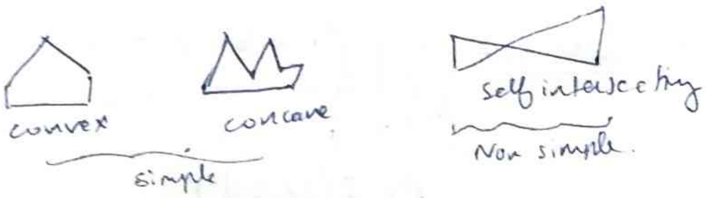
CG Mod 2

Polygon Filling Algorithm

- **Polygon:-** closed fig. represented by a collectn of more than 2 line segments connected end to end
  edges.



Polygon Filling → Boundary Fill
→ Flood Fill
→ Scan-Line Fill

* Boundary Fill

- only if the color with which the region has to be filled & color of boundary of region are diff.
- boundary → one single color.
- 4 connected & 8 connected «algo».

* Flood Fill

- seed point
- multiple colors boundary.
- 4 connected / 8 connected «algo»

```
if (getpixel (x,y) = old color)
{
  ...
}
```

- Disadvantage :-
  → very slow algo → fails for large polygons
  → initial pixel needs more knowlege about surrounding pix.

* Scan line Polygon Fill Algo

- filling regions of a polygon that are geometrically defined by the coordinates of vertices of this polygon graph.
- scans lines at a time (not pixels). ∴ faster.

→ intersection of scan line with edges of the polygon.
→ sort the intersections increasing order of x-coordinates.
→ make pairs & fill color b/w pairs.

- <algo>

1) Y_{min} & Y_{max}
2) each edge of the poly from Y_{min} to Y_{max}, all the edges are intersected by scanline. Each points of intersection are named as $p_0, p_1, p_2 \dots$
3) sort the intersection point in increasing order of x coordinate (p_0, p_1), (p_1, p_2) $(p_2, p_3) \dots$
4) fill all pairs of cordinates inside the polygon & ignore alternate pairs.

$$m = \frac{(y_{k+1} - y_k)}{(x_{k+1} - x_k)}$$

$y_{k+1} - y_k = 1$

$x_{k+1} = x_k + \frac{1}{m}$

(+ rounding to nearest int)

- coherence property:- pply of one part of the scene are related to other parts of the scene.

INSIDE OUT TEST

- Check whether a point lies inside/outside of a polygon
- → Even/odd or Odd-Even / Odd Parity Rule
- → Winding Number Method.

* Even-Odd / Odd Parity Rule

- Crossing number or ray casting algorithm.
- A ray coming from ∞ crosses through border of polygon then it goes from outside to inside & in to out. alternatively.

<algo>

- Consider a line segment from point to examine to out of poly.
- count no. of intersections
- if odd, point = inside
  else, point = outside.
- Time complexity = O(S) S → sides in poly.

* Winding Number Algo / Non-Zero Algo

- score for each intersection with boundary of polygon & sum
- directions assigned counter clockwise
- if edge starts from below the line ⇒ −1
  else → +1
- Time :- O(S)

if non zero, inside
else, outside.

2D transformations

geometric changes of an obj from current state → modified state.

- alters coordinate descriptions of an object

Transformation
- object: Alter coordinate desc. Translate, rotate, ... coordinate system unchanged.
- coordinate: Produce a diff coordinate system

* Translation * Rotation * Scaling * Reflection * Shear.

TRANSLATION

- moves all points in an obj along same straight line path to new pos.

$$P' = P + t$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} tx \\ ty \end{pmatrix}$$

- rigid body transformation → no deformation

ROTATION

- repositions all points in an object along a circular path in the plane centered at the pivot point.
- <derivation>

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$

$$P' = R \cdot P$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- clockwise :-

$$x' = x\cos\theta + y\sin\theta$$
$$y' = -x\sin\theta + y\cos\theta$$

fixed point :- $x' = x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta$

$y' = y_r + (x - x_r)\sin\theta + (y - y_r)\cos\theta$

SCALING

$$P' = S \cdot P$$

$$S = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix}$$

fixed point :-

$$x' = x_f + (x - x_f) \cdot Sx$$
$$y' = y_f + (y - y_f) \cdot Sy$$

Matrix rep.

→ for easy processing

$$P' = \underbrace{M_1}_{\text{multiplicative}} \cdot P + \underbrace{M_2}_{\text{translational}}$$

→ coordinate positions → homogenous coordin. triples (x_h, y_h, h)

$$x = x_h / h$$
$$y = y_h / h$$

$\} \; h = 1$

Translation :- $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

$$P' = T(tx, ty) \cdot P$$

Rotn :- $= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = R(\theta) \cdot P$

Scaling :- $= \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = S(Sx, Sy) \cdot P$

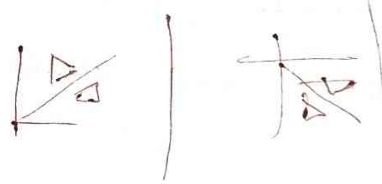
REFLECTION

mirror img.
180° rotn

$$\begin{array}{cc} x & y \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

line

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

SHEARING

aka skewing.
slants the shape of an obj

x, y, xy

$x' = x + sh_x y$
$y' = y$

$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & sh_x & -sh_x \cdot y_{ref} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$x' = x + sh_x (y - y_{ref})$
$y' = y$

3D Translation

$P' = T \cdot P$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D Rotation

$x \to y \to z \to x$.

Axis of rotation :-

* z axis :- $x' = x\cos\theta - y\sin\theta$

  $y' = x\sin\theta + y\cos\theta$

  $z' = z$

* x axis :-

  $y' = y\cos\theta - z\sin\theta$

  $z' = y\sin\theta + z\cos\theta$.

  $x' = x$

* y axis :-

  $z' = z\cos\theta - x\sin\theta$

  $x' = z\sin\theta + x\cos\theta$.

  $y' = y$

Scaling

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

fixed point :- $\begin{bmatrix} S_x & 0 & 0 & (1-S_x)x_f \\ 0 & S_y & 0 & (1-S_y)y_f \\ 0 & 0 & S_z & (1-S_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Reflections

about XY plane :- $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

YZ $\begin{pmatrix} -1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}$ XZ : $\begin{pmatrix} 1 & & & \\ & -1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}$

3D Shear

$$SH_z = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

about X :-

$$SH_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ b & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Y :- $SH_y = \begin{bmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

— x — x — x — x — x —