dda

Thursday, May 02, 2024 9:34 PM

```
#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 colorr){

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

   int x1 = p1[0];
   int x2 = p2[0];

   int y= p1[1];

   while(x1 <  x2){

      putpixel(x1,getmaxy()-y,colorr);
```

```
        x1 +=1;
    }

}

void draw_v(int *p1, int *p2,int colorr){



        if (p1[1]>p2[1])
    { swap(p1,p2);
    }

    int y1 = p1[1];
    int y2 = p2[1];

    int x= p1[0];



    while(y1 <  y2){

        putpixel(x,getmaxy()-y1,colorr);
        y1 +=1;
    }

}

void dda(int *p1, int *p2,int colorr){


    int flag =-1;

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

    if(flag==0){
        //horizontal


draw_h(p1,p2,colorr);
        return ;
    }

    if(flag==1){
        //vertical

        draw_v(p1,p2,colorr);
        return ;
    }



    if( abs(m) > 1){ // y is larger than x


     if (p2[1]<p1[1])
    { swap(p1,p2);
    }
        int y2 = p2[1];
        int y = p1[1];
        double x = p1[0];


        for(;y<=y2; y++){
```

```
            x  = x + double(1.0/m);
            putpixel(round(x),getmaxy()-y,colorr);
          }
      }else{ // x is larger than y
      if (p1[0]>p2[0])
       {swap(p1,p2);
       }
        int x2 = p2[0];
          int x = p1[0];
          double y = p1[1];


          for(;x<=x2; x++){

             y  = y + double(m);
            putpixel(round(x),getmaxy()-y,colorr);
          }

      }
      //

}




void dda(int x1 , int y1, int x2 , int y2,int colorr ){


    int *p1 = new int(2);
    p1[0] = x1;
    p1[1] = y1;


    int *p2 = new int(2);
    p2[0] = x2;
    p2[1] = y2;




    int flag =-1;

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

    if(flag==0){
       //horizontal


draw_h(p1,p2,colorr);
       return ;
    }

    if(flag==1){
       //vertical

       draw_v(p1,p2,colorr);
       return ;
    }



    if( abs(m) > 1){ // y is larger than x


     if (p2[1]<p1[1])
     { swap(p1,p2);
```

```
    }
      int y2 = p2[1];
      int y = p1[1];
      double x = p1[0];


      for(;y<=y2; y++){


         x  = x + double(1.0/m);
        putpixel(round(x),getmaxy()-y,colorr);
      }
    }else{ // x is larger than y
    if (p1[0]>p2[0])
     {swap(p1,p2);
     }
      int x2 = p2[0];
      int x = p1[0];
      double y = p1[1];


      for(;x<=x2; x++){

         y  = y + double(m);
        putpixel(round(x),getmaxy()-y,colorr);
      }

    }


}




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


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


int main()
{









int a,b;


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


// INPUT 2 POINT
p1[0] = 1;
p1[1] =2;


p2[0] = 100;
p2[1] = 30;
```

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




print_point(p1);
print_point(p2);




dda(p1,p2,RED);



getch();
    closegraph();


}
```