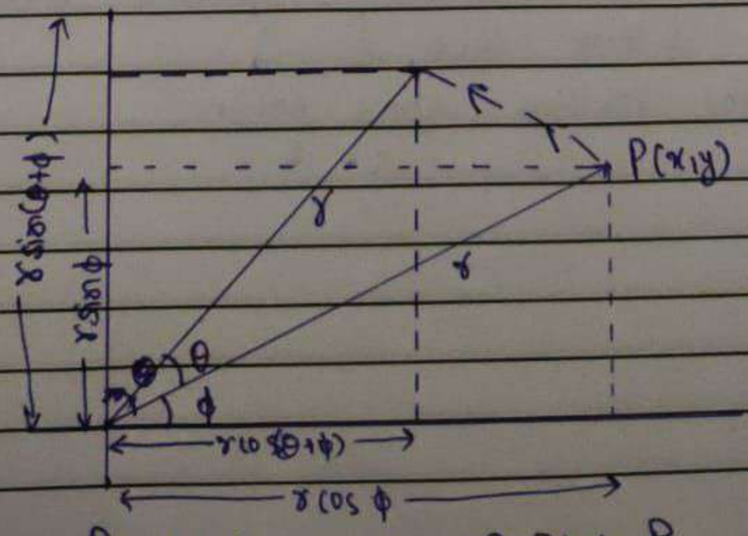
Assignment-1

Q1 Derive matrix for 2D Rotation transformation.

2D rotation is described by repositioning all the points of an object along a circular path in the 2D plane.
To perform rotation we need a rotation angle and the refrence point (x_r, y_r) ~~with re~~ wrt. which object is to be rotated. The refrence point is also called a pivot point. If rotation is performed in an anticlockwise direction, the value of the angle is considered positive, otherwise it is negative.

Rotation about Origin →
lets derive the tranformation matrix for rotation about the origin. As shown in figure $P(x,y)$ is the original point which is to be rotated in XY plane by the angle θ in an anticlockwise direction $P'(x',y')$ is the rotated point.



The distance of P from the origin is let's say r. Consider the angle of P with x-axis is ϕ is rotated by angle θ to an anticlockwise direction. So point P makes angle $(\phi+\theta)$ with x-axis.

$\therefore$ for P $\Rightarrow$ $\sin\phi = \frac{y}{r}$ $\Rightarrow$ $y = r\sin\phi$

$$\cos\phi = \frac{x}{r} \Rightarrow x = r\cos\phi$$

For P' $\Rightarrow$ $\cos(\phi+\theta) = \frac{x'}{r}$ $\Rightarrow$ $x' = r\cos(\phi+\theta)$

$\Rightarrow r(\cos\phi\cos\theta - \sin\phi\cdot\sin\theta)$ $\Rightarrow$ $x' = x\cos\theta - y\sin\theta$.

$\sin(\phi+\theta) = \frac{y'}{r}$

$\therefore y' = r\sin(\phi+\theta) = r(\sin\theta\cos\phi + \sin\phi\cos\theta)$

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

Matrix representation of above equation is written as $\rightarrow$

$P' = R\cdot P.$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

where $R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ represent rotation.

Matrix for rotation about origin

filled by this method are ~~ca~~ called 4-connected/ 4-way connected.

- Algorithm →

```
Boundary-fill (x, y, fcolor, bcolor)
current ← getcolor (x, y)
if current ≠ fcolor && current ≠ bcolor
putpixel (x, y, fcolor)
Boundary-fill (x-1, y, fcolor)
Boundary-fill (x+1, y, fcolor)
Boundary-fill (x, y-1, fcolor)
Boundary-fill (x, y+1, fcolor)
```

	(x, y+1)	
(x-1, y)	(x, y)	(x+1, y)
	(x, y-1)	

Q2. Explain the boundary fill algorithm for 4-way connectivity.

Working Mechanism →

- Boundary fill algorithm starts with some interior pixel of a polygon called seed pixel and keep filling neighbour pixels in ~~outward~~ outward directions until the boundary color is encountered.
- Boundary fill algorithm starts with three parameters:- interior point (x, y), fill color, boundary color.
- This approach retrives the color of the current pixel and compares it with fill color and boundary color.
- If the color of the current pixel is neither fill color nor boundary color, then fill it with the fill color and make a recursive call, otherwise skip the pixel under consideration.
- Neighbour pixel are approached using 4-connectivity or 8-connectivity.
- This method is useful in interactive painting packages where the selection of interior pixel can be done very easily using input device like a mouse.
- To make a solid region, set the fill color to boundary color, so that boundary and interior region becomes indistinguishable after filling.
- Recursively this algorithm checks all pixels in given polygon and fills them if not already filled.
- For 4-connected pixel → After painting a pixel, the function is called for four neighbouring points. These are the pixel positions that are right, left, above and below the current pixel. Area

Q3] Explain different anti-aliasing techniques.

When straight line (except. horizontal and vertical) lines are drawn on monitor, it appears Zig-zag. This effect is Aliasing happens due to poor algorithm or hardware limitations.
Zig-zag lines on screen gives the illusion of smooth line, that effect is called anti-aliasing.

Methods of Anti-Aliasing →
There are four methods of Anti-Aliasing. These methods are mentioned below.
1) High Resolution Display.
2) Post-Filtering ~~(super~~
3) Pre-Filtering
4) Pixel Phasing.

1) Using High-Resolution Display →
One way to reduce the aliasing effect and increase the sampling rate is to simply display object at higher resolution. Using resolution, the jagging become so small that they become indistinguishable from the human eye. Hence, jagged edges get blurred out and edges appear smooth.

2) Post Filtering (Super Sampling) →
In this method, we are increasing the sampling resolution by treating the screen as

it's made of a much more fine grid, due to which the effective pixel size is reduced. But the screen resolution remains the same. Now, intensity from each subpixel is calculated and the average intensity of the pixel is found from the average of intensities of subpixels. Thus we do sampling at a higher resolution and display the image at a lower resolution or resolution of screen, hence this technique is called supersampling. This method is also known as post filtration as this procedure is done after generating the rasterized image.

3] Pre - Filtering (Area Sampling) →
In area sampling, pixel intensities are calculated proportionally to areas of overlap of each pixel with objects to be displayed. Here pixel color is computed based on the overlap of scene's objects with a pixel area.

4] Pixel Phasing →
It's a technique to remove aliasing. Here pixel positions are shifted to nearly approximate position near object geometry. Some system allows the size of individual pixels to be adjusted for distributing intensities which is helpful in pixel phasing.

Q4] What are homogeneous coordinates and discuss the use of it in computer graphics.

Homogenous coordinates are an extension of cartesian coordinates used primarily in projective ~~game~~ geometry and computer graphics. They provide a way to represent points in space that makes mathematical operations like translation, rotation ~~like t~~ and projections more uniform and efficient.

Properties of Homogeneous Coordinate Representation.

- Any 2D point in the homogeneous coordinate system is represented by a triplet (x, y, h) where x, y and h are not all zero. $(0,0,0)$ does not represent any point. Origin is represented by $(0,0,1)$
- In homogeneous coordinate system, two points are ~~ind~~ identical, if one point is derived by multiplying some constant to the ~~tw~~ second point.
- If h is not zero, then point (x_h, y_h, h) in a homogeneous coordinate system is represented as $(x_h, y_h, y_h/h)$ in the ~~Eati~~ Cartesian coordinate system.
- If ~~A is~~ h is 0, point represented is at infinity - Homogeneous representation allows us to write all geometric transformation equation in matrix multiplication form.

- It brings uniformity in operation. Implement of transformation operation in programming language becomes easier with this representation as all the operation are performed using matrix multiplication operation only.

- Basic transformation operation matrix using homogeneous coordinate are shown below→

Translation →

$$P' = T \cdot P$$

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

Rotation →

$$P' = R \cdot P$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scaling →

$$P' = S \cdot P$$

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