Lab Manual
Of
Computer Graphics

Academic Year 2012-13
EVEN Semester

Program: B. Tech.
Branch: IT
Subject Code: ETCS258
Credits: 1

Indira Gandhi Institute of Technology
Guru Gobind Singh Indraprastha University
Kashmere Gate, Delhi

Prepared by: Ms. Kokila Gupta

0. GUIDELINES

This document titled ‘Course Planner’ is written based on following guidelines:

1. Course work is divided into 14 weeks. Each week has certain objectives to be accomplished.
2. Each week has 3 lectures supplemented with 2 hours Lab work.
3. Being a course based on programming, it is expected and strongly encouraged that students utilize their Lab hours efficiently.

1. SYLLABUS

As taken from www.ipu.ac.in website*.
**Academics → Academics → Scheme Syllabus (Affiliated Institutes, w.e.f. 2005-06)*

UNIT – I

Transformation, Projections, and Clipping Algorithms: Bresenham's Line Drawing Algorithm, Homogeneous Coordinate System for 2D and 3D, Various 2D, 3D Transformation matrices (Translation, Scaling, Rotation, Shear), Rotation about an arbitrary point (2D), Rotation about an arbitrary axis (3D), Computing location of V.P, Clipping Algorithms, Sutherland-Cohen Clipping Algorithm.
[No. of Hrs. : 11]

UNIT – II

Curves and Surfaces: Bresenham's Circle Drawing Algorithm, Bezier Curves, 4 point and 5 point Bezier curves using Bernstein Polynomials, Conditions for smoothly joining curve segments, Bezier bi-cubic surface patch, B-Spline Curves, Cubic B-Spline curves using uniform knot vectors, Testing for first and second order continuities **[No. of Hrs: 11]**

UNIT – III

Projection and Solid Modelling: Parallel Projection, Oblique Projection on xy plane, Isometric Projection, Perspective Projection, One Vanishing Point (V.P.) projection from a point on z axis, Generation of 2 V.P. Projection, Isometric Projection, Perspective, Projection, one vanishing Pint (VP), projection from 0 point on z axis, Generation of 2 VP Projector & Projections, Solid Modelling. **[No. of Hrs: 11]**

UNIT – IV

Shading and Hidden Surface Removal: Shading, Illumination Model for diffused Reflection, Effect of ambient lighting, distances, Specular Reflection Model, Computing Reflection Vector, Curved Surfaces, Polygonal Approximations, Gourard Shading, Phong Model, Hidden Surface Removal, Back Face Detection, Depth Buffer (Z-Buffer, A-Buffer) Method, Scan Line Method, Depth Sorting Method, Area Subdivision Method.
[No. of Hrs: 11]

TEXT BOOKS:

1. Foley et. al., "Computer Graphics Principles & practice", Addison Wesley, 1999.
2. David F. Rogers, "Procedural Elements for Computer Graphics", McGraw Hill Book Company, 1985.

REFERENCES BOOKS:

1. D. Rogers and J. Adams, “Mathematical Elements for Computer Graphics”, MacGraw-Hill International Edition, 1989.
2. D. Hearn and P. Baker, “Computer Graphics”, Prentice Hall, 1986.
3. R. Plastock and G. Kalley, “Theory and Problems of Computer Graphics”, Schaum’s Series, McGraw Hill, 1986.

2. AIM OF THE LABORATORY

Computer graphics is an art learnt through algorithms. The lab work covers all the basic algorithms for line drawing, circle, ellipse, basics of 2d and 3d transformations, b-spline and bezier curves, clipping, viewing, visibility and rendering. In additions to the above, the student is expected to implement programmes for object and their movement. The ultimate goal of this lab work is to prepare a ground work for next higher level of learning and that is, image processing.

3. HARDWARE AND SOFTWRAE REQUIREMENT

a) Software requirement :

- Turbo C / C++

b) Hardware requirement :

- Intel Pentium III800 MHz Processor
- Intel chipset 810 Motherboard
- 14” colour Monitor
- Mouse
- Keyboard
- 2GB HDD
- 256 MB RAM

4. LAB WORK PLAN

The only way to learn computer graphics is, to write programs in it solving various problems. The more we practice and get our hands dirty, the more we learn. And as we move on to solve increasingly challenging problems, we start enjoying it.

4.1 Lab problems

These are the set of problems that are already put up on the notice boards in the Labs.
Students are required to solve these problems progressively as and when the related conceptual topics are covered in the theory lectures. The respective Lab Instructor (faculty in-charge) can modify, add or delete problems from this set. These problems MUST be covered in each Lab every week and record of the same is to be maintained in a file.

5. GUIDELINES TO STUDENTS:

- Equipment in the lab for the use of student community. Students need to maintain a proper decorum in the computer lab. Students must use the equipment with care. Any damage is caused is punishable.
- Students are required to carry their observation / programs book with completed exercises while entering the lab.
- Students are supposed to occupy the machines allotted to them and are not supposed to talk or make noise in the lab. The allocation is put up on the lab notice board.
- Lab can be used in free time / lunch hours by the students who need to use the systems should take prior permission from the lab in-charge.
- Lab records need to be submitted on or before date of submission.
- Students are not supposed to use pen drives/CD's

6. COMPUTER GRAPHICS LABORATORY

Submissions of laboratory records should include:

a) Aim of the program
b) Program
c) Observed output.

7. LIST OF EXPERIMENTS

Week 1
Study and prepare list of all graphic functions.

Week 2
WAP to display your name using font style, direction and font size.

Week 3
WAP to draw three lines using different functions of graphics library.
WAP to draw ten stylish lines of varying thickness using graphic functions.

Week 4
WAP to draw 2 rectangles and fill it with patterns and color.

Week 5
WAP to draw a house.
WAP to develop a clock

Week 6
WAP to draw a line with DDA algorithm.
WAP to draw a line with Bresenham's algorithm.

Week 7
WAP to draw a circle using Bresenham's algorithm.
WAP to draw a circle using mid point algorithm.

Week 8

WAP to translate a line.

WAP to scale a line.

WAP to rotate a line

WAP to perform all the transformation as per user's choice.

Week 9

WAP to clip the lines using Cohen Sutherland line clipping algorithm.

Week 10

WAP to draw a Bezier curve.

8. RATIONAL BEHIND THE COVERAGE

a) **User Interface** -: Most application that run on personal computer and workstations have user interfaces that rely on desktop window system to manage multiple simultaneous activities , and point and click facilities to allow user to select menu items and other things.

b) **Interactive plotting in business science and technology** -: Computer graphics provide facilities to create 2D and 3D graphs of mathematical, Physical and Economical functions; histograms , bar and pie charts; task scheduling charts; inventory and production charts; and the like.

c) **Office Automation and Electronics Publishing** -: Office automation and electronic publishing can produce both traditional and printed (hardcopy) documents and electronic (softcopy) documents that contain text, tables, graphs and other form of drawn or scanned graphics.

d) **Computer Aided Design** -: In computer Aided Design (CAD), interactive graphics is used to design Components and systems of mechanical, electrical , electromechanical, and electronic devices including structures such as buildings , automobile bodies , aero plane and ship hulls , very large scale integration chips , optical system , and telephone and computer networks.

e) **Simulation and animation for scientific visualization and entertainment** -: Computer produced animated movies and displays of the time- varying behavior of real and simulated objects and become increasingly popular for scientific and engineering visualization.

f) **Art and Commerce -:** Computer graphics is used to produce pictures that express a message and attract attention. Slide production for commercial, Scientific or educational presentation is another coast effective use of graphics.

g) **Process Control -:** whereas flight simulators or arcade games let users interact with a simulation of a real or artificial world, many other application enable people to interact with some aspect of the real world itself.

h) **Cartography -:** Computer graphics is used to produce both accurate and schematic representation of geographical and other natural phenomena from measuring data.

9. BASIC GRAPHICS FUNCTION

1) INITGRAPH

- Initializes the graphics system.

Declaration

- Void far initgraph(int far *graphdriver)

Remarks

- To start the graphic system, you must first call initgraph.
- Initgraph initializes the graphic system by loading a graphics driver from disk (or validating a registered driver) then putting the system into graphics mode.
- Initgraph also resets all graphics settings (color, palette, current position, viewport, etc) to their defaults then resets graph.

2) GETPIXEL, PUTPIXEL

- Getpixel gets the color of a specified pixel.
- Putpixel places a pixel at a specified point.

Decleration

- Unsigned far getpixel(int x, int y)
- Void far putpixel(int x, int y, int color)

Remarks

- Getpixel gets the color of the pixel located at (x,y);
- Putpixel plots a point in the color defined at (x, y).

Return value

- Getpixel returns the color of the given pixel.
- Putpixel does not return.

3) CLOSE GRAPH

- Shuts down the graphic system.

Decleration

- Void far closegraph(void);

Remarks

- Close graph deallocates all memory allocated by the graphic system.
- It then restores the screen to the mode it was in before you called initgraph.

Return value

- None.

4) ARC, CIRCLE, PIESLICE

- arc draws a circular arc.
- Circle draws a circle
- Pieslice draws and fills a circular pieslice

Decleration

- Void far arc(int x, int y, int stangle, int endangle, int radius);
- Void far circle(int x, int y, int radius);
- Void far pieslice(int x, int y, int stangle, int endangle, int radius);

Remarks

- Arc draws a circular arc in the current drawing color
- Circle draws a circle in the current drawing color
- Pieslice draws a pieslice in the current drawing color, then fills it using the current fill pattern and fill color.

5) ELLIPSE, FILLELIPSE, SECTOR

- Ellipse draws an elliptical arc.
- Fillellipse draws and fills ellipse.
- Sector draws and fills an elliptical pie slice.

Decleration

- Void far ellipse(int x, int y, int stangle, int endangle, int xradius, int yradius)
- Void far fillellipse(int x, int y, int xradius, int yradius)
- Void farsectoe(int x, int y, int stangle, int endangle, int xradius, int yradius)

Remarks

- Ellipse draws an elliptical arc in the current drawing color.

- Fillellipse draws an elliptical arc in the current drawing color and than fills it with fill color and fill pattern.
- Sector draws an elliptical pie slice in the current drawing color and than fills it using the pattern and color defined by setfillstyle or setfillpattern.

6) FLOODFILL

- Flood-fills a bounded region.

Decleration

- Void far floodfill(int x, int y, int border)

Remarks

- Floodfills an enclosed area on bitmap device.
- The area bounded by the color border is flooded with the current fill pattern and fill color.
- (x,y) is a “seed point”

¾ If the seed is within an enclosed area, the inside will be filled.

¾ If the seed is outside the enclosed area, the exterior will be filled.

- Use fillpoly instead of floodfill wherever possible so you can maintain code compatibility with future versions.
- Floodfill doesnot work with the IBM-8514 driver.

Return value

- If an error occurs while flooding a region, graph result returns ‘1’.

7) GETCOLOR, SETCOLOR

- Getcolor returns the current drawing color.
- Setcolor returns the current drawing color.

Decleration

- Int far getcolor(void);
- Void far setcolor(int color)

Remarks

- Getcolor returns the current drawing color.
- Setcolor sets the current drawing color to color, which can range from 0 to getmaxcolor.

- To set a drawing color with setcolor , you can pass either the color number or the equivalent color name.

8) LINE,LINEREL,LINETO

- Line draws a line between two specified pints.
- Onerel draws a line relative distance from current position(CP).
- Linrto draws a line from the current position (CP) to(x,y).

Decleration

- Void far lineto(int x, int y)

Remarks

- Line draws a line from (x1, y1) to (x2, y2) using the current color, line style and thickness.

  It does not update the current position (CP).
- Linerel draws a line from the CP to a point that is relative distance (dx, dy) from the CP, then advances the CP by (dx, dy).
- Lineto draws a line from the CP to (x, y), then moves the CP to (x,y).

Return value

- None

9) RECTANGLE

- Draws a rectangle in graphics mode.

Decleration

- Void far rectangle(int left, int top, int right, int bottom)

Remarks

- It draws a rectangle in the current line style, thickness and drawing color.
- (left, top) is the upper left corner of the rectangle, and (right, bottom) is its lower right corner.

Return value

- None.