q1

Monday, April 18, 2022 12:29 PM

1) Write a program to implement Bresenhams line drawing algorithm

```
#include  <iostream>
#include  <conio.h>
#include  <math.h>



#include <graphics.h>

using namespace std;



void swap(int *&p1 , int *&p2){
int *temp = p2;
p2 = p1;
p1 = temp;

}



double slope(int *p1, int *p2,int &flag){
if(p2[1] - p1[1] ==0){
flag =0;                        // 0 means
return 0;
}
if(p2[0] - p1[0]==0){
flag =1;
return 0;
}

double m = double(double(p2[1] - p1[1] ) / double(p2[0] - p1[0]));
return m;
}



void draw_h(int *p1, int *p2){
int x1 = p1[0];
int x2 = p2[0];
int y= p1[1];
if (p2[0]<p1[0])
{ swap(p1,p2);
}
while(x1 <  x2){
putpixel(x1,getmaxy()-y,255);
x1 +=1;
}
}

void draw_v(int *p1, int *p2){
int y1 = p1[1];
int y2 = p2[1];
int x= p1[0];
if (p2[1]<p1[1])
{ swap(p1,p2);
}
while(y1 <  y2){
putpixel(x,getmaxy()-y1,255);
y1 +=1;
}
```

```
}

void mid_point(int *p1, int *p2){
int flag =-1;
double m = slope(p1,p2,flag);


if(m>1){




double m = slope(p1,p2,flag);

   int dy =  p2[1] - p1[1];
   int dx =  p2[0] - p1[0];


   int a = dy; //    y = dy/dx x   + c     ,, ax + by +c =  0   ,  F(x,y)
= dy x  - dx y  + B dx = 0
   int b = -dx;


cout<<"called steepe positive"<<endl;
   int d = -(2*b+a);    // d is decsion parameter   // 200 - 20 - ->
180

   int S = -2*b;              // -20
   int SE = -2*(a + b);       //  2(180)


   putpixel(p1[0],getmaxy()-p1[1] , 255) ;


   int x = p1[0];
   int y = p1[1];

   while(y <  p2[1]){

      if(d < 0){
         d = d + S;    // if d is negative than choose E ,
         cout<<"negative"<<endl;
      }else{
         d =  d+ SE;
         x +=1;
         cout<<"positive"<<endl;
      }

      y +=1;

      putpixel(x,getmaxy()-y,255);

   }

}


else if(m>0 && m<=1){   // steepe positive



   int dy =  p2[1] - p1[1];
   int dx =  p2[0] - p1[0];
```

```
    int a = dy; //    y = dy/dx x   + c     ,, ax + by +c =  0   ,  F(x,y)
= dy x  - dx y  + B dx = 0
    int b = -dx;



    int d = 2*a + b;    // d is decsion parameter
    int E = 2*a;
    int NE = 2*a + 2*b;



    putpixel(p1[0],p1[1] , 255) ;



    int x = p1[0];
    int y = p1[1];

    while(x <  p2[0]){

       if(d <=0){
          d = d + E;    // if d is negative than choose E ,
       }else{
          d =  d+NE;
          y +=1;
       }

       x +=1;

       putpixel(x,getmaxy()-y,255);

    }


}else if(m < -1){      // steepe negative




    int dy =  p2[1] - p1[1];
    int dx =  p2[0] - p1[0];



    int a = dy; //    y = dy/dx x   + c     ,, ax + by +c =  0   ,  F(x,y)
= dy x  - dx y  + B dx = 0
    int b = -dx;

    int d = (a - 2*b);    // d is decsion parameter
    int S = -2*b;
    int SE = 2*(a -b);

    putpixel(p2[0],getmaxy()- p2[1] , 255) ;
    int x = p2[0];
    int y = p2[1];

    while(y <  p1[1]){   // 0 > 100 , 1>100 , 2>100  ,,,  90 , 89 ,88

       if(d <=0){
          d = d + S ;    // if d is negative than choose E ,
       }else{
          d =  d+ SE;
          x -=1;
       }
       y += 1;
       putpixel(x,getmaxy()-y,255);

    }
```

```
}else{


cout<<"this is gradual negative"<<endl;

if(p2[0]< p1[0]){
   swap(p1,p2);
}

   int dy =  p2[1] - p1[1];
   int dx =  p2[0] - p1[0];


   int a = dy; //    y = dy/dx x   + c     ,, ax + by +c =  0   ,  F(x,y)
= dy x  - dx y  + B dx = 0
   int b = -dx;


   int d = (2*a - b);    // d is decsion parameter
   int E = 2*a;
   int SE = 2*(a -b);

   putpixel(p2[0],getmaxy()- p2[1] , 255) ;
   int x = p2[0];
   int y = p2[1];

   while(x >  p1[0]){   // 200 , 400:::   ,  300 , 350--->   300< 200

      if(d >= 0){
         d = d + E ;    // if d is negative than choose E ,
      }else{
         d =  d+ SE;
         y +=1;
      }

      x -= 1;

      putpixel(x,getmaxy()-y,255);

   }



}







}



void print_point(int *p){
cout<<" x : "<<p[0]<<" y : "<<p[1]<<endl;
}

// -lbgi -lgdi32 -lcomdlg32 -luuid -loleaut32 -lole32

int main()
{
```

```
int *p1 = new int(2);
int *p2 = new int(2);

p1[0] =  200;
p1[1] =  400;

p2[0] = 300;
p2[1] = 350;

//  cout<<" give point 1 :";
//  cin>>p1[0]>>p1[1];

//  cout<<" give point 2 :";
//  cin>>p2[0]>>p2[1];

int gd = DETECT, gm;
char pathtodriver[] = "";
initgraph(&gd, &gm, pathtodriver);


mid_point(p2,p1);


getch();
closegraph();
}
```