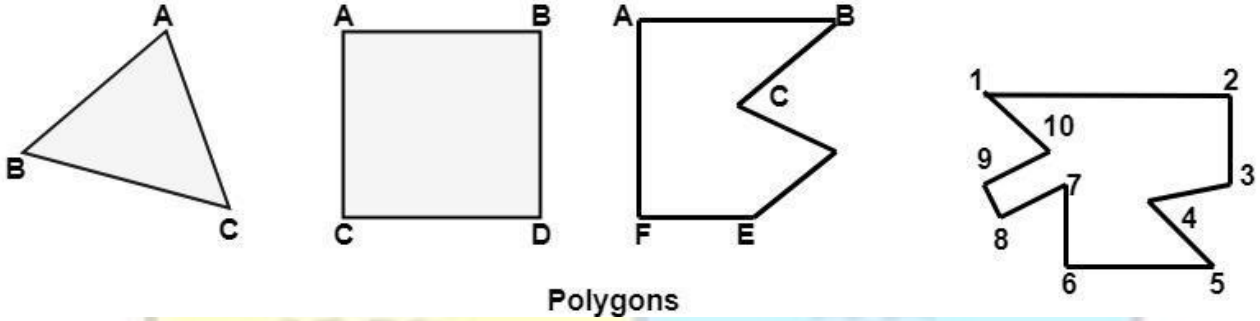
Assginment No. 1

Title	A concave polygon filling using scan fill algorithm
Aim/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.
CO Mapped	
Pre-requisite	1. Basic programming skills of C++ 2. 64-bit Open source Linux 3. Open Source C++ Programming tool like G++/GCC
Learning Objective	To learn scanline polygon fill algorithm.

Theory:

Polygon:

A polygon is a closed planar path composed of a finite number of sequential line segments. A polygon is a two-dimensional shape formed with more than three straight lines. When starting point and terminal point is same then it is called polygon.



Polygons

Types of Polygons

1. Concave
2. Convex
3. Complex

A convex polygon is a simple polygon whose interior is a convex set. In a convex polygon, all interior angles are less than 180 degrees.

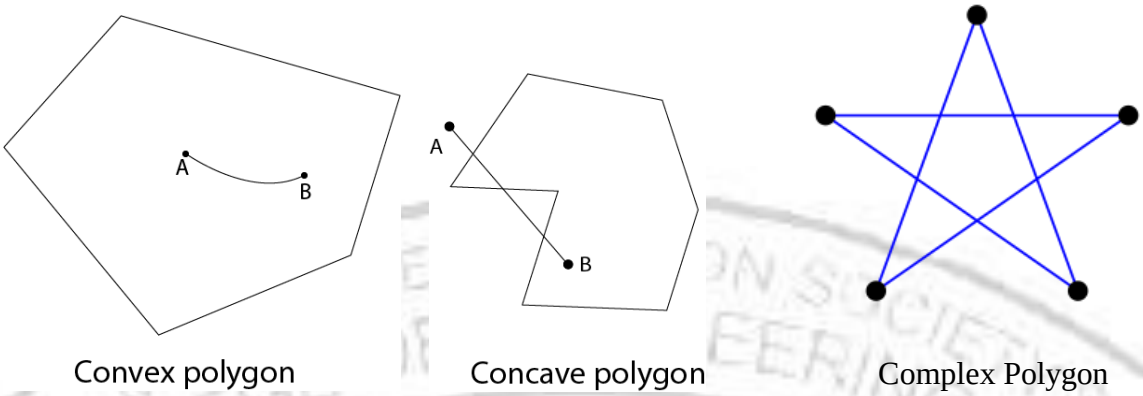
The following properties of a simple polygon are all equivalent to convexity:

- Every internal angle is less than or equal to 180 degrees.
- Every line segment between two vertices remains inside or on the boundary of the polygon.

Convex Polygons: In a convex polygon, any line segment joining any two inside points lies inside the polygon. A straight line drawn through a convex polygon crosses at most two sides.

A concave polygon will always have an interior angle greater than 180 degrees. It is possible to cut a concave polygon into a set of convex polygons. You can draw at least one straight line through a concave polygon that crosses more than two sides.

Complex polygon is a polygon whose sides cross over each other one or more times.

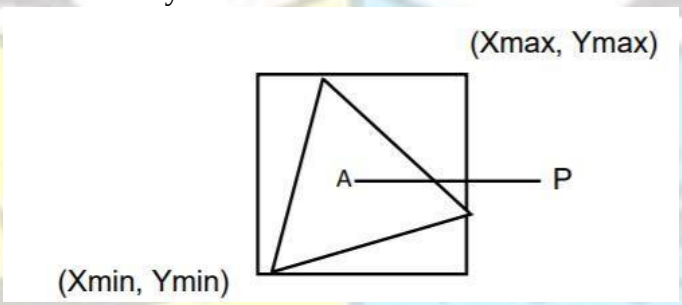


Convex polygon Concave polygon Complex Polygon

Inside outside test (Even- Odd Test):

We assume that the vertex list for the polygon is already stored and proceed as follows.

1. Draw any point outside the range X_{min} and X_{max} and Y_{min} and Y_{max}. Draw a scan line through P up to a point A under study



2. If this scan line

i) Does not pass through any of the vertices then its contribution is equal to the number of times it intersects the edges of the polygon. Say C if

a) C is odd then A lies inside the polygon.

b) C is even then it lies outside the polygon.

ii) If it passes through any of the vertices then the contribution of this intersection say V is,

a) Taken as 2 or even. If the other points of the two edges lie on one side of the scan line.

b) Taken as 1 if the other end points of the 2 edges lie on the opposite sides of the scan- line. c) Here will be total contribution is C + V.

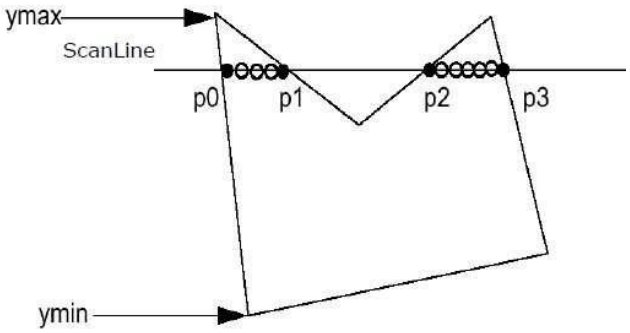
Polygon Filling:

For filling polygons with particular colors, you need to determine the pixels falling on the border of the polygon and those which fall inside the 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 ν Then the fill colors are applied to each section of a scan line that lies within the interior of the fill region ν The scan-line fill algorithm identifies the same interior regions as the odd-even rule.

It is an image space algorithm. It processes one line at a time rather than one pixel at a time. It uses the concept area of coherence. This algorithm records edge list, active edge list. So accurate bookkeeping is necessary. The edge list or edge table contains the coordinate of two endpoints. Active Edge List (AEL) contain edges a given scan line intersects during its sweep. The active edge list (AEL) should be sorted in increasing order of x. The AEL is dynamic, growing and shrinking.



Algorithm

Step1: Start algorithm

Step2: Initialize the desired data structure

1. Create a polygon table having color, edge pointers, coefficients
2. Establish edge table contains information regarding, the endpoint of edges, pointer to polygon, inverse slope.
3. Create Active edge list. This will be sorted in increasing order of x.
4. Create a flag F. It will have two values either on or off.

Step3: Perform the following steps for all scan lines

1. Enter values in Active edge list (AEL) in sorted order using y as value
2. Scan until the flag, i.e. F is on using a background color
3. When one polygon flag is on, and this is for surface S1enter color intensity as I1into refresh buffer
4. When two or image surface flag are on, sort the surfaces according to depth and use intensity value Sn for the nth surface. This surface will have least z depth value
5. Use the concept of coherence for remaining planes.

Step4: Stop Algorithm

Conclusion:

Questions:

1. Which are the different approaches to fill a polygon?
2. What are advantages and drawbacks of scan line polygon fill algorithm?