

DATE:06/09/2023

GROUP A : PRATICAL .1

NAME: SAYALI TANAJI PAWAR

ROLL NO: S213002

CLASS: SE

DIV: C

BATCH: C1

PROBLEM STATEMENT:

Write C++ program to draw a concave polygon and fill it with desired color using scan fill algorithm. Apply the concept of inheritance.

CODE:

```
#include <iostream>
#include <graphics.h>
#include <stdlib.h>
using namespace std;
class point                                //point class is created
{
public:
int x,y;
};
class poly                                //class name poly is created
{
private:
point p[20];                             //array to store polygon
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()                         //func to read polygon vertices
{
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()                    //func to create polygon prop
{
    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()                  //func to display menu
{
    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) //func to find intersection
{
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++) //for filling
{
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;
int gd=DETECT,gm;
initgraph(&gd,&gm,NULL); //initialization of graphics
//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();
closegraph(); //closing graph
getch();
return 0;
}

```

INPUT:

```
d_comp_sli_02@d-comp-sli-02:~$ g++ cg.cpp -o abc -lgraph
d_comp_sli_02@d-comp-sli-02:~$ ./abc
```

Scan Fill Algorithm

Enter Number Of Vertices Of Polygon: [xcb] Unknown sequence number while processing queue

[xcb] Most likely this is a multi-threaded client and XInitThreads has not been called

[xcb] Aborting, sorry about that.

abc: ../../src/xcb_io.c:260: poll_for_event: Assertion

`!xcb_xlib_threads_sequence_lost' failed.

4

Enter co-ordinate no. 1 :

x1=300

y1=300

Enter co-ordinate no. 2 :

x2=300

y2=400

Enter co-ordinate no. 3 :

x3=400

y3=300

Enter co-ordinate no. 4 :

x4=400

y4=400

Enter The Color You Want :(In Range 0 To 15)->10

MENU:

1 . Scan line Fill

2 . Exit

Enter your choice: 1
Do you want to continue?:

OUTPUT:

