Name → Ameya Barapatre
Roll No – 06
SE – DS

Experiment No – 6

Aim – To implement Scan line polygon filling algorithm in Turbo C and to make hexagon fill with colours.

Resource Required – Turbo C, stationary, pointer.

Theory – Scan line polygon filling algorithm is a method Used in C.G to fill polygons with solid color or pattern by determining which pixels within a given polygon should be colored.

- It identifies which segment of each scan line lie inside the polygon and then filling these segments.
- It computes the visible scan of each scan-line within polygon boundaries.
- Works with any type of polygon ie. Convex, concave, complex polygon.

1. Scan lines – Horizontal line that intersects a polygon. Processes one scan line at a time.

2. Edge Table (ET) – Used to keep track of the edges of the polygon and how they interact with the scan lines.

Edge table contains →

Ymin - edge starts on the scanline

Ymax - edge ends on the scan line.

X coordinate at Ymin — X coord where edge intersects the scan line.

The inverse of slope (1/m) determines have X - coordinate changes.

3. AET - Active Edge Table - Begins and ends with null-node.

AEL - Active Edge List contains all the edges containing the scan lines intersecting.

Adds new edges when scan line reaches Ymin

Removes when line passes Ymax.

4. Filling the Polygon → AET and AEL is used to determine which scan line are inside the polygon.

- It sorts the x intersecting pixels and then b/w those fill colors.
- For multiple edges alternate pairs of intersections represents start and end inside the polygon.

5. Global Edge Table is the edge table (ET) containing Ymax, Ymin, 1/m.

6. Efficiency : It only process the pixels that lie within the polygon. Uses edge table and helps to minimize the compulation needed.

7. Advantages → Easy implementation.
Works for convex / concave polygons.
Algorithm is precise in filling required pixels only.

→ Applications:
2D Computer Graphics : Software rasterizes CAD and Modelling Software for designing shapes and applications.
Video Games & Animation : Used in Graphics Engine for rendering solid objects.

Algorithm→

Step 1 : Sort vertices of Polygon from $Y_{min} \rightarrow Y_{max}$

Step 2 : Create GET / ET for all scanlines by creating linked list, node represented as →

Y_{max}	XY_{min}	$1/m$

Step 3 : Generate AET, initialize with NULL node.

Step 4 : Repeat following steps until AET, GET gets empty.
1) More edge from GET to AET.
2) Fill visible span
3) Remore edge $Y_{max} = Y_{given}$
4) $Y_{n+1} = Y_n + 1$
5) $X_{n+1} = X_n + 1/m$

Conclusion → Successfully implemented the scan line polygon filling algorithm as it effectively determines which pixels die inside a polygon. Handles the convex, complex polygons. Used in CAD, 2D games engins and computational efficiency.