**ĐỒ HỌA MÁY TÍNH**

BÀI TẬP THỰC HÀNH NHÓM 1

Thành viên nhóm:

* Phan Văn Hà – 17IT3
* Hồ Nhật Huy – 17IT3
* Nguyễn Cảnh Thông – 17CE
* Nguyễn Quang Pháp- 17CE

1. **Thuật toán CohenSutherLand:**
2. Code:

//code

#include<graphics.h>

#include<math.h>

#include<dos.h>

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

void Cohensutherland(int x1,int y1,int x2,int y2,int xa, int ya, int xb,int yb)

{

int a[4],b[4],i,kthuc=0;

while(kthuc==0)

{

a[1]=a[2]=a[3]=a[4]=0;

if(xa<x1) a[1]=1;

if(xa>x2) a[2]=1;

if(ya<y1) a[3]=1;

if(ya>y2) a[4]=1;b[1]=b[2]=b[3]=b[4]=0;

if(xb<x1) b[1]=1;

if(xb>x2) b[2]=1;

if(yb<y1) b[3]=1;

if(yb>y2) b[4]=1;

int ktra=0;

for(i=1;i<=4;i++)

{

if( (a[i]==1) || (b[i]==1) ) ktra=1;

};

if(ktra==0) {

setcolor(10);line(xa,ya,xb,yb);kthuc=1;

}

else

{

int ktra1=1;

for(i=1;i<=4;i++)

{

if( (a[i]==1) && (b[i]==1) ) ktra1=0;

}

if(ktra1==0) {

kthuc=1;

}

else

{

if(a[1]==0 && a[2]==0 && a[3]==0 && a[4]==0)

{

for(i=1;i<=4;i++)

{

int tg1=a[i];a[i]=b[i];b[i]=tg1;

}

int tg2=xa;xa=xb;xb=tg2;

int tg3=ya;ya=yb;yb=tg3;

}

if(a[1]==1)

{

ya=(yb-ya)\*(x1-xa)/(xb-xa)+ya;

xa=x1; //cout<<xa<<" "<<ya<<endl;

}

if(a[2]==1)

{

ya=(yb-ya)\*(x2-xa)/(xb-xa)+ya;

xa=x2;

}

if(a[3]==1)

{

xa=(xb-xa)\*(y1-ya)/(yb-ya)+xa;

ya=y1;

}

if(a[4]==1)

{

xa=(xb-xa)\*(y2-ya)/(yb-ya)+xa;

ya=y2;

}

}

}

}

}

int main(){

int md=0, dr=0;

initgraph(&md,&dr,"c:\\tc\\bgi");

setcolor(11);

rectangle(200,200,400,400);

Cohensutherland(200,200,400,400,150,100,300,300);

getch();

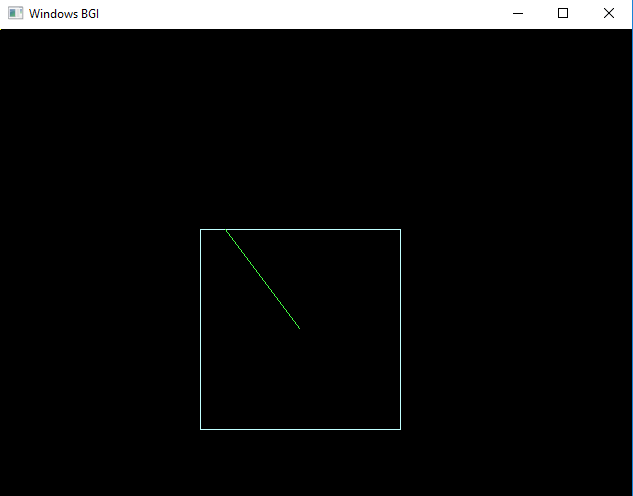
}

//code

1. Dữ liệu thử:

* Tiêu đề: “Thuat toan Line Midpoint”
* Tọa độ điểm: x1 = 150; y1 = 100; x2 = 300; y2 = 300;
* Màu: color = 2

1. Màn hình kết quả:



1. **Thuật toán tô màu theo đường biên (Tô loang):**
2. Code:

//code

#include <conio.h>

#include <winbgim.h>

#include <iostream>

#include <queue>

using namespace std;

struct ToaDo

{

int x,y;

};

int MauNen;

void NhapDaGiac(int &n,int &x,int &y,ToaDo a[])

{

cout<<"Nhap so dinh cua da giac n= "; cin>>n;

for (int i=1;i<=n;i++)

{

cout<<"Toa do dinh P["<<i<<"].x= "; cin>>a[i].x;

cout<<"Toa do dinh P["<<i<<"].y= "; cin>>a[i].y;

}

cout<<"Nhap diem (x,y) thuoc da giac:\n";

cout<<"nhap x="; cin>>x;

cout<<"nhap y="; cin>>y;

}

void VeDaGiac(int n,ToaDo a[],int color)

{

setcolor(color);

for (int i=1;i<=n;i++)

{

int j;

if (i==n) j=1; else j=i+1;

line(a[i].x,a[i].y,a[j].x,a[j].y);

}

}

void ToLoang(int x,int y,int color)

{

// Khai bao queue chua pixel chua duoc to mau

queue<ToaDo> Q;

ToaDo m, Tg;

if (getpixel(x,y)==MauNen && x<getmaxx() && y<getmaxy())

{

m.x = x;

m.y = y;

putpixel(m.x, m.y, color);

Q.push(m); // Them 1 diem vao queue, queue size tang 1

while(Q.empty() == false) //Xet 4 diem xung quanh voi moi diem luu trong queue (neu queue con phan tu)

{

Q.pop();// Xoa 1 diem phia dau queue, queue size giam 1

//Xet cac diem lan can cua 1 diem

if(getpixel(m.x+1, m.y) == MauNen)

{

putpixel(m.x+1, m.y, color );

Tg.x = m.x+1;

Tg.y = m.y;

Q.push(Tg);// Them 1 diem vao cuoi queue

}

if(getpixel(m.x-1, m.y) == MauNen)

{

putpixel(m.x-1, m.y, color);

Tg.x = m.x-1;

Tg.y = m.y;

Q.push(Tg);

}

if(getpixel(m.x, m.y+1) == MauNen)

{

putpixel(m.x, m.y+1, color);

Tg.x = m.x;

Tg.y = m.y+1;

Q.push(Tg);

}

if(getpixel(m.x, m.y-1) == MauNen)

{

putpixel(m.x, m.y-1, color);

Tg.x = m.x;

Tg.y = m.y-1;

Q.push(Tg);

}

m = Q.front();// Dua ve gia tri dau tien cho hang doi

delay(1);

}

}

}

int main()

{

int x,y,n,Gd,Gm=VGAMAX;

ToaDo a[10];

NhapDaGiac(n,x,y,a);

Gd=DETECT;

initgraph(&Gd,&Gm,"");

VeDaGiac(n,a,15);

MauNen=getpixel(x,y);

ToLoang(x,y,10);

getch();

closegraph();

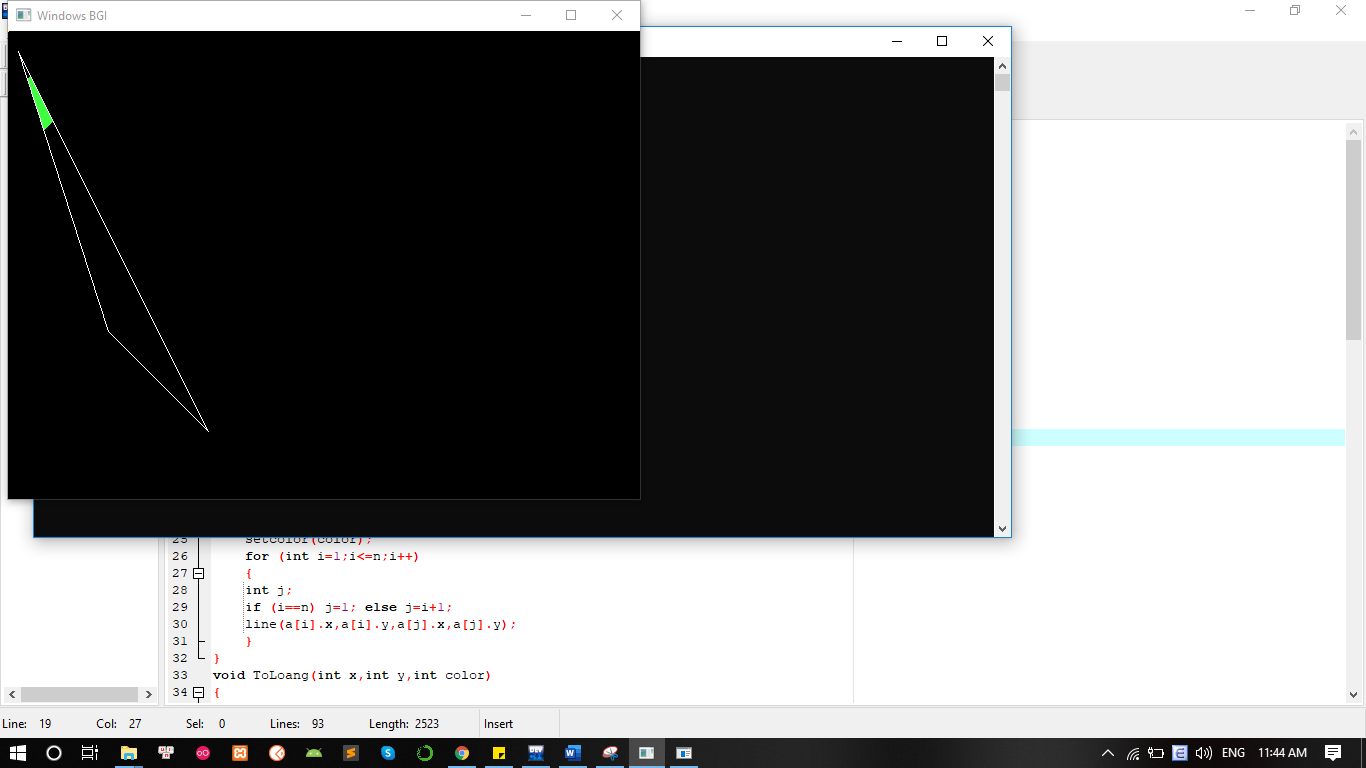
}

//code

1. Dữ liệu thử:

* Tọa độ điểm: x1 = 10; y1 = 10; x2 = 100; y2 = 300; x3 = 200; y3 = 400,x0 = 30, y0 = 70;
* Màu: color = 2

1. Màn hình kết quả:



1. **Thuật toán tô màu theo dòng quét(Scan-line):**
2. Code:

//code

#include <conio.h>

#include <iostream>

#include <graphics.h>

#include <stdlib.h>

using namespace std;

struct ToaDo

{

int x,y;

};

//==============================================

void nhapDaGiac(ToaDo p[], int v)

{

int i;

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

cout<<"\nNhap toa do dinh "<<i+1<<" : ";

cout<<"\n\tx["<<(i+1)<<"] = "; cin>>p[i].x;

cout<<"\n\ty["<<(i+1)<<"] = "; cin>>p[i].y;

}

p[i].x=p[0].x;

p[i].y=p[0].y;

}

//==============================================

void veDaGiac(ToaDo p[], int v)

{

for(int i=0;i<v;i++)

line(p[i].x,p[i].y,p[i+1].x,p[i+1].y);

}

//==============================================

void ScanLine(ToaDo p[], int v)

{

int xmin,xmax,ymin,ymax,c,mang[50];

xmin=xmax=p[0].x;

ymin=ymax=p[0].y;

for(int i=0;i<v;i++){ if(xmin>p[i].x) xmin=p[i].x;

if(xmax<p[i].x) xmax=p[i].x; if(ymin>p[i].y) ymin=p[i].y;

if(ymax<p[i].y) ymax=p[i].y;

}

float y;

y=ymin+0.01;

while(y<=ymax){ //v?i y tang d?n t? ymin > ymax,tìm các giao di?m c?a t?ng y v?i các c?p c?nh

int x,x1,x2,y1,y2,tg;

c=0; //ch? s? c?a m?ng ph?n t?

for(int i=0;i<v;i++){ //xét trên t?t c? các d?nh

//xét 2 d?nh li?n k? nhau

x1=p[i].x;

y1=p[i].y;

x2=p[i+1].x;

y2=p[i+1].y;

if(y2<y1){ //s?p x?p l?i y c?a 2 di?m liên ti?p

tg=x1;x1=x2;x2=tg;

tg=y1;y1=y2;y2=tg;

}

//m?ng giao di?m

if(y<=y2&&y>=y1){

if(y1==y2) x=x1; //n?u y c?a 2 d?nh liên ti?p trùng nhau => b? qua

else{

x=((y-y1)\*(x2-x1))/(y2-y1); //h? s? góc

x+=x1; //300

}

if(x<=xmax && x>=xmin)

mang[c++]=x; //cho ph?n t? c = x sau dó c++

}

}

//v?i t?ng y tang d?n ta v? luôn du?ng th?ng n?i 2 giao di?m

for(int i=0; i<c;i+=2){

delay(30);

line(mang[i],y,mang[i+1],y);

} //line(302,91,300,91

y++;

}

}

//nhap 10 dinh:

//1.(75,250),2.(210,250),3.(250,128),4.(291,250),5.(425,250)

//6.(318,331),7.(360,460),8.(249,380),9.(140,460),10.(182,331)

//3 7 1 5 9

int main()

{

int cl,v;

do{

cout<<"\n Nhap so dinh da giac:"; cin>>v;

}while(v<3);

ToaDo p[v];

nhapDaGiac(p,v);

cout<<"\nChon mau (0-15) : "; cin>>cl;

initwindow(500,600);

veDaGiac(p,v);

setcolor(cl);

ScanLine(p, v);

getch();

}//code

1. Dữ liệu thử:

* Tọa độ điểm: x1 = 10; y1 = 10; x2 = 100; y2 = 300; x3 = 200; y3 = 400 ; x0 = 30, y0 = 70;
* Màu: color = 2

1. Màn hình kết quả:

