

MMTE-004(Computer Graphics)

(Note: All the programs are to be written in C using OpenGL library.)

Sessions 1 and 2

1. Understanding the basic structure of a simple graphics program using OpenGL API.

Write a C program to create a window of specified size and position and draw the following objects with dimensions of your choice, to fit within the window.

(a) A point (b) A rectangle (c) A line segment

(d) A triangle (e) A filled rectangle.

Use OpenGL to generate these objects.

2. Write a C program which takes points as input from mouse clicks (left button) and then performs an action. Apply your program to generate a closed polygon as follows: Every time a left button is pressed, a point on the screen is selected at mouse position and corresponding edge is drawn. When the right button is clicked, the polygon is created. Use one of the line drawing algorithms to plot your line segment.
3. Write a simple C-code to generate a circular arc between two angle values. Use this to draw Fig. 11.

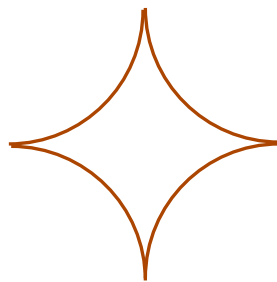


Fig. 11

4. Write a C-code which generates a font interactively. This means after every n mouse clicks, a Bezier curve is generated and then the terminal point of the last drawn Bezier curve is taken as the initial point of the next Bezier curve and so on, until the user right clicks the mouse. In this case the curve of desired degree is drawn and the multi piece font boundary is generated. If the font generation required further drawing, the program

allows starting again on the same window with previously drawn multi-piece curve. But this time the new curve will start with the fresh selected new point.

Session 3

1. Implement the Scan line polygon fill algorithm for any arbitrary polygon in C-language and then use your code to fill each of the following type of polygon.
 - i) Convex polygon
 - ii) Concave polygon
 - iii) Self intersecting polygon.

Session 4

1. Implement the boundary fill algorithm and flood fill algorithm in C-language and use your code to fill two different types of closed areas such as
 - i) A Circle
 - ii) A self intersecting polygon

Compare the results of two algorithms for the self intersecting polygon.

2. Write a code using C-language to generate an arbitrary character by using (i) bitmap method (ii) outline method. Use this code to generate three different characters of your own mother tongue.
3. Write a code in C-language for character generation using the standard OpenGL functions (both bitmap and outline style).

Session 5

1. Write a C-code for an interactive program which allows a user to draw a polygon object in a window and then gives various choices of geometric transformations on the polygon. Once the user selects a choice, the transformed polygon is also plotted on the window. To begin with, you can use a triangle in place of a polygon.

OR

2. Write a C-code that plots an object on the window and on the user's click of mouse on the window, the object starts rotating continuously until the user presses the mouse again.
3. Use your C-code of character generation to draw the same character with regular font type and italic font type by making use of shear transformation.

Session 6

1. Implement Cohen Sutherland and Liang Barsky line clipping algorithms in C-language. Test your code for line segments with end points falling in various regions.

Session 7

1. Write a code to generate a composite matrix for general 3D rotation matrix. Test your code and rotate continuously a cube about an axis.
2. Write a code for (i) general parallel projection where the view vector is at the choice of user and projection plane is x-y plane (ii) perspective projection where the projection reference point is at the user's choice and the projection plane is again xy-plane.