave Shreiner, Pearson Education, 2011.

2. Hughes, Van Dam, et al. Computer Graphics Principles and Practice 3e, Pearson, 2014

3. Fundamentals of Computer Graphics, 2e, A K Peters, 2005

4. OpenGL Programming Guide , Addison-Wesley, 2004. OpenGL Reference Manual, Addison-Wesley, 2004.

5. E. Angel, OpenGL: A Primer Addison-Wesley, 2004

Self-Learning Material:

Topic	Reference (Website, Book, Journal, Conference Paper, Magazine etc)
Transformations	**www.mnit.ac.in\nnain**
Projections	**www.mnit.ac.in\nnain**
Hidden surface removal	**www.mnit.ac.in\nnain**

Teaching Methodology / Tools used

This course will be a mixture of lectures, discussions, and demonstrations. The student is expected to actively participate in all class activities. The student is also expected to do onsite practice assignments, take home work on assignments and to complete a set of programming assignments. There will be several (6-8) homework assignments throughout the semester. These may involve written work, programming, or both, and may vary significantly in difficulty.

Course Objectives:

Sr. No.	Objectives
CO1	To impart a basic understanding of the core concepts of computer graphics and to provide a broad exposure to the computer graphics field in order to be prepared for follow-on study.
CO2	Understand a typical graphics pipeline. Usage of open source software tools and development of platform independent robust programs.
CO3	Students will be introduced to algorithms and techniques fundamental to 3D computer graphics and will understand the relationship between the 2D and 3D versions of such algorithms.
CO4	To empower them in translating mathematical formulation of Computer Graphics into visualization. Students will be able to

	reason about and apply these algorithms and techniques in new situations.
CO5	Students be capable of using OpenGL API to create interactive computer graphics.

Course Outcomes:

Sr. No.	**Outcomes**
CO1	Become familiar with basic 2D rendering concepts and algorithms.
CO2	Understand 3D rendering techniques including hierarchical model structures, geometric transformations, projections, and hidden surface removal. Learn free hand curve drawing techniques. Simulate a walk through in a 3D scene with different transformation and projection parameters.
CO3	Understand basic lighting and shading techniques. Understand the basics of color models as they relate to computer graphics.
CO4	Become familiar with basic aspects of geometric and solid modeling. Be able to write basic graphics programs using Open source software APIs.
CO5	To develop the ability to generate and visualize pictures with computer. To further motivate them for real time simulation of the real world using graphics.

Mapping Between Course Outcomes & Programme Outcomes:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	yes		yes	yes			yes					yes
CO2	yes	yes	yes	yes	yes	yes	yes	yes				yes
CO3	yes			yes		yes	yes	yes				
CO4	Yes	yes	yes	yes				yes				

CO5				yes	yes	yes			yes			yes

Lesson Plan and Evaluation:

Lesson No.	Syllabus Content
1.	Graphics primitives, Graphics pipeline, Vector and Raster graphics
2.	Window, Viewport, Viewing transformation.
3.	Scan conversion, DDA and Bresenham line drawing techniques.
4.	Bresenham circle and ellipse drawing techniques.
5.	Clipping, Cohen Sutherland outcode line clipping algorithm with examples.
6.	Midpoint and parametric line clipping techniques and example exercises.
7.	Defining a polygon using polygon table, edge table and vertex table. Sutherland-Hodgman polygon clipping technique and example exercises.
8.	Filling: flood filling (4-connected and 8-connected) with examples.
9.	Scan line polygon filling techniques with practice exercises.
10.	Introduction to 2D Transformations: Translation, Scaling, Rotation
11.	Transformations in 3D, Transformations about an arbitrary axis.
12.	Transformations exercises and take home assignments.
13.	Classification of projections, theory.
14.	Parallel projections theory and mathematical derivations. Orthographic and Axonometric (isometric, diametric, trimetric) projections.
15.	Oblique parallel projections. Deriving cabinet and cavalier projections.
16.	Perspective projections mathematical derivations. Defining a viewpoint for projection.
17.	Projection class exercises, quizes and take home assignments.
18.	Introduction to Hidden surface and line removal. Object space and image space algorithms.
19.	Back face removal derivation and practice exercises.
20.	Z-buffer, scan line and Painters algorithm.
21.	Area subdivision, BSP tree and Ray tracing algorithms.
22.	Practice exercises and class quiz.
23.	Color models and light and illumination. Introduction to CIE chromaticity diagram, RGB and HSL color models.
24.	Ambient and Diffuse light sources and Specular highlights.
25.	Derivation of Phong simple light illumination model.
26.	Defining a polygon, computing a surface normal. Shading models: Flat, Gouraud and Phong.
27.	Computing illumination at a point on an object surface using Phong illumination model for various shading techniques. Example exercises.

Assessment Tools for Attainment of Course Outcomes:

Sr. No.	Assessment Method	Weightage
1	Class perticipation - Question Answering	5%
2	Homework & Assignments	25%
3	Mid Terms Test	30%
4	End Term Examinations	40%