MIT/CSE/LM/13/R0

COMPUTER GRAPHICS LAB MANUAL

FIFTH SEMESTER

Department of Computer Science & Engineering

10pt. CREDIT SYSTEM

(2014)

Prepared by

(Dr. P. C. Siddalingaswamy)

Approved by

(Head of the Department)

Manipal Institute of Technology
Manipal University
Manipal - 576104

Table of Contents

INSTRUCTIONS TO STUDENTS

1. Students should be regular and come prepared for the lab.
2. In case a student misses a lab session, it is his/her responsibility to complete that missed experiment(s).
3. Students should bring the observation book, lab journal and lab manual. Prescribed textbook and class notes can be kept ready for reference if required.
4. They should implement the given experiment individually.
5. While conducting the experiments students should see that their programs meet the following criteria
    - Programs should be interactive with appropriate prompt messages. Error messages if any and descriptive messages for outputs.
    - Programs should perform input validation (Data type, range error, etc.) and give appropriate error messages and suggest corrective actions.
    - Comments should be used to give the statement of the problem and every function should indicate the purpose of the function, inputs and outputs.
    - Statements within the programs should be properly indented.
    - Use meaningful names for variables and functions.
    - Make use of constants and type definitions wherever needed.
6. Once the experiment(s) get executed, they should show the program and results to the instructions and copy the same in their observation book.
7. Questions for lab tests and exam need not necessarily be limited to the questions in the manual, but could involve some variations and/or combination of the experiments

PROCEDURE OF EVALUATION

Students will be evaluated based on following criteria. The marks distribution is as follows:

Implementation of experiments, observation and/or Journal and viva voce	60 Marks
Lab Test	40 Marks

1. Introduction to OpenGL Programming

Write a Hello world application to understand working with IDE.
Write a program to initialize OpenGL and display some basic geometric shapes. Example, line, polygon.
Implement Basic incremental line drawing algorithm.

2. Drawing of Lines

Write a function for drawing a line using

- Simple DDA algorithm
- Bresenham's Line Drawing algorithm
- Mid-point algorithm (for any slope)

Modify the function suitably for adding attributes like color, thickness and nature (dashed, dotted, etc.).
Write a sample program demonstrating the use of lines.

3. Drawing of Circles and Ellipses

Write a function for drawing a circles using mid-point circle drawing algorithm.
Write a function for drawing an ellipse using mid-point ellipse drawing algorithm.
Modify the function suitably for adding attributes and generating arcs.
Write a sample program demonstrating the use of circles and ellipses

4. Area Filling

Write a function for filling a given area using

- Flood fill algorithm
- Boundary fill algorithm

Write a sample program to demonstrate the use of filling algorithms.

5. Polygon Filling

Write a function for filling a given polygon using the scan line polygon fill algorithm.
Write a sample program demonstrating the use of the polygon filling algorithm.

6. Clipping of Lines

Write a function for clipping a given line against the given rectangular window using

- Cohen-Sutherland method.
- Liang Barsky method

Write a sample program demonstrating the use of line-clipping algorithms.

7. Clipping of Polygons

Write a function to clip a given polygon against the given rectangular window using Sutherland-Hodgeman algorithm.

Write a sample program demonstrating the use of polygon clipping algorithm.

8. 2-D Transformations

Write functions for the following 2-D transformations on 2-D graphic objects

- Translation
- Scaling
- Rotation
- Shear
- Reflection

Write a sample program to demonstrate the use of the various 2-D transformations.

9. 3-D Transformations and Projections

Write a function for the following 3-D transformation of a 3-D object

- Translation
- Rotation

Write a function to display a 3-d object using

- Orthographic Projection
- Perspective Projection

Write a sample program to demonstrate the use of the 3D transformations and projections.

10. Curve Drawing

Write a function to generate a

- Bezier Curve through N+1 control points
- B-spline Curve through N+1 control points

Write a sample program to demonstrate the use of Bezier and B-spline curves.

11. Shading & Lighting

Write program to demonstrate the effects of shading and lighting on a polygon.

12. Animation

Write a program to render a scene and animate objects defined in the scene.

13. Lab Test

Experiments 01 – 12

Guidelines

1. All the experiments have been designed as implementation of graphic primitives. The sample programs are intended to test the application of the primitives. The student is expected to think of an application area and show the use of the primitives in the application areas. The typical applications of graphics are as follows
    - Graph Drawing
    - Engineering Simulations
    - Engineering Drawing
    - CAD
    - Robotics
    - Computer Games
    - Computer Animation
    - Math Tutor
2. For each of the primitive the student should define a suitable data-type and interface. It should be compiled into library and accessible for other experiments.
3. Also as an additional exercise the student should compare the primitives with primitives available in the commercial packages like OpenGL.
4. For the multimedia experiments the student is encouraged to use different tools and compare the advantage and disadvantages of the various tools.
5. Student is encouraged to provide lot of interactivity to their programs.

References

1. Donald Hearn, Pauline Baker M., (2009) "Computer Graphics with OpenGL", Pearson Education, 3rd Edition.

2. Edward Angel, (2009) "Interactive Computer Graphics- A top down approach using OpenGL", Pearson Education, 5th Edition.

3. Dave Shreiner, The Khronos OpenGL ARB Working Group (Author), Bill Licea-Kane, Graham Sellers, "OpenGL Programming Guide: The Official Guide to Learning OpenGL", Version 4.1 (2011), Addison-Wesley Professional, 8th Edition.