

Assignment 1

Aim : A concave Polygon filling using Scan Fill Algorithm.

Code :

```
#include <conio.h>
#include <iostream>
#include <graphics.h>
#include <stdlib.h>
using namespace std;
class point
{
public:
int x,y;
};
class poly
{
private:
point p[20];
int inter[20],x,y;
int v,xmin,ymin,xmax,ymax;
public:
int c;
void read();
void calcs();
void display();
void ints(float);
void sort(int);
};
void poly::read()
{
int i;
cout<<"\n Scan Fill Algorithm ";
cout<<"\n Enter Number Of Vertices Of Polygon: ";
cin>>v;
if(v>2)
{
for(i=0;i<v; i++) //ACCEPT THE VERTICES
{
cout<<"\nEnter co-ordinate no. "<<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;
xmin=xmax=p[0].x;
ymin=ymax=p[0].y;
}
else
cout<<"\n Enter valid no. of vertices.";
}
void poly::calcs()
{
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;
}
}
void poly::display()
{
int ch1;
char ch='y';
float s,s2;
do
{
cout<<"\n\nMENU:";
cout<<"\n\n\t1 . Scan line Fill ";
cout<<"\n\n\t2 . Exit ";
cout<<"\n\nEnter your choice:";
cin>>ch1;
switch(ch1)
{
case 1:
s=ymin+0.01;
delay(100);
cleardevice();
while(s<=ymax)
{
ints(s);
sort(s);

```

```

s++;
}
break;
case 2:
exit(0);
}
cout<<"Do you want to continue?: ";
cin>>ch;
}while(ch=='y' || ch=='Y');
}
void poly::ints(float z)
{
int x1,x2,y1,y2,temp;
c=0;
for(int i=0;i<v;i++)
{
x1=p[i].x;
y1=p[i].y;
x2=p[i+1].x;
y2=p[i+1].y;
if(y2<y1)
{
temp=x1;
x1=x2;
x2=temp;
temp=y1;
y1=y2;
y2=temp;
}
if(z<=y2&& z>=y1)
{
if((y1-y2)==0)
x=x1;
else
{
x=((x2-x1)*(z-y1))/(y2-y1);
x=x+x1;
}
if(x<=xmax && x>=xmin)
inter[c++]=x;
}
}
}
void poly::sort(int z) // sorting
{

```

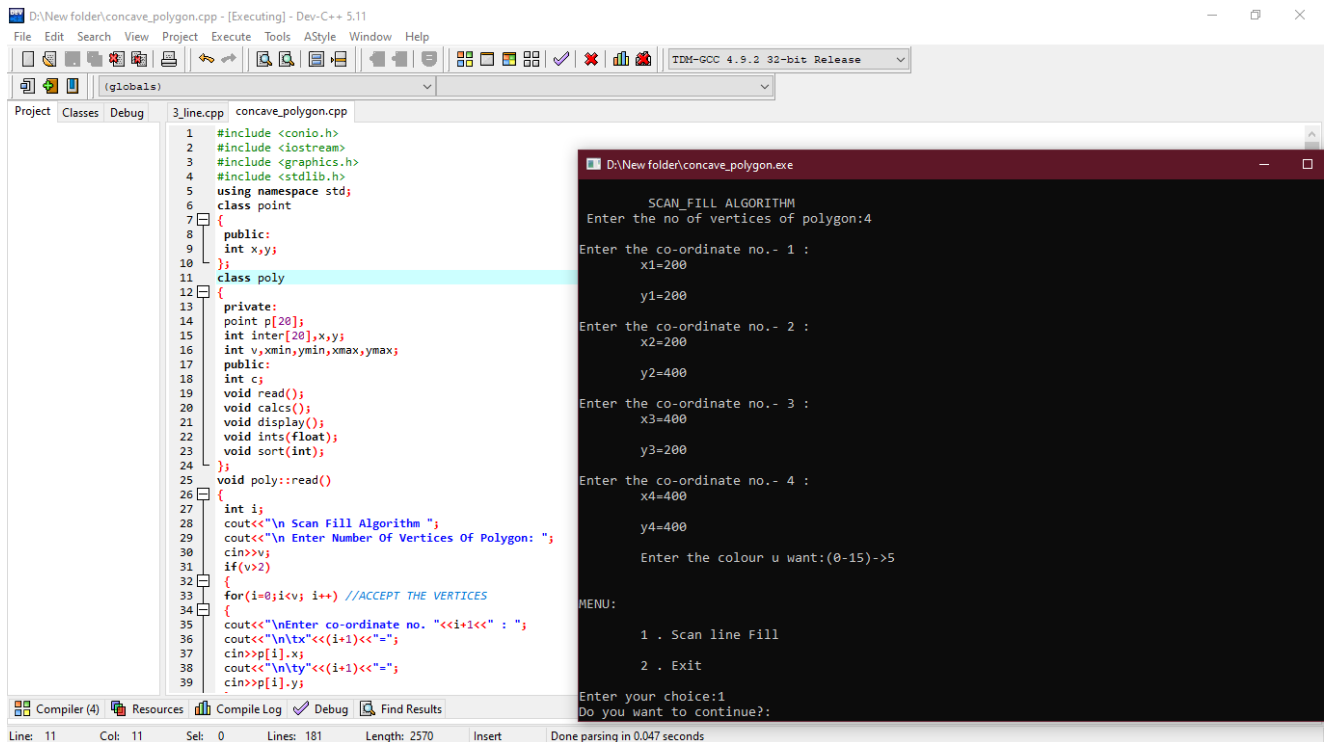
```

int temp,j,i;
for(i=0;i<v;i++)
{
line(p[i].x,p[i].y,p[i+1].x,p[i+1].y);
}
delay(100);
for(i=0; i<c;i+=2)
{
delay(100);
line(inter[i],z,inter[i+1],z);
}
}
int main() //main
{
int cl;
initwindow(500,600);
cleardevice();
poly x;
x.read();
x.calcs();
cleardevice();
cout<<"\n\tEnter The Color You Want :(In Range 0 To 15 )->"; //selecting color
cin>>cl;
setcolor(cl);
x.display();
getch();
closegraph(); //closing graph
return 0;
}
//Input :
//Number of Vertices : 4
//Cordinates 1st :
//x1= 200
//y1= 200
//Cordinates 2st :
//x2= 200
//y2= 40
//Cordinates 3st :
//x3= 400
y3= 200
//Cordinates 4st :
//x4= 400

```

//y4= 400

Output :



The screenshot shows a C++ IDE with the source code for a Scan Fill algorithm. The code defines a `point` class and a `poly` class. The `poly` class has methods for reading vertices, calculating the polygon, displaying it, and sorting the vertices. The `main` function prompts the user to enter the number of vertices, the coordinates of the vertices, and the color of the polygon. It then calls the `Scan Fill` algorithm and displays the result.

```
1 #include <conio.h>
2 #include <iostream>
3 #include <graphics.h>
4 #include <stdlib.h>
5 using namespace std;
6 class point
7 {
8 public:
9     int x,y;
10 };
11 class poly
12 {
13 private:
14     point p[20];
15     int inter[20],x,y;
16     int v,xmin,ymin,xmax,ymax;
17 public:
18     int c;
19     void read();
20     void calcs();
21     void display();
22     void ints(float);
23     void sort(int);
24 };
25 void poly::read()
26 {
27     int i;
28     cout<<"\n Scan Fill Algorithm ";
29     cout<<"\n Enter Number Of Vertices Of Polygon: ";
30     cin>>v;
31     if(v>2)
32     {
33         for(i=0;i<v; i++) //ACCEPT THE VERTICES
34         {
35             cout<<"\nEnter co-ordinate no. "<<i+1<<" : ";
36             cout<<"\n\tx"<<(i+1)<<"=";
37             cin>>p[i].x;
38             cout<<"\nty"<<(i+1)<<"=";
39             cin>>p[i].y;
```

The execution output shows the following steps:

```
SCAN_FILL ALGORITHM
Enter the no of vertices of polygon:4
Enter the co-ordinate no.- 1 :
x1=200
y1=200
Enter the co-ordinate no.- 2 :
x2=200
y2=400
Enter the co-ordinate no.- 3 :
x3=400
y3=200
Enter the co-ordinate no.- 4 :
x4=400
y4=400
Enter the colour u want:(0-15)->5
MENU:
1 . Scan line Fill
2 . Exit
Enter your choice:1
Do you want to continue?:
```

