Computer Graphics Assignment-2

Filling

Name-Ashish Goyal

Id-2016ucp1100

Batch-A (1, 2)

Boundary Filling Algorithm

Code:

```
from graphics import *
##############################
#   less_one   slope -1 to 1
boundary_color=[]
fill_color=[]
bcolor="black"
fcolor="yellow"
def zero_to_one(x0,y0,x1,y1):  #slope 0 to 1
    a=y1-y0
    b=-1*(x1-x0)
    di=2*a+b
    dne=2*(a+b)
    de=2*a
    y=y0
```

```
    for x in range(x0,x1,1):
        #pixel=Point(x,y)
        #pixel.draw(win_obj)
        win_obj.plot(x,y,bcolor)
        time.sleep(0.02)
        if(di>0):
            y=y+1
            di=di+dne
        else:
            di=di+de
        boundary_color.append([x,y])
        print("point"+"["+str(x)+","+str(y)+"]")
def zero_to_n_one(x0,y0,x1,y1): #slope -1 to 0
    a=y1-y0
    b=-1*(x1-x0)
    di=2*a-b
    dne=2*(a-b)
    de=2*a
    y=y0
    for x in range(x0,x1,1):
        #pixel=Point(x,y)
        #pixel.draw(win_obj)
        win_obj.plot(x,y,bcolor)
        time.sleep(0.02)
        if(di>0):
```

```
            di=di+de
        else:
            y=y-1
            di=di+dne
        boundary_color.append([x,y])
        print("point"+"["+str(x)+","+str(y)+"]")
################################
#  greater_one   slope <-1 and >1
def pure_greater_one(x0,y0,x1,y1):  #slope >1
    b=-1*(y1-y0)
    a=x1-x0
    dne=2*(a+b)
    de=2*a
    di=2*a+b
    x=x0
    for y in range(y0,y1,1):
        #pixel=Point(x,y)
        #pixel.draw(win_obj)
        win_obj.plot(x,y,bcolor)
        time.sleep(0.02)
        if(di>0):
            x=x+1
            di=di+dne
        else:
            di=di+de
```

```
        boundary_color.append([x,y])
        print("point"+"["+str(x)+","+str(y)+"]")
def less_negative_one(x0,y0,x1,y1):  #slope <-1
    a=x1-x0
    b=-1*(y1-y0)
    di=2*a-b
    dne=2*(a-b)
    de=2*a
    x=x0
    for y in range(y0,y1,1):
        #pixel=Point(x,y)
        #pixel.draw(win_obj)
        win_obj.plot(x,y,bcolor)
        time.sleep(0.02)
        if(di>0):
            di=di+de
        else:
            x=x-1
            di=di+dne
        boundary_color.append([x,y])
        print("point"+"["+str(x)+","+str(y)+"]")
################################

def less_one(x0,y0,x1,y1):  #slope -1 to 1
    a=y1-y0
```

```
    b=-1*(x1-x0)
    if(a<0):
        zero_to_n_one(x0,y0,x1,y1)
    else:
        zero_to_one(x0,y0,x1,y1)

#the greater_one cases are mirror image of less_one cases so simply replace x and y

def greater_one(x0,y0,x1,y1):  #slope > 1 and <-1
    a=x1-x0
    b=-1*(y1-y0)
    if(a<0):
        less_negative_one(x0,y0,x1,y1)
    else:
        pure_greater_one(x0,y0,x1,y1)




def helper(x0,y0,x1,y1):
    boundary_color.append([x0,y0])
    initial_point=Text(Point(x0,y0),"("+str(x0)+","+str(y0)+")")
    initial_point.draw(win_obj)
    boundary_color.append([x1,y1])
    final_point=Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")")
```

```
    final_point.draw(win_obj)
    pixel=Point(x0,y0)
    pixel.draw(win_obj)
    pixel=Point(x1,y1)
    pixel.draw(win_obj)

    if(abs(x0-x1)<abs(y0-y1)):              #slope > 1 and <-1
        if(y1>y0):
            greater_one(x0,y0,x1,y1)
        else:
            greater_one(x1,y1,x0,y0)
    else:
        if(x1>x0):                          #slope -1 to 1
            less_one(x0,y0,x1,y1)
        else:                               #we always increase x by 1 therefore start point should always less, so swap both points
            less_one(x1,y1,x0,y0)



#A GraphWin object represents a window on the screen
```

```
t=int(input("enter sides:"))
lista=[]
for i in range(t):
    x01=int(input("enter "+str(i+1)+" point x coordinate x0:"))
    y01=int(input("enter "+str(i+1)+"th point y coordinate y0:"))
    list_temp=[x01,y01]
    lista.append(list_temp)
list_ini_point=lista[0]
x0=list_ini_point[0]
y0=list_ini_point[1]


x01=x0
y01=y0



win_obj=GraphWin("Boundary Fill User Window",700,700) #set viewport size  700,700 are device coordinates
win_obj.setBackground("Light Green")
win_obj.setCoords(-350,-350,350,350) #set window  use coordinates are set
x_axis=Line(Point(-350,0),Point(350,0))   #obj for x axis
y_axis=Line(Point(0,-350),Point(0,350))    #obj for y axis


x_axis.setOutline("Black")
y_axis.setOutline("Black")
x_axis.setArrow('both')
```

```
y_axis.setArrow('both')
x_axis.draw(win_obj)
y_axis.draw(win_obj)

info_x=Text(Point(320,-10),"+x axis")
info_x.draw(win_obj)
info_nx=Text(Point(-320,-10),"-x axis")
info_nx.draw(win_obj)
info_y=Text(Point(0,330),"+y axis")
info_y.draw(win_obj)
info_ny=Text(Point(0,-330),"-y axis")
info_ny.draw(win_obj)

origin=Text(Point(-10,-10),"origin")
origin.draw(win_obj)

j=1
while (j!=t):
    list_t=lista[j]
    x02=list_t[0]
    y02=list_t[1]
    j=j+1
    helper(x01,y01,x02,y02)
    x01=x02
    y01=y02
```

```
helper(x01,y01,x0,y0)




print("<-select seed inside the diagram->")

seed_point=(win_obj.getMouse())
stack=[]
a=int(seed_point.getX())
b=int(seed_point.getY())
stack.append([a,b])

while len(stack)!=0:
    temp_point=stack[len(stack)-1]
    stack.pop()
    a=temp_point[0]
    b=temp_point[1]
    print(temp_point)
    li=[a,b]
    if( (li not in boundary_color) and (li not in fill_color) ):
        win_obj.plot(a,b,fcolor)
        fill_color.append(li)
        if (([a+1,b] not in boundary_color) and ([a+1,b] not in fill_color)):
            stack.append([a+1,b])
```

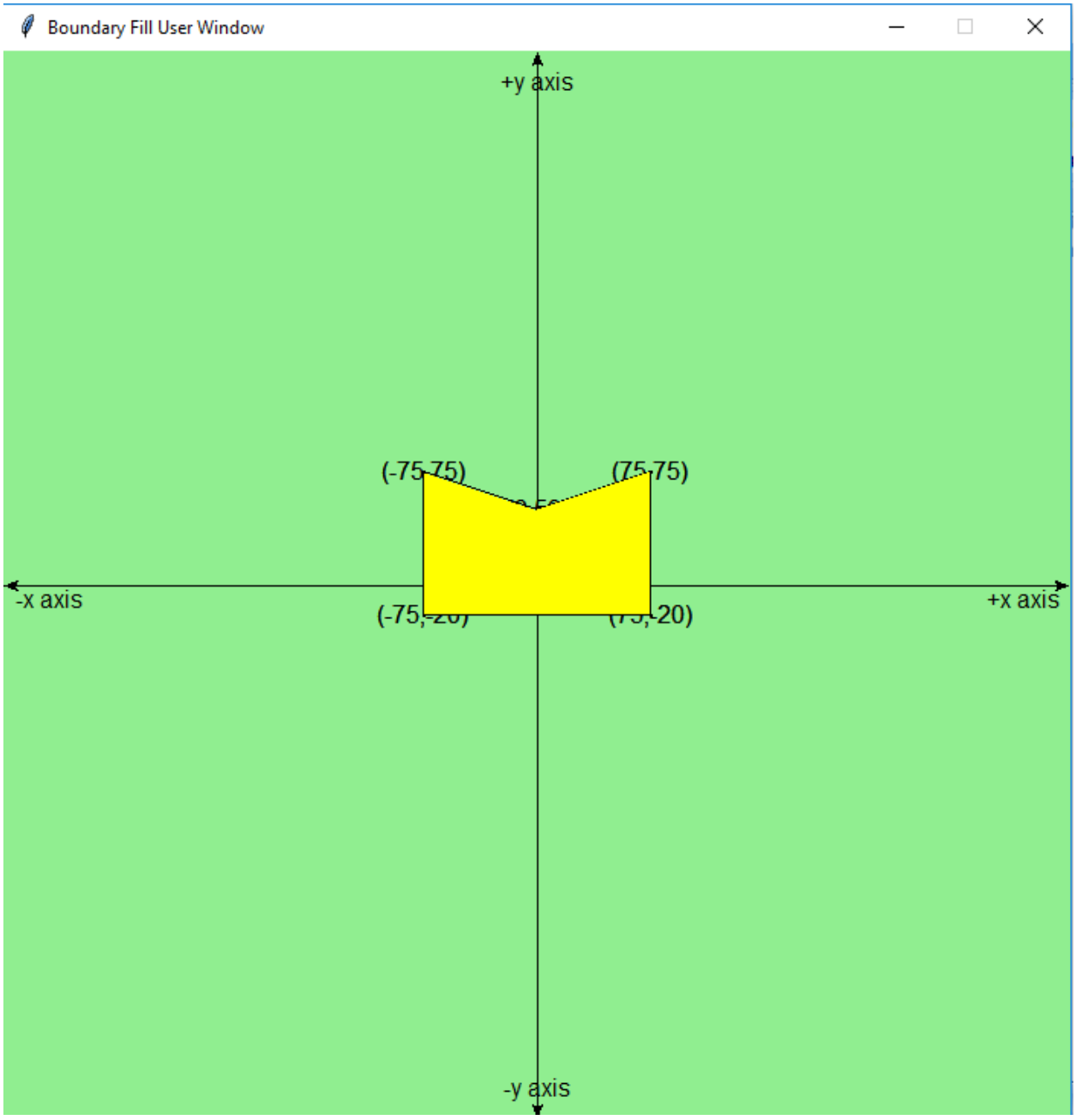
```
    if (([a,b+1] not in boundary_color) and ([a,b+1] not in fill_color)):
        stack.append([a,b+1])
    if (([a-1,b] not in boundary_color) and ([a-1,b] not in fill_color)):
        stack.append([a-1,b])
    if (([a,b-1] not in boundary_color) and ([a,b-1] not in fill_color)):
        stack.append([a,b-1])



win_obj.getMouse()
win_obj.close()
```

Example1:

```
============ RESTART: C:\Users\Ashish\De
enter sides:5
enter 1 point x coordinate x0:75
enter 1th point y coordinate y0:-20
enter 2 point x coordinate x0:75
enter 2th point y coordinate y0:75
enter 3 point x coordinate x0:0
enter 3th point y coordinate y0:50
enter 4 point x coordinate x0:-75
enter 4th point y coordinate y0:75
enter 5 point x coordinate x0:-75
enter 5th point y coordinate y0:-20
```


Boundary Fill User Window
+y axis
(-75,75)
(75,75)
-x axis
+x axis
(-75,-20)
-y axis

Scanline Polygon Filling Algorithm

Code:

```
from graphics import *
##############################
#   less_one   slope -1 to 1
def zero_to_one(x0,y0,x1,y1):  #slope 0 to 1
    a=y1-y0
    b=-1*(x1-x0)
    di=2*a+b
    dne=2*(a+b)
    de=2*a
    y=y0
    for x in range(x0,x1,1):
        pixel=Point(x,y)
        pixel.draw(win_obj)
        #time.sleep(0.02)
        if(di>0):
            y=y+1
            di=di+dne
        else:
            di=di+de
        print("point"+"["+str(x)+","+str(y)+"]")
def zero_to_n_one(x0,y0,x1,y1): #slope -1 to 0
    a=y1-y0
    b=-1*(x1-x0)
```

```
    di=2*a-b
    dne=2*(a-b)
    de=2*a
    y=y0
    for x in range(x0,x1,1):
        pixel=Point(x,y)
        pixel.draw(win_obj)
        #time.sleep(0.02)
        if(di>0):
            di=di+de
        else:
            y=y-1
            di=di+dne
        print("point"+"["+str(x)+","+str(y)+"]")
################################
#  greater_one   slope <-1 and >1
def pure_greater_one(x0,y0,x1,y1):  #slope >1
    b=-1*(y1-y0)
    a=x1-x0
    dne=2*(a+b)
    de=2*a
    di=2*a+b
    x=x0
    for y in range(y0,y1,1):
        pixel=Point(x,y)
```

```
        pixel.draw(win_obj)
        #time.sleep(0.02)
        if(di>0):
            x=x+1
            di=di+dne
        else:
            di=di+de
        print("point"+"["+str(x)+","+str(y)+"]")
def less_negative_one(x0,y0,x1,y1):  #slope <-1
    a=x1-x0
    b=-1*(y1-y0)
    di=2*a-b
    dne=2*(a-b)
    de=2*a
    x=x0
    for y in range(y0,y1,1):
        pixel=Point(x,y)
        pixel.draw(win_obj)
        #time.sleep(0.02)
        if(di>0):
            di=di+de
        else:
            x=x-1
            di=di+dne
        print("point"+"["+str(x)+","+str(y)+"]")
```

```
################################


def less_one(x0,y0,x1,y1):  #slope -1 to 1
    a=y1-y0
    b=-1*(x1-x0)
    if(a<0):
        zero_to_n_one(x0,y0,x1,y1)
    else:
        zero_to_one(x0,y0,x1,y1)


#the greater_one cases are mirror image of less_one cases so simply replace x and y


def greater_one(x0,y0,x1,y1):  #slope > 1 and <-1
    a=x1-x0
    b=-1*(y1-y0)
    if(a<0):
        less_negative_one(x0,y0,x1,y1)
    else:
        pure_greater_one(x0,y0,x1,y1)
```

```
def helper(x0,y0,x1,y1):
    initial_point=Text(Point(x0,y0),"("+str(x0)+","+str(y0)+")")
    initial_point.draw(win_obj)
    final_point=Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")")
    final_point.draw(win_obj)
    if(abs(x0-x1)<abs(y0-y1)):              #slope > 1 and <-1
        if(y1>y0):
            greater_one(x0,y0,x1,y1)
        else:
            greater_one(x1,y1,x0,y0)
    else:
        if(x1>x0):                          #slope -1 to 1
            less_one(x0,y0,x1,y1)
        else:                           #we always increase x by 1 therefore start point should always less, so swap both points
            less_one(x1,y1,x0,y0)



################################################################################
#######################



def scanline():
   # print("ashish")
    y_curr=500
```

```
aet=[]
#print(len(edge_table))
for i in range(len(edge_table)):
    if y_curr>edge_table[i][1]:
        y_curr=edge_table[i][1]


while len(edge_table)!=0 or len(aet) !=0:
    count_y=0
    for i in range (len(edge_table)):
        if edge_table[i][1]==y_curr:
            count_y=count_y+1
    for i in range(0,count_y):
        for j in range(len(edge_table)):
            if edge_table[j][1]==y_curr:
                break
        aet.append(edge_table[j])
        del edge_table[j:j+1]
    aet.sort()
    count=0
    for i in range(len(aet)):
        if aet[i][3]==y_curr:
            count=count+1
    for i in range (0,count):
        for j in range(len(aet)):
            if(aet[j][3]==y_curr):
```

```
            break
    del aet[j:j+1]
bool_option=1
length=len(aet)-1
for i in range(length):
    x_initial=aet[i][0]
    x_final=aet[i+1][0]
    if (bool_option==1):
        index=x_initial
        while index<x_final :
            win_obj.plot(index,y_curr,"yellow")
            index=index+1
        bool_option=0
    else:
        bool_option=1
y_curr=y_curr+1
for i in range(len(aet)):
    aet[i][0]=aet[i][0]+aet[i][4]
```

```
##############################################################################
######################

#A GraphWin object represents a window on the screen




t=int(input("enter sides:"))
lista=[]
for i in range(t):
    x01=int(input("enter "+str(i+1)+" point x coordinate x0:"))
    y01=int(input("enter "+str(i+1)+"th point y coordinate y0:"))
    list_temp=[x01,y01]
    lista.append(list_temp)
list_ini_point=lista[0]
x0=list_ini_point[0]
y0=list_ini_point[1]
lista.append(list_ini_point)
x01=x0
y01=y0



win_obj=GraphWin("Scanline Polygon filling User Window",700,700) #set viewport size
700,700 are device coordinates
win_obj.setBackground("Light Green")
win_obj.setCoords(-350,-350,350,350) #set window  use coordinates are set
```

```
x_axis=Line(Point(-350,0),Point(350,0))   #obj for x axis
y_axis=Line(Point(0,-350),Point(0,350))    #obj for y axis

x_axis.setOutline("Black")
y_axis.setOutline("Black")
x_axis.setArrow('both')
y_axis.setArrow('both')
x_axis.draw(win_obj)
y_axis.draw(win_obj)

info_x=Text(Point(320,-10),"+x axis")
info_x.draw(win_obj)
info_nx=Text(Point(-320,-10),"-x axis")
info_nx.draw(win_obj)
info_y=Text(Point(0,330),"+y axis")
info_y.draw(win_obj)
info_ny=Text(Point(0,-330),"-y axis")
info_ny.draw(win_obj)

origin=Text(Point(-10,-10),"origin")
origin.draw(win_obj)
edge_table=[]
j=0
while (j!=t):
    list_t1=lista[j]
```

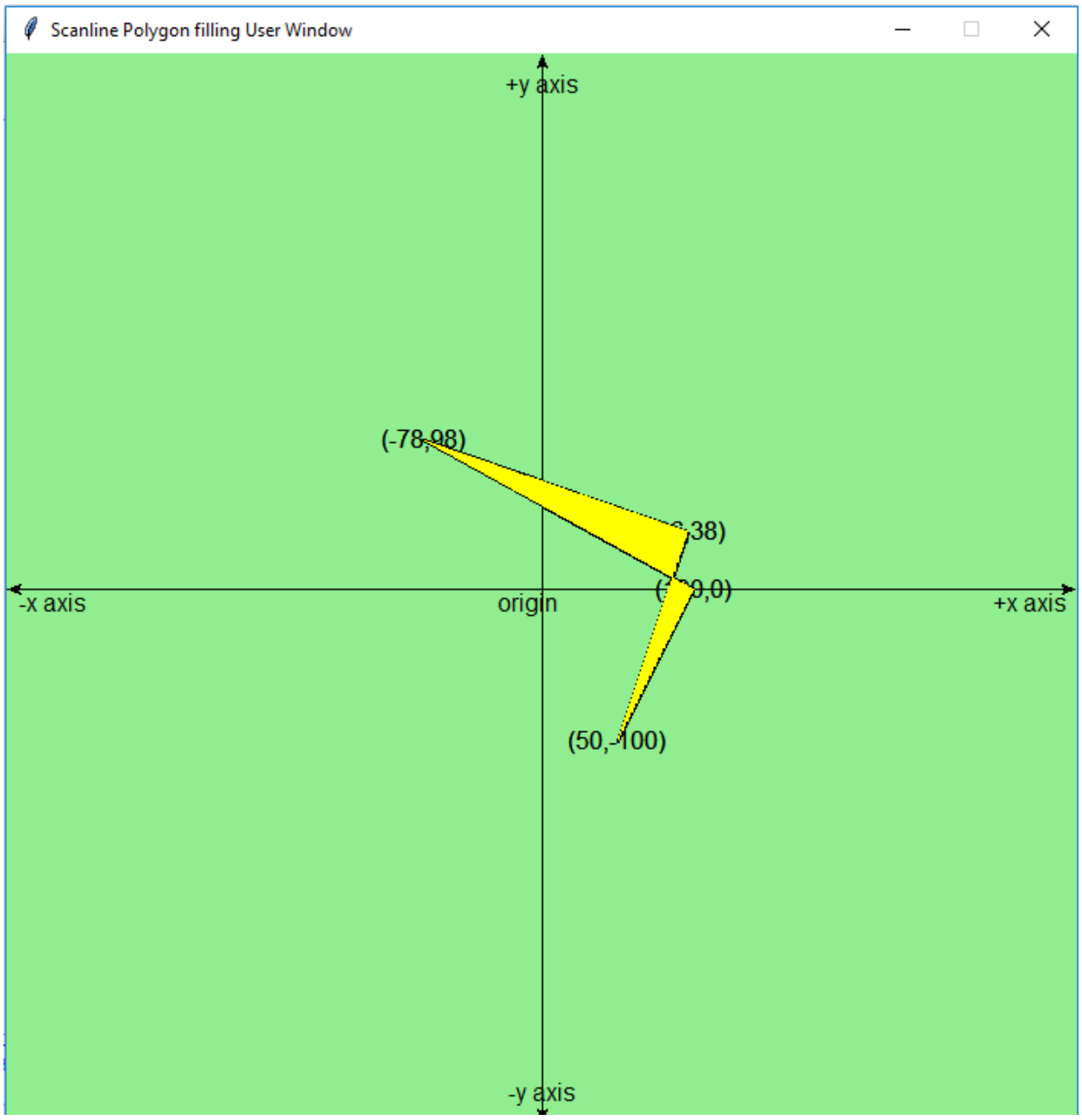
```
    list_t2=lista[j+1]
    x01=list_t1[0]
    y01=list_t1[1]
    x02=list_t2[0]
    y02=list_t2[1]
    if (y01!=y02):
        if(y01>y02):
            x01,x02=x02,x01
            y01,y02=y02,y01
        m=(x02-x01)/(y02-y01)
        edge_table.append([x01,y01,x02,y02,m])

    helper(x01,y01,x02,y02)
    j=j+1
print(edge_table)
scanline()

win_obj.getMouse()
win_obj.close()
```

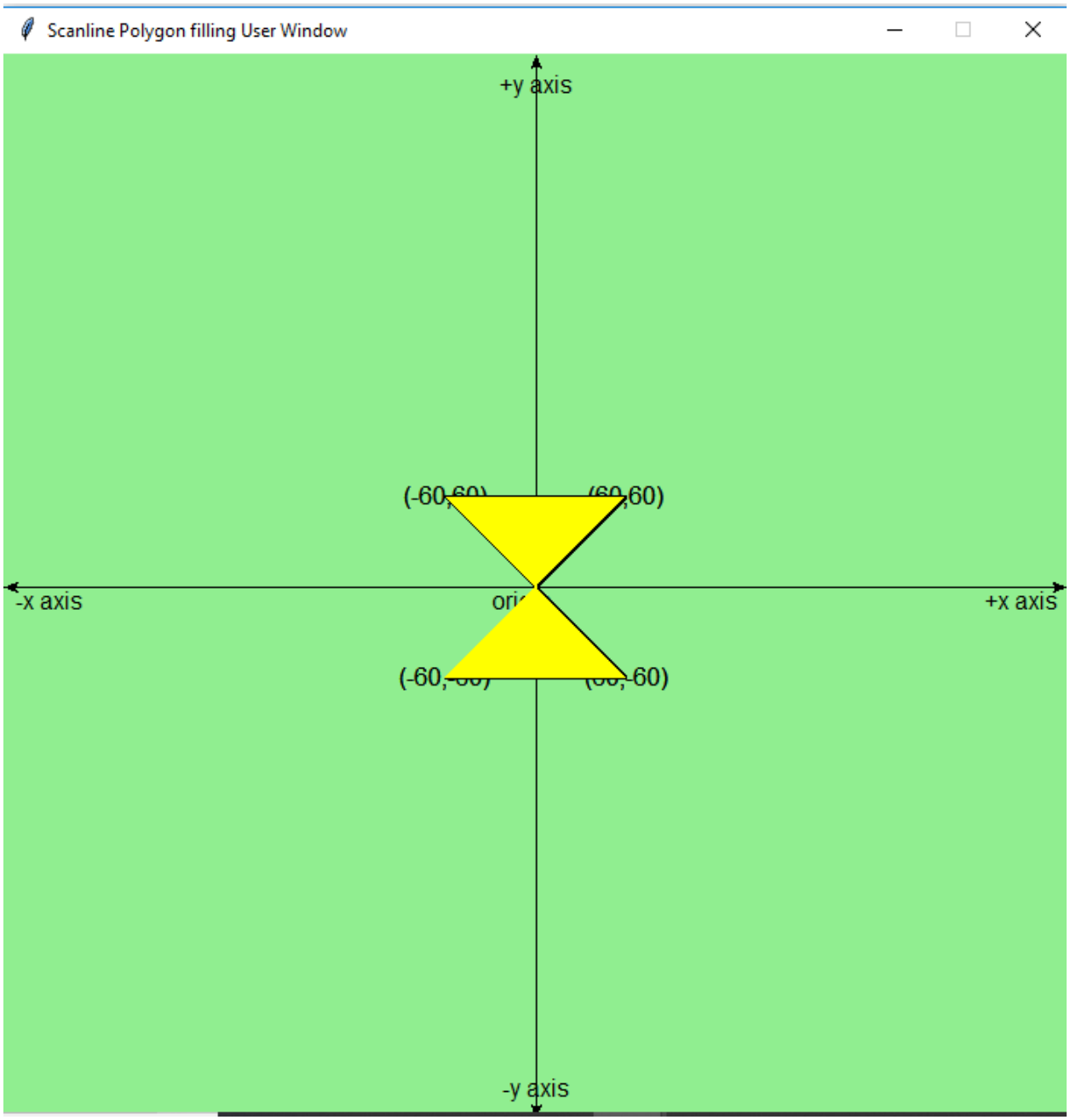
Example2:

```
============== RESTART: C:\Users\Ashish\
enter sides:4
enter 1 point x coordinate x0:100
enter 1th point y coordinate y0:0
enter 2 point x coordinate x0:50
enter 2th point y coordinate y0:-100
enter 3 point x coordinate x0:96
enter 3th point y coordinate y0:38
enter 4 point x coordinate x0:-78
enter 4th point y coordinate y0:98
```



Example3:

```
============== RESTART: C:\Users\Ashish
enter sides:4
enter 1 point x coordinate x0:60
enter 1th point y coordinate y0:60
enter 2 point x coordinate x0:-60
enter 2th point y coordinate y0:-60
enter 3 point x coordinate x0:60
enter 3th point y coordinate y0:-60
enter 4 point x coordinate x0:-60
enter 4th point y coordinate y0:60
```


Scanline Polygon filling User Window
+y axis
(-60,60)
(60,60)
-x axis
+x axis
(-60,-60)
(60,-60)
-y axis