Navlakhi's

CohanSutherland

Methodology and Program

By Abhishek Navlakhi

Semester 4: Computer Graphics

```
/*- Line clipping using cohen sutherland algo -*/
/*-------------------------------------------------------------*/

        #include<iostream.h>
        #include<graphics.h>
        #include <conio.h>


        enum {      TOP=0x1, BOTTOM=0x2, RIGHT=0x4, LEFT=0x8 };
//helps in binary 'OR' and 'AND' ALSO recollect 8,4,2,1 code of digital MSB=8,... LSB=1


 int  calcode (float x,float y,float xwmin,float ywmin,float xwmax,float
ywmax)
 {
   int  code =0;

   if(y> ywmax)
        code |=TOP;
   else if( y<ywmin)
        code |= BOTTOM;
   else if(x > xwmax)
        code |= RIGHT;
   else if ( x< xwmin)
        code |= LEFT;

   return(code);
}

/*-----------------------------------------------*/
```

```
void lineclip(float x0,float y0,float x1,float y1,float xwmin,float ywmin,float
xwmax,float ywmax )
{

        int gd=VGA,gm=VGAHI;
        unsigned int  code0,code1,codeout;
        int accept = 0, done=0;

        code0 = calcode(x0,y0,xwmin,ywmin,xwmax,ywmax);
        code1 = calcode(x1,y1,xwmin,ywmin,xwmax,ywmax);

        do{
            if(!(code0 | code1))          //code0 and code1 both all zero
            {       accept =1 ; done =1; }
            else
             if(code0 & code1)     done = 1;   //code0 & code1 have same bit as 1
             else
             {
               float x,y;
               codeout = code0 ? code0 : code1;
               if(codeout & TOP)
               { //if codeout has 1 in TOP ie.LSB posn (1)
                    x = x0 + (x1-x0)*(ywmax-y0)/(y1-y0);
                    y = ywmax;
               }
               else
                 if( codeout & BOTTOM)
                 {          //if codeout has 1 in BOTTOM ie.2nd from the right (2)
                     x = x0 + (x1-x0)*(ywmin-y0)/(y1-y0);
                     y = ywmin;
                 }
                 else
                    if ( codeout & RIGHT)
                    { //if codeout has 1 in RIGHT ie.2nd bit from the left (4)
                      y = y0+(y1-y0)*(xwmax-x0)/(x1-x0);
                      x = xwmax;
                    }
```

```
                    else
                    {//if codeout has 1 in LEFT ie.1st  bit from the left i.e. MSB (8)
                       y = y0 + (y1-y0)*(xwmin-x0)/(x1-x0);
                       x = xwmin;
                     }
                if( codeout == code0)
                { //if we had chosen code0 for trimming above then recalculate
                  //code for code0
                    x0 = x;  y0 = y;
                    code0=calcode(x0,y0,xwmin,ywmin,xwmax,ywmax);
                }
                else
                { //if we had chosen code1 for trimming above then recalculate
                  //code for code1
                   x1 = x;   y1 = y;
                   code1 = calcode(x1,y1,xwmin,ywmin,xwmax,ywmax);
                }
             }
          } while( done == 0);

          if(accept)  line(x0,y0,x1,y1);

          rectangle(xwmin,ywmin,xwmax,ywmax);

          getch();

}
/*-------------------------------------------------------------------*/
```

```
main( )
{

        float x2,y2,x1,y1,xwmin,ywmin,xwmax,ywmax;
        int gd=VGA,gm=VGAHI;


        cout<<"\n\n\tEnter the co-ordinates of Line :";

        cout<<"\n\n\tX1  Y1  :  ";
        cin>>x1>>y1;

        cout<<"\n\n\tX2  Y2  :  ";
        cin>>x2>>y2;


        cout<<"\n\tEnter the co_ordinates of window  :\n ";
        cout<<"\n\txwmin , ywmin  : ";
        cin>>xwmin>>ywmin;

        cout<<"\n\txwmax , ywmax  : ";
        cin>>xwmax>>ywmax;

        initgraph(&gd,&gm,"D:\\TC\\BGI");

        line(x1,y1,x2,y2);
        rectangle(xwmin,ywmin,xwmax,ywmax);
        getch();
        cleardevice();

        lineclip(x1,y1,x2,y2,xwmin,ywmin,xwmax,ywmax );
        getch();
        closegraph();

}

/****************** COHEN-SUTHERLAND LINE CLIPPING ***************/
```