

## Assignment - 4

Name: Tanmay Karmarkar  
Roll No.: 21143

Performance Date:  
Submission Date:

### Problem Statement:

Write C++ program to draw polygon & fill it with desired colour using scan fill algorithm.

### Objective:

To learn how to fill using scan fill algorithm.

Prerequisites: Concept of inheritance.

### Theory:

#### Polygon:

A polygon is a closed planar path composed of a finite no of sequential line segments. A polygon is a 2-dimensional shape formed with more than 3-line segments.

#### Types of polygon:

1. Concave polygon
2. Convex polygon
3. Complex polygon

1. A convex polygon is a simple polygon whose interior is a convex set. All the interior angles are less than  $180^\circ$ .

2. Any line segment joining any 2 points inside lies inside polygon.

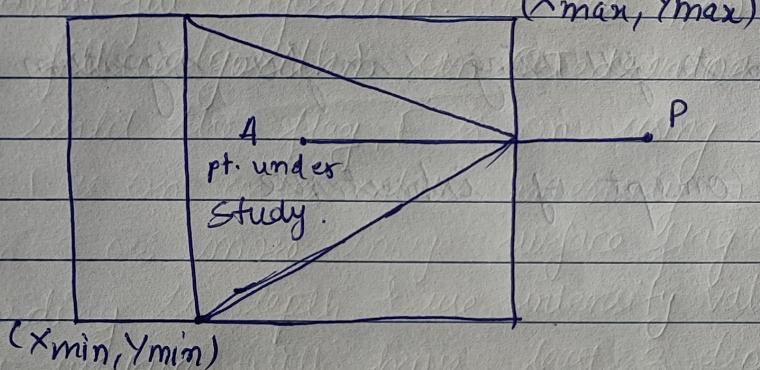
1. A concave polygon will have an angle greater than  $180^\circ$ .

One can draw at least one line through a concave polygon that crosses 2 or more sides.

1. A complex polygon whose sides cross over each other one or more times.

• Inside Outside Test :

1. We assume vertices of polygon are stored & proceed as follows.
2. Draw any point outside  $x_{min}$  to  $x_{max}$ ,  $y_{min}$  to  $y_{max}$ . Draw scan line through P upto the pt. under study.



3. If scan line does not cross any vertex its contribution is equal to no of times it crosses. (C).

4. If C is odd then A lies inside polygon if even lies outside.

• Polygon filling :

For filling, you need to determine pixels on the border & inside of polygon.

• Scan Fill Algorithm:

A scan line fill of a region is performed by first determining the intersection positions of the boundaries of the fill region with the screen scan lines. Then the fill colours are applied to each section of scan lines that lie in the interior. It identifies the same interior regions as the odd even rule.

### Algorithm:

1. Start
2. Create a polygon table having colour, edge pointers, coefficients.
3. Establish edge table content information regarding the endpoints of edges, pointer to polygon, inverse slope,
4. Create active edge list (AEL) sorted in increasing order of x.
5. Create a flag F. It will have 2 values either on (or) off.

### Scan Line:

1. Enter values in (AEL) in sorted value using y as parameter.
2. Scan until the flag, F is on using background.
3. When F is on enter colour intensity I into refresh buffer.
4. When 2 or more image surface flags are on, sort surfaces according to depth & use intensity value  $S_n$  for the  $n^{th}$  surface. This surface will have least 3 depth value.
5. Stop.

### Test Cases:

Input	Exp O/P	Actual O/P	Result
1. Square	Shape filled properly	Shape filled properly	Pass
2. Concave poly.	Shape filled properly	Shape filled properly	Pass
3. Concave poly	Shape filled properly	Shape filled properly	Pass

4. Complex poly



shape filled  
properly

shape filled  
properly

Pass

Conclusion:

We learnt about polygons, types of polygon, even-odd test, scan fill algorithm. We are now able to implement scan fill algorithm in QT creator using C++.

Holes

No holes

No hole

Two

Two

One hole

One hole

Two

Two holes

Two holes

Two

Two

Two holes

Two holes

Two

Two

Two holes

Two holes

Two