Graphics and Visual Computing

Course Title: Graphics and Visual Computing **Credit: 3**
Course No: CSIT.315 **Number of periods per week: 3+3**
Nature of the Course: Theory + Lab **Total hours: 45+45**
Year: Third, Semester: Fifth
Level: B. Sc. CSIT

1. Course Introduction

This course provides introduction to computer graphics algorithms, software and hardware. Topics include: description of different IO devices used in displaying graphics, algorithms for drawing different output primitives, 2D and 3D transformations, techniques of hidden surface removal, surface rendering methods, and color models.

2. Objectives

Through this course, students shall

- have a knowledge and understanding of the structure of an interactive computer graphics system, and the separation of system components.
- be able to use C and OpenGL for Graphics Programming
- have algorithmic understanding of output primitives and 2D geometrical transformations.
- be able to represent 3D geometrical objects and transform them
- have a knowledge and understanding of techniques of hidden surface removal, surface rendering and color models.

3. Specific Objectives and Contents

Specific Objectives	Contents
• Exemplify application areas of computer graphics • Describe visualization of images and colors in monitors • Explain working of different input devices	**Unit I: Computer Graphics Hardware (5)** 1.1. Introduction, Application Areas of Computer Graphics, Frame Buffer and Display Buffer, Stair Case Effect 1.2. Graphics Devices: Cathode Ray Tube, Raster and Random Scan Displays, CRTs for Color Display, Beam Penetration CRT, The Shadow - Mask CRT, Direct View Storage Tube, 1.3. Input Devices: Keyboards, Mouse, Tablets, The light Pen, Joysticks, Three Dimensional Devices
• Apply C Library functions in drawing graphics • Explain importance of OpenGL in Graphics Programming	**Unit III: Computer Graphics Software (5)** 2.1. C Graphics Basics: Graphics programming, initializing the graphics, C Graphical Functions, Simple Programs using Library Functions.

• Use OpenGL for Graphics programming	2.2. Introduction to OpenGL: Basic OpenGL Syntax, Related Libraries, Header Files, Display-Window Management Using GLUT, A Complete OpenGL Program, Error Handling in OpenGL 2.3. Coordinate Reference Frames, Screen Coordinates Absolute and Relative Coordinate Specifications, Specifying A Two-Dimensional World-Coordinate Reference Frame in OpenGL, OpenGL Point Functions, OpenGL Line Functions, OpenGL Curve Functions
• Explain Line drawing algorithms and Implement them. • Discuss circle and ellipse generating algorithms and implement them. • Demonstrate Filling Algorithms by writing Programs	**Unit III: Output Primitives (6)** 3.1. Line Drawing Algorithms: Line Equation, DDA algorithm, Bresenham's Algorithm, Displaying Polylines 3.2. Circle Drawing Algorithm: Properties of Circle, Mid-point Circle Algorithm 3.3. Ellipse Generating Algorithms: Properties of Ellipse, Mid-point Ellipse Algorithm 3.4. Filing Algorithms: Scan-Line Filling Algorithm, Boundary Filling Algorithm
• Apply transformations such as translation, rotation, scaling, reflection and shear to images. • Use homogeneous coordinate system to represent geometrical transformations • Explain need and process of world to view-port coordinate transformation. • Discuss and exemplify clipping algorithms	**Unit IV: 2D Transformations Clipping & Windowing(8)** 4.1. Transformations: Basic Transformations (Translation. Rotation, Scaling), Other Transformations (Reflection, Shear), Matrix Representations and Homogeneous Coordinates 4.2. Composite Transformations: Translation, Rotation, Scaling General Pivot-point Rotation, General Fixed-point Scaling, Affine Transformation 4.3. 2D Viewing: Viewing Pipeline, Viewing coordinate Refrence Frame, Window to Viewport Coordinate Transformation 4.4. Cliping: Point Cliping, Line Cliping (Cohen-Sutherland Line Cliping and Liang-Barsky Line Cliping), Polygon Cliping (Sutherland-Hodgeman Cliping)
• Able to represent 3D objects using different data structures. • Describe Bezier curves and B-splines used to represent curved surfaces. • Explain 3D transformations and use homogeneous coordinate system to represent it. • Use projection to display 3D objects in 2D display devices.	**Unit V: 3D Concepts & Transformations (8)** 5.1. 3D Object Representations: Polygon Surfaces (polygon Tables, Plane Equations, Polygon Meshes), Bezier Curve and Surfaces, B-Splines. 5.2. 3D Transformations: Basic Transformations (Translation, Scaling, Rotation), Other Transformations (Shear, Reflection), General 3D Rotations, Fixed Point Scaling, Composite Transformations. 5.3. 3D Viewing: Viewing Pipeline, Viewing Coordinates, Transformation from World to Viewing Coordinates, Projections (Parallel Projection, Perspective Projection),
• Understand the concepts behind	**Unit VI: Visible Surface Detection (5)**

visible surface detection and classify the techniques. • Explain image space methods used for visible surface detection. • Describe object space methods and hybrid methods in detecting visible surfaces.	6.1. Classification of Visible-Surface Detection Algorithms: Object Space Methods, Image Space Methods 6.2. Object Space Methods: Blackface Detection, 6.3. Image Space Methods: Depth-Buffer Method, A-Buffer Method, Scan-Line Method, Ray-casting Method 6.4. Hybrid Methods: Depth-Sorting Method, Area Sub-division method, Octree Method
• Discuss different light sources and their applications in surface rendering • Explain illumination models and compare them • Discuss different algorithms used in rendering polygon surfaces	**Unit VII: Surface Rendering Methods (4)** 7.1 Light Sources: Point Source, Distributed Light Source, Diffuse Reflection, Specular Reflection 7.2 Illumination Models: Ambient Light, Diffuse Reflection, Specular Reflection, Phong Specular Reflection, Intensity Attanuation. 7.3 Polygon Rendering Methods: Constant Intensity Shading, Gouraud Shading, Phong Shading, Fast Phong Shading, Ray-Tracing Methods
• Use & explain different models used in generating colors and their applications • Describe conversion between RGB and HSV color model	**Unit VIII: Color Models and Applications (4)** 8.1. Properties of Light, XYZ Color Model and CIE Chromaticity Diagram 8.2. Color Models: RGB Color Model, YIQ Color Model, CMY Color Model, HSV Color Model 8.3. Conversion between HSV and RGB Models, Color Selection and Applications

Evaluation System

Undergraduate Programs							
External Evaluation	**Marks**	**Internal Evaluation**	**Weightage**	**Marks**	**Practical**	**Weightage**	**Mark**
End semester Examination	60	Assignments	20%	20	Practical Report copy	25%	20
(Details are given in the separate table at the end)		Quizzes	10%		Viva	25%	
		Attendance	20%		Practical Exam	50%	
		Internal Exams	50%				
Total External	60	Total Internal	100%	20		100%	20
Full Marks 60+20+20 = 100							

External evaluation

1. **End semester examination:**
   It is a written examination at the end of the semester. The questions will be asked covering all the units of the course. The question model, full marks, time and others will be as per the following grid.

2. **External Practical Evaluation:**
   After completing the end semester theoretical examination, practical examination will be held. External examiner will conduct the practical examination according to the above mentioned evaluation. There will be an internal examiner to assist the external examiner. Three hours time will be given for the practical examination. In this examination Students must demonstrate the knowledge of the subject matter.

   Full Marks: 100, Pass Marks: 45, Time: 3 Hrs

Nature of question	Total questions to be asked	Total questions to be answered	Total marks	Weightage
Group A: multiple choice*	20	20	20×1 = 20	60%
Group B: Short answer type questions	8	6	6×8 = 48	60%
Group C: Long answer type question/long menu driven programs	3	2	2×16 =32	60%
			100	100%

Each student must secure at least 50% marks in internal evaluation in order to appear in the end semester examination. Failed student will not be eligible to appear in the end semester examinations.

Internal evaluation

Assignment: Each student must submit the assignment individually. The stipulated time for submission of the assignment will be seriously taken.

Quizzes: Unannounced and announced quizzes/tests will be taken by the respective subject teachers. Such quizzes/tests will be conducted twice per semester. The students will be evaluated accordingly.

Attendance in class: Students should regularly attend and participate in class discussion. Eighty percent class attendance is mandatory for the students to enable them to appear in the end semester examination. Below 80% attendance in the class will signify NOT QUALIFIED (NQ) to attend the end semester examination.

Presentation: Students will be divided into groups and each group will be provided with a topic for presentation. It will be evaluated individually as well as group-wise. Individual students have to make presentations on the given topics.

Mid-term examination: It is a written examination and the questions will be asked covering all the topics in the session of the course.

Discussion and participation: Students will be evaluated on the basis of their active participation in the classroom discussions.

Instructional Techniques: All topics are discussed with emphasis on real-world application. List of instructional techniques is as follows:

- Lecture and Discussion
- Group work and Individual work
- Assignments
- Presentation by Students
- Quizzes
- Guest Lecture

Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during that period. If a student fails to attend a formal exam/quiz/test, there won't be any provision for re-exam. Unless and until the student clears one semester he/she will not be allowed to study in the following semesters.

Laboratory Work

Student should write programs, prepare lab sheet for each of the topics discussed in classes. Minimum 3 lab hour per week in required. Students can write programs by using C programming language. It is recommended to use widely used graphics library OpenGL in laboratory. Students can also use C-Builder to implement algorithms studied in class. Lab sheet of around 30 programming problems is recommended.

Prescribed Text

- ***Donald Hearn and M. Pauline Baker,*** *Computer Graphics C Vesrion***,** Second Edition, Pearson Education, 2003.
- ***Donald Hearn and M. Pauline Baker,*** *Computer Graphics with OpenGL***,** Fourth Edition, Prentice Hall, 2010.

References

- ***James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes****, Computer Graphics: Principles and Practice*, Third Edition, Addison-Wesley, 2013
- ***Dave Shreiner, Graham Sellers , John M. Kessenich, Bill M. Licea-Kane****, OpenGL Programming Guide: The Official Guide to Learning OpenGL***,** 8th Edition**,** 2013