Assignment-03

Cohen Sutherland line clipping algorithm

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

using namespace std;
class drawpoly
{
public:
int x,y, temp,n,i,j,k,dy,dx;
int a[20][2],xi[20];
float slope[20];

void drawpoly1()
{

cout<<"\n enter the no .of vertices of
polygoan";
cin>>n;
```

```
cout<<"\n enter the coordinates of edges
one by one";
for(i=0;i<n;i++)
{
cout<<"X"<<i<<"Y"<<i<<":";
  cin>>a[i][0]>>a[i][1];

}
a[n][0]=a[0][0];//last point should be
connceted to first point to make a close
fig
a[n][1]=a[0][1];

for(i=0;i<n;i++)
{
line(a[i][0],a[i][1],a[i+1][0],a[i+1][1]);/
/ draw poly
   }
getch();

  }
```

```
}d;
class fillpoly : public drawpoly
{
 public:
  void fillpoly1()
  {

for(i=0;i<n;i++)
{
dy=a[i+1][1]-a[i][1];  //dy=y2-y1
dx=a[i+1][0]-a[i][0];   //dx=x2-x1
if(dy==0)
slope[i]=1.0; // y is always one because
scanline diff is only by -1
if(dx==0)
slope[i]=0.0;
if(dy!=0 && dx!=0)  //calculate inverse
slope if polygon is in 4th quadradent.
{
slope[i]=(float)dx/dy; //formula for
inverse slope
```

```
}
}

for(y=0;y<480;y++) // for reverse
filling(y=480;y>0;y--)
{
k=0;
for(i=0;i<n;i++)
{
if(((a[i][1]<=y) &&
(a[i+1][1]>y))||((a[i][1]>y)&&
(a[i+1][1]<=y) ))// check ymin or ymaxis
less or greater
{
xi[k]=(int)(a[i][0])+slope[i]*(y-a[i][1]);/
/ calculate intersection
k++;
}
}
for(j=0;j<k-1;j++)
{
for(i=0;i<k-1;i++)
```

```
{
if(xi[i]>xi[i+1])
{
temp=xi[i];
xi[i]=xi[i+1];
xi[i+1]=temp;

}
}
setcolor(13);
for(i=0;i<k;i+=2)
{
line(xi[i],y,xi[i+1],y);//draw the line
using intersection points only
delay(20);
}
}
}
}
}f;
int main()
{
```

```
int gd=DETECT,gm;
initgraph(&gd,&gm,NULL);
  f.drawpoly1();
      f.fillpoly1();
closegraph();
return 0;
}
```

```
Input :
X1 , Y1:
100
200
X2, Y2 :
500
100
```

Explanation

This C++ program implements the Scanline Polygon Filling Algorithm, which is used to fill a polygon by determining the intersection points of

horizontal scan lines with the edges of the polygon. The program uses the graphics.h library for graphical output.

Key Components of the Code:

1.Classes:

•drawpoly: This class is responsible for reading the vertices of the polygon from the user and drawing the polygon on the screen.

•fillpoly: This class inherits from drawpoly and is responsible for filling the drawn polygon using the scanline fill algorithm.

2.Polygon Vertex Input:

•The user is prompted to enter the number of vertices and their coordinates. The last vertex is connected back to the first to close the polygon.

3.Line Drawing:

•The drawpoly1() method uses the line() function to draw the edges of the polygon.

4.Filling the Polygon:

•The fillpoly1() method calculates the slopes of each edge and determines where the horizontal scan lines intersect with the edges of the polygon.

•It uses the slopes to compute the intersection points and stores them in an array.

•For each horizontal line (from the bottom to the top of the window), it draws lines between the calculated intersection points to fill the polygon.

5.Graphics Initialization:

•The program initializes the graphics mode and displays the polygon and its filling.