

(ISO 9001:2015 Certified), Accredited with ‘A’ Grade by NAAC

: 08258 - 281039 – 281263, Fax: 08258 – 281265

### Department of Computer Science and Engineering

B.E.CSE Program Accredited by NBA,New Delhi from 1-7-2018 to 30-6-2021

Report on Mini Project

“Simulation of Scanline Polygon filling Algorithm”

**Course Code: 18CS607**

**Course Name: Computer Graphics Lab**

Semester: VI Section :C

**Submitted To:**

**Mr.Puneeth R P**

**Asst Prof Gd II,Dept of CSE,NMAMIT**

**Submitted By:**

Name: S Dhruva USN:4nm18cs140

Name:Samit D Manvar USN:4nm18cs152

**Date of submission:28-05-2021**

**Signature of Course Instructor**

# ABSTRACT

This project is aimed at developing an OpenGL application that represents the Global edge table and Active Edge table for scan-line algorithm used for filling a given polygon .

Our mini project covers all the necessary details starting from introduction about our mini project , the theory related to Global Edge table and Active Edge table by taking a static example . A menu interface is used to switch between different screen such as Introduction page , Home screen, Bucket Edge screen etc. We have explained the poygon filling algorithm by taking a triangle polygon . The Bucket Edge(Global Edge) and the Active Edge table is created with help of keyboard and mouse interactions. Finally with help of active edge table the scanlines of the given polygon is plotted .

**INTRODUCTION**

Computer Graphics is one of the most widely used powerful and interesting feature of computer programming which gives us the power to handle the graphical data very efficiently and also process them rapidly and effectively.The user controls the contents,structure and appearances of the object and of their displayed images by using input devices such as keyboard,mouse or touchscreen.Computer Graphics is used in animation and to simulate real world scenes

In this project ,a particular graphic software system called OpenGL is used .OpenGL is a [cross-language](https://en.wikipedia.org/wiki/Language-independent_specification), [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) (API) for rendering [2D](https://en.wikipedia.org/wiki/2D_computer_graphics) and [3D](https://en.wikipedia.org/wiki/3D_computer_graphics) [vector graphics](https://en.wikipedia.org/wiki/Vector_graphics). The application are designed to access OpenGL directly through functions from three libraries. The first is the GL library , the second is the OpenGL Utility library which uses only Glfunctions thus simplifying viewing .The third library is OpenGL Utility Toolkit.

With help of OpenGL , this project will demonstrate the working of scanline polygon filling algorithm using bucket edge table(Global edge table) and active edge table.

Scanline filling is basically filling up of polygons using horizontal lines or scanlines.

The purpose of the SLPF algorithm is to fill (color) the interior pixels of a polygon given only the vertices of the figure.This mini project will demonstrate by taking a simple polygon (triangle) .

Bucket Edge Table(Global Edge Table)

An initial table constructed and maintains information on edges intersecting the scan line based on the smallest y-coordinate values.

Active Edge Table

It maintains the information of the edges that are active for a particular scanline.Edges must be sorted in lower x coordinate values. This is simulated with the help of stack and linked list where the stack represents the y-coordinates and the linked list represent the edges that compose the figure.

# Implementation Details(PsudoCode)

# #include<bits/stdc++.h>

# #include<GL/glut.h> /\* header files\*/

# int window;

# int menu\_id;

# static int itemno=0; /\*global declarations\*/

# void \*font = GLUT\_BITMAP\_HELVETICA\_18; /\*font style\*/

# void \*fonts[] =

# {

# GLUT\_BITMAP\_9\_BY\_15,

# GLUT\_BITMAP\_TIMES\_ROMAN\_10,

# GLUT\_BITMAP\_TIMES\_ROMAN\_24

# };

# void output(int x, int y, char \*str)

# {

# /\*function to render text or characters on the screen where x and y are the coordinates

# And str is the character to be displayed\*/

# glRasterPos2f(x, y);

# }

# void display1()

# {

# /\*coordinates to display the college name, project title,name,usn etc \*/

# }

# void introduction()

# {

# /\* coordinates to display the introduction about the mini project \*/

# }

# void introduction2(){

# /\* introduction about the topic and some theoritical concepts \*/

# }

# Class button{

# int x1,y1,x2,y2; //variables to get the coordinates to draw a button

# public:

# button(int x11,int y11,int x22,int y22,char \*str1)

# { /\* takes the coordiantes of button \*/

# }

# /\* function declaration\*/

# };

# Class stack{

# button s[size];

# int top;

# public:

# stack()

# {

# top=-1;

# }

# int stfull();

# void push(int item);

# int stempty();

# void displaystack();

# };

# stack st,st1;

# };

# class linkedList //linked list class for displaying intersecting edges

# {

# button que[MAX];

# int front;

# public:

# linkedList()

# {

# front=-1;

# }

# void displaylinkedList();

# void insert\_element(int ch);

# };

# linkedList q;

# void button::draw()

# {

# /\* draws a button based on coordinate values \*/

# }

# void button::togglestate(void){

# /\* state of the button\*/

# }

# int stack::stfull(){} // check if stack is full

# void stack::push(int item) //push an item to stack

# void stack::displaystack(){} //display the values

# void linkedList::insert\_element(int ch) {}

# void linkedList::displaylinkedList(){}

# void show(void) //display bucket edge values

# void show1(void) //display bucket edge values

# void display()//display bucket edge table screen

# void mouse(int btn, int state, int x, int y) //mouse interaction \

# void display3(){} //display a polygon for which we find the active edges

# void scanfill(float x1,float y1,float x2,float y2,float x3,float y3,float x4,float y4)

# / \* scanfill() is used to fill the polygon \*/

# void activeybox1() //display y coordinate button for active edge y=1

# void activeybox2() //display y coordinate button for active edge y=2

# void activeybox3() //display y coordinate button for active edge y=3

# void activeybox4() //display y coordinate button for active edge y=4

# void activeybox5() //display y coordinate button for active edge y=5

# void activeybox6() //display y coordinate button for active edge y=6

# /\* function to display active edges for corressponding y values for active edges\*/

# void activebox1()

# void activebox2()

# void activebox3()

# void activebox4()

# void activebox5()

# void activebox6()

# void activebox7()

# void activebox8()

# void activebox9()

# void activebox10()

# void activebox11()

# void activebox12()

# /\* end of activebox() functions\*/

# void scanline(){} // plot the scanline for the given polygon

# void keyboard(unsigned char key,int x,int y) //keyboard interrupt

# void fillMenu(int option)

# {

# /\* switch between the screens \*/

# }

# void createMenu(void) {

# /\* create menu with different options \*/

# }

# int main(int argc, char\* argv[])

# {

# glutInit(&argc,argv);

# glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

# glutInitWindowSize(600,600);

# glutInitWindowPosition(500,100);

# glutCreateWindow("Scan-line polygon filling Algorithm");

# init();

# glutDisplayFunc(display1);

# glutKeyboardFunc(keyboard);

# glutMouseFunc(mouse);

# glutIdleFunc(idle);

# createMenu();

# glutMainLoop();

# }

# Conclusion

# The primary goal of this project is to illustrate concepts learnt in the theorey and to give better knowledge of these concepts.By visually representing the concept of active edge table and bucket edge(global edge table) the viewers get a clear idea of it.The function available in Opengl are henceforth demonstrated.The keyboard and mouse driven interfaces are used to show various features of the project and to change the appearance of the object.This reduces the complexity and improves the case with which any kind of user can run it.The mini project has also helped in understanding the working of computer graphics using opengl and the various concepts ,functions and methodologies required for the development of a graphics package

# REFERENCES

1. <https://www.opengl.org>
2. <https://learnopengl.com/>

**Appendix**

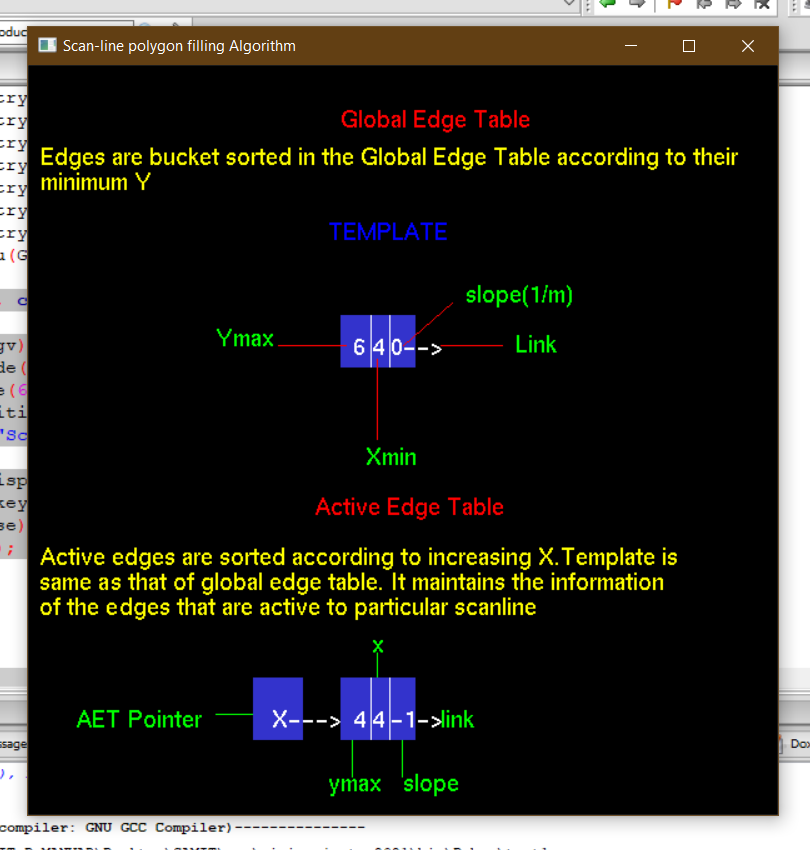
**Screenshots of the project**

# 

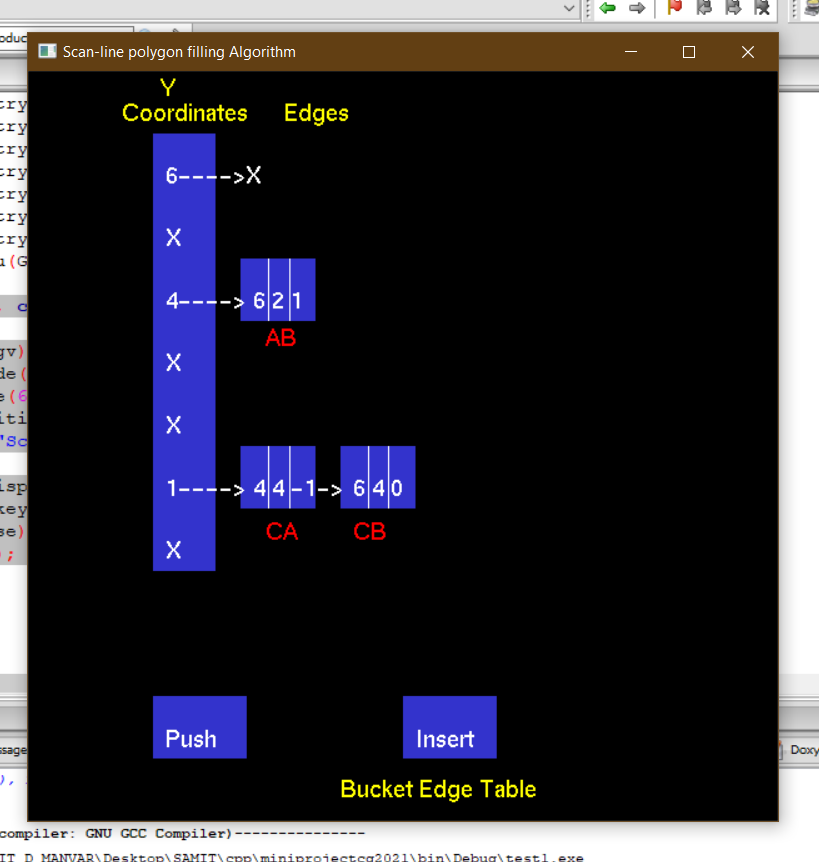
# Home page with menu

# 

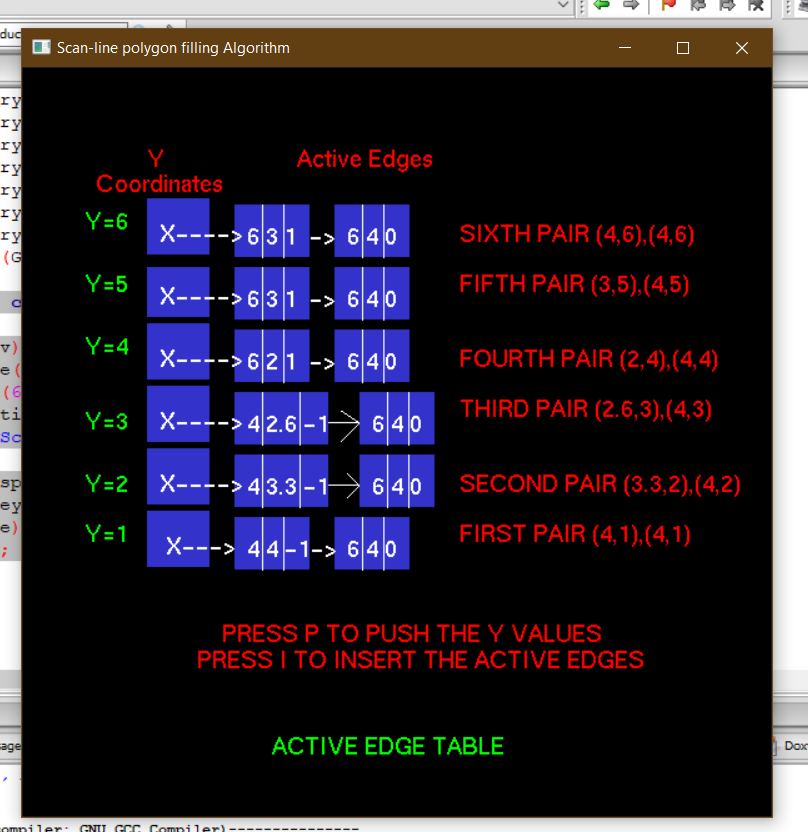
# Introduction page(1)



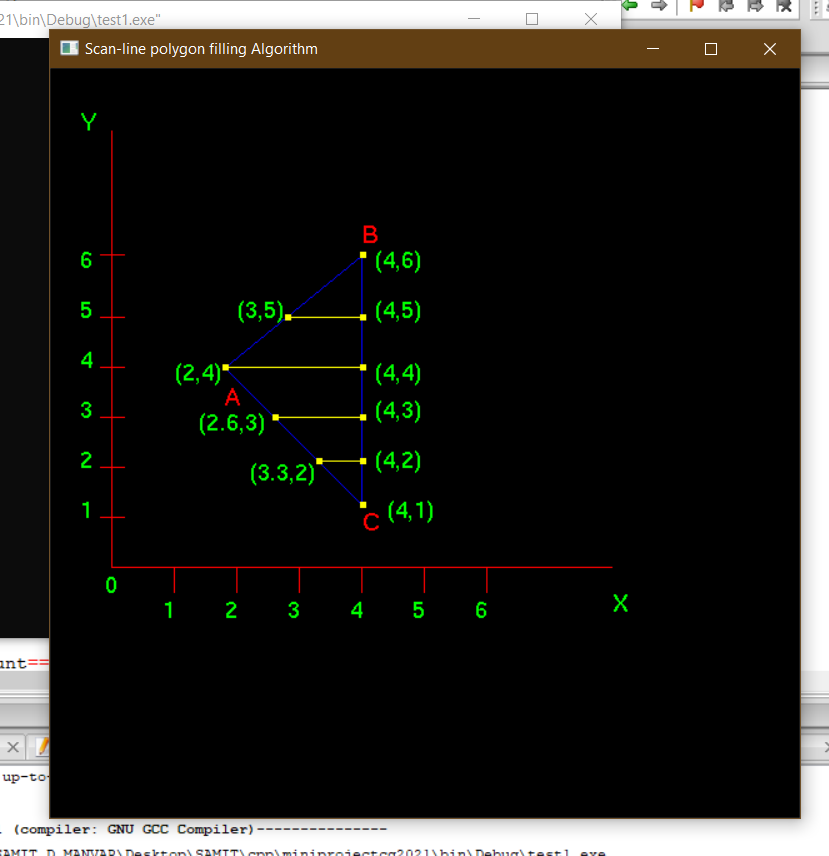
Introduction page(2)



Bucket Edge Table



Active Edge Table



Scanlines for the given polygon