Clip a lines using Cohen-Sutherland algorithm

#include<stdio.h>

#include<GL/glut.h>

double xmin=50,ymin=50,xmax=100,ymax=100;

double xvmin=200,yvmin=200,xvmax=300,yvmax=300;

double lines[10][4]={};

int n=0;

const int TOP=8;

const int BOTTOM=4;

const int RIGHT=2;

const int LEFT=1;

int acceptMain=0;

int ComputeOutCode(double x,double y){

int code=0;

if(y>ymax){

code|=TOP;

}else if(y<ymin){

code |=BOTTOM;

}

if(x>xmax){

code |=RIGHT;

}else if(x<xmin){

code |=LEFT;

}

printf("code= %d x= %lf : y= %lf\n",code,x,y);

return code;

}

void CohenSutherlandLineClipAndDraw(double x0,double y0,double x1,double y1){

int outCode1,outCode0,outCodeOut;

int accept=0,done=0;

outCode0=ComputeOutCode(x0,y0);

outCode1=ComputeOutCode(x1,y1);

//printf("outCodes = %d : %d\n",outCode0,outCode1);

do{

if((outCode0 | outCode1)==0 ){

accept=1;

done=1;

}else if(outCode0 & outCode1){

done=1;

}else{

outCodeOut=outCode0? outCode0 : outCode1;

double x,y;

if(outCodeOut==outCode0){

x=x0;

y=y0;

}else{

x=x1;

y=y1;

}

float slope=(y1-y0)/(x1-x0);

if(outCodeOut & TOP){

x=x0+(1/slope)\*(ymax-y0);

y=ymax;

}else if(outCodeOut & BOTTOM){

x=x0+(1/slope)\*(ymin-y0);

y=ymin;

}else if(outCodeOut & RIGHT){

y=y0+slope\*(xmax-x0);x=xmax;

}else{

y=y0+slope\*(xmin-x0);x=xmin;

}

if(outCodeOut==outCode0){

x0=x;

y0=y;

outCode0=ComputeOutCode(x0,y0);

}else{

x1=x;

y1=y;

outCode1=ComputeOutCode(x1,y1);

}

}

}while(!done);

if(accept){

acceptMain=1;

double sx=(xvmax-xvmin)/(xmax-xmin);

double sy=(yvmax-yvmin)/(ymax-ymin);

double vx0=xvmin+sx\*(x0-xmin);

double vy0=yvmin+sy\*(y0-ymin);

double vx1=xvmin+sx\*(x1-xmin);

double vy1=yvmin+sy\*(y1-ymin);

//printf("main %lf: %lf : %lf : %lf \n",x0,y0,x1,y1);

//printf("%lf: %lf : %lf : %lf \n",vx0,vy0,vx1,vy1);

glColor3f(0.0,0.0,1.0);

glBegin(GL\_LINES);

glVertex2d(vx0,vy0);

glVertex2d(vx1,vy1);

glEnd();

}

}

void display(){

int i=0;

//double x0=60,y0=20,x1=80,y1=120;

glClear(GL\_COLOR\_BUFFER\_BIT);

//CohenSutherlandLineClipAndDraw(x0,y0,x1,y1);

glColor3f(0.0,0.0,1.0);

glBegin(GL\_LINE\_LOOP);

glVertex2f(xmin,ymin);

glVertex2f(xmax,ymin);

glVertex2f(xmax,ymax);

glVertex2f(xmin,ymax);

glEnd();

for(i=0;i<n;i++){

glColor3f(1.0,0.0,0.0);

glBegin(GL\_LINES);

glVertex2d(lines[i][0],lines[i][1]);

glVertex2d(lines[i][2],lines[i][3]);

glEnd();

CohenSutherlandLineClipAndDraw(lines[i][0],lines[i][1],lines[i][2],lines[i][3]);

}

if(acceptMain){

glColor3f(1.0,0.0,0.0);

glBegin(GL\_LINE\_LOOP);

glVertex2f(xvmin,yvmin);

glVertex2f(xvmax,yvmin);

glVertex2f(xvmax,yvmax);

glVertex2f(xvmin,yvmax);

glEnd();

}

glFlush();

}

void myInit(){

glClearColor(1.0,1.0,1.0,1.0);

glColor3f(1.0,0.0,0.0);

gluOrtho2D(0,500,0,500);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

}

void main(int argc,char\*\* argv){

int i=0,j=0;

printf("Enter number of lines=");

scanf("%d",&n);

printf("Enter the 4 coordinates of lines:\n");

for(i=0;i<n;i++){

printf("Line %d:",i+1);

for(j=0;j<4;j++){

scanf("%lf",&lines[i][j]);

}

}

printf("THe lines");

for(i=0;i<n;i++){

printf("%lf %lf %lf %lf\n",lines[i][0],lines[i][1],lines[i][2],lines[i][3]);

}

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500,500);

glutCreateWindow("Line clipping");

myInit();

glutDisplayFunc(display);

glutMainLoop();

}