

Assignment No: 08

Title/ Problem Statement:

Write C++/Java program to draw a concave polygon and fill it with desired pattern using scan line algorithm.

Prerequisites:

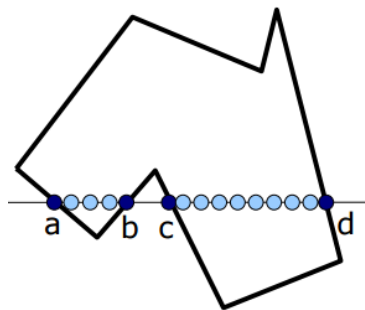
Objectives:

1. To understand Scan Line drawing algorithms used for polygon filling computer graphics.
2. To understand concept of different line styles using line algorithms for filling polygon with desired pattern.

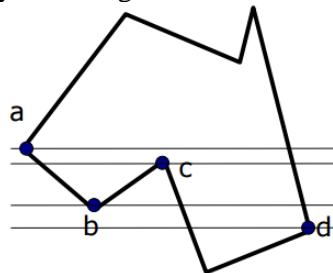
Theory :

Scan-line Polygon Fill

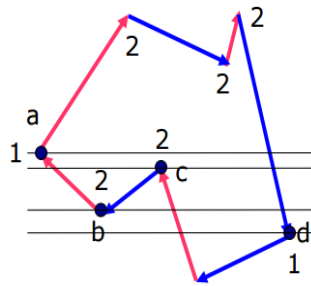
- For each scan-line:
 - Locate the intersection of the scan line with the edges ($y=y_s$)
 - Sort the intersection points from left to right.
 - Draw the interiors intersection points pairwise. (a-b), (c-d)
- Problem with corners. Same point counted twice or not?



- a , b , c and d are intersected by 2 line segments each.



- Count b,c twice but a and d once. Why?

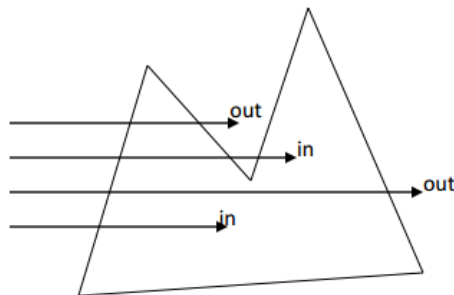


Solution: Make a clockwise or counter-clockwise traversal on edges. Check if y is monotonically increasing or decreasing. If direction changes, double intersection, otherwise single intersection.

Basic algorithm:

- Assume scan line start from the left and is outside the polygon.
- When intersect an edge of polygon, start to color each pixel (because now we're inside the polygon), when intersect another edge, stop coloring ...
- Odd number of edges: inside
- Even number of edges: outside

- **Advantage of scan-line fill:** It does fill in the same order as rendering, and so can be pipelined.



Scan-Line Polygon Fill Algorithm

• Odd-parity rule

Calculate span extrema (intersections) on each scan line

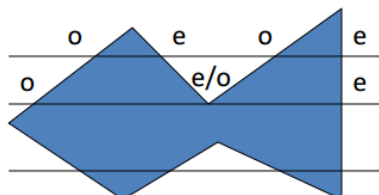
Using parity rule to determine whether or not a point is inside a region

Parity is initially even

◊each intersection encountered thus inverts the parity bit

parity is odd -> interior point (draw)

parity is even-> exterior point (do not draw)



setlinestyle

Syntax

```
#include <graphics.h>
```

```
void setlinestyle(int linestyle, unsigned upattern, int thickness);
```

Description

setlinestyle sets the style for all lines drawn by line, lineto, rectangle, drawpoly, and so on. The linesettingstype structure is defined in graphics.h as follows:

```
struct linesettingstype
{
    int linestyle;

    unsigned upattern;

    int thickness;
};
```

linestyle specifies in which of several styles subsequent lines will be drawn (such as solid, dotted, centered, dashed). The enumeration line_styles, which is defined in graphics.h, gives names to these operators:

Name	Value	Description
SOLID_LINE	0	Solid line
DOTTED_LINE	1	Dotted line
CENTER_LINE	2	Centered line
DASHED_LINE	3	Dashed line
USERBIT_LINE	4	User-defined line style

thickness specifies whether the width of subsequent lines drawn will be normal or thick.

Name	Value	Description
NORM_WIDTH	1	1 pixel wide
THICK_WIDTH	3	3 pixels wide

upattern is a 16-bit pattern that applies only if linestyle is USERBIT_LINE (4). In that case, whenever a bit in the pattern word is 1, the corresponding pixel in the line is drawn in the current drawing color. For example, a solid line corresponds to a upattern of 0xFFFF (all pixels drawn), and a dashed line can correspond to a upattern of 0x3333 or 0x0F0F. If the linestyle parameter to setlinestyle is not USERBIT_LINE (in other words, if it is not equal to 4), you must still provide the upattern parameter, but it will be ignored.

Note: The linestyle parameter does not affect arcs, circles, ellipses, or pie slices. Only the thickness parameter is used.

Return Value

If invalid input is passed to setlinestyle, graphresult returns -11, and the current line style remains unchanged.

Output:

Polygon is filled with pattern.

Conclusion: Thus we have studied C++/Java program to draw a concave polygon and fill it with desired pattern using scan line algorithm.