3/01/22. 21C32501

21920PE0433 A

Date __/__/____ CQ CS C305

Saathi

Q.1] D

Q.2] D

Q.3] D

Q.4] B

Q.5] A

Q.6] B

Q.7] C

Q.8] B

Q.9] D

Q.10] B

3/01/22

2020PE0433

21CS2001

Date __/__/____

CG CSC305

Saathi

Q7) iii) Rasterization:-

→ It is technique of taking an image described in vector graphics format and transform it into a set of pixels and rendering is the process of generating a photorealistic and non-photorealistic image from 2D or 3D model.

(b) Scan Conversion:-

→ Process of representing graphical objects a collection of pixels.

3/01/22 21CB0001

2020PE0433

Date __/__/____ CG CSC305

Saath

Q2ai) Given: r=4, xc=0, yc=0

Plot 1st point as (0,r) i.e. y=r

Plot(0,4)

Hence plot

plot (4,0)

plot (0,-4)

plot (-4,0)

find p = 1-r

p = 1-4 = -3

till (x<y)

if (p<0)

In this case p = -3

∴ we have to increase x by 1 and modify p = p + 2x +

∴ p = -3 + 2(0) + 1

= -3 + 1 = -2

Now we are not increasing y

Plot (x+1, y)

i.e. (1, 4)

Now, p = -2 i.e. p<0

Increase x by 1

p as p + 2x + 1

p = -2 + 2(1) + 1 = 1

No change in y

Plot (x+1, y) i.e. (2, 4)

Now, p = 1 i.e. p > 0

Increase x by 1, decrease y by 1

p = p + 2(x-y) + 1

p = 1 + 2(2-4) + 1

= -2

Plot (x+1, y-1) i.e. (3, 3)

As x = y, stop here

x	y
0	4
1	4
2	4
3	3

Page No.

3101122 21C30001

2020PBD493

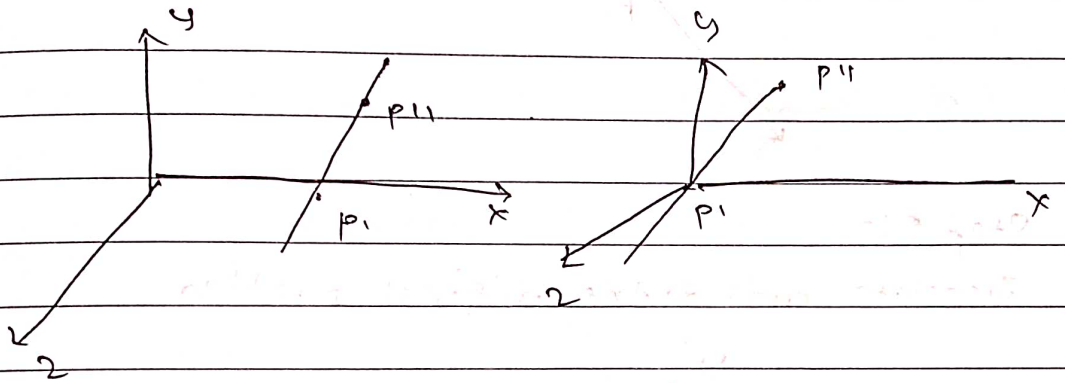
Date __/__/____ CG CSC305

Saathi

Q2(B) ii) When object is rotated about axis that that is not parallel to any one of the cordinates axis which is x, y, z. Then additional transformations are required. Following steps are required

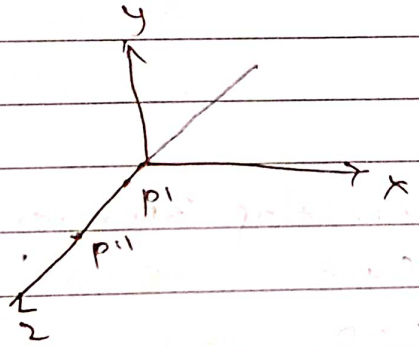
1. Step 1 →

Translate the object to origin:-



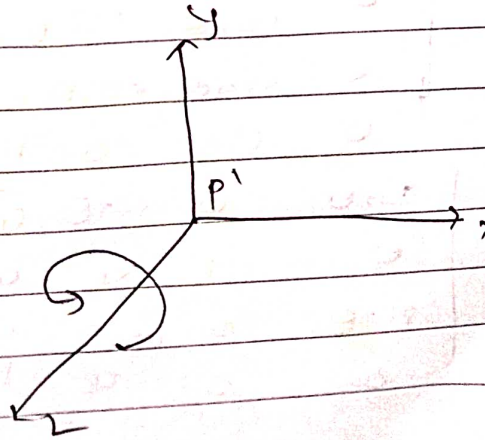
Step 2:-

Rotate object so that the object coincide with any of the cordinate axis



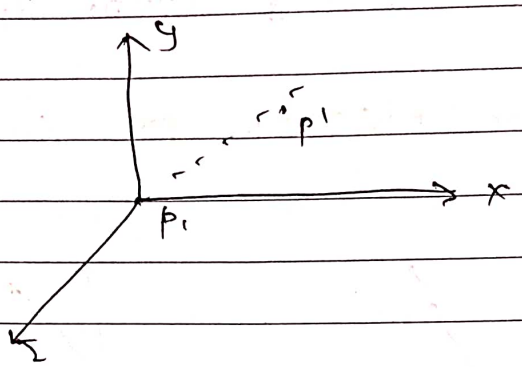
Step 3:-

Perform rotation about cordinate axis which when coincide is done.

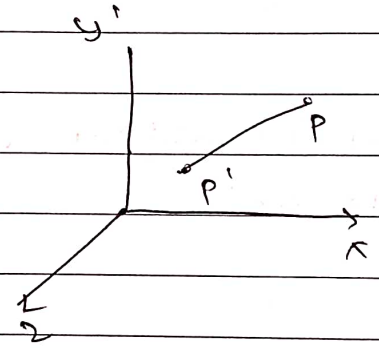


Page No.

Step 4:-
Apply inverse rotation to bring the rotation back to original position



Step 5:-
Translate axis to the original position



Matrix for representing 3D rotation about:-

z-axis:-
$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

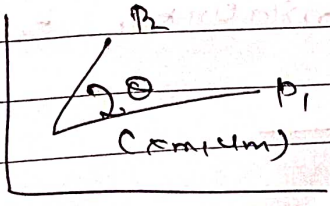
x-axis:-
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

y-axis:-
$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Q.3(iii) Animation refers to the movement on the screen of the display device created by displaying sequence of still images.
- A person who creates animation is called as animator
- Animation refers to the technique of designing, drawing, preperation of photographic static images.
- There are 10 principles of Animation
(a) Staging
(b) Squash and streten
(c) 3D-effect
(d) Timing
(e) Appeal
(f) Overlap
(g) Arcs
(h) Secondary Action
(i) Slow in - Slow out

Q.3(ii) - It is also called as Plane Equation method
- It is an object space method in which objects and parts of objects are compared to find the visible surfaces
- It is used to check if the triangle facing will be facing away from the viewer or not

Q.3(a)(i) When reference point of rotation is other than origin then in that case we have to follow series of transformation which is said to be composite transformation



2021PE0433 3/01/22 21C99001

Date __/__/__ CG CSC305

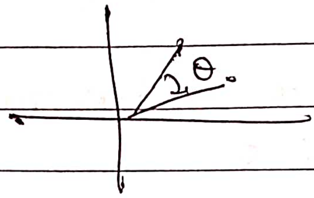
Assume we have to rotate point P_2 with respected to (x_m, y_m) then we have to perform following steps

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_m & -y_m & 1 \end{bmatrix}$$

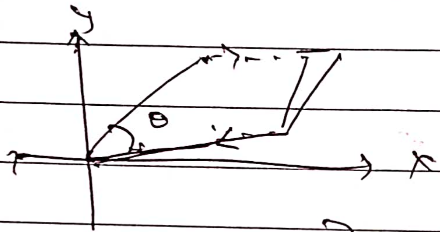
$t_x = -x_m$ $t_y = -y_m$

- then rotate it in clockwise direction :-

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- now translate back to normal



$$t_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_m & y_m & 1 \end{bmatrix}$$

translational * Rotation * translational

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_m & -y_m & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_m & y_m & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_m & -y_m & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ x_m & y_m & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ -x_m\cos\theta - y_m\sin\theta + x_m & x_m\sin\theta - y_m\cos\theta + y_m & 1 \end{bmatrix}$$

2020PE0493 3/01/22 21CS0001

Date __/__/____ CG CSC305

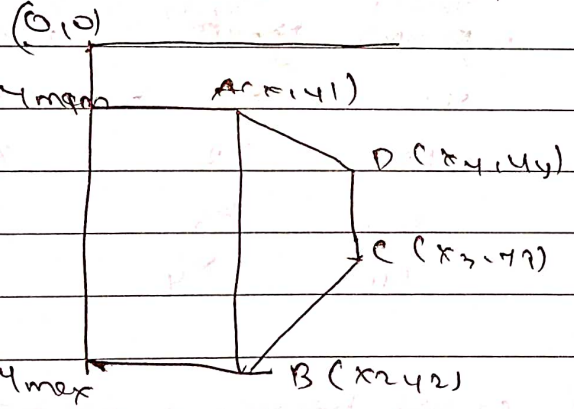
Saathi

Q.3b)ii) Normalizer is an active transformation used to convert single row into multiple column and vice-versa. R

- If in a single row there is repeating data in multiple columns, then it can be split into multiple rows.

Q.3b ii) Scan line polygon is filling up polygons using horizontal lines or scan lines.

- Scan line fill algorithm is defined at geometric level ie cordinates, edges, vertices etc.
- It checks every pixel on that scan line satisfies inside point test or not.
- This method avoid need for seed point for example



- Here the algorithm begins with the first scan line that the polygon occupies ymax and proceeds line by line towards last scan line ymin
- From the above considering the first and last scan line of polygon not. Each edge of polygon are storing five attributes. We are going to store xmax, xmin, ymax, ymin of edges of the polygon

xmax: x2

ymax: y2

xmin: x1 ymin: y1

Page No.

2020PE0433

3/01/22

CSC305

CA

Date __/__/__

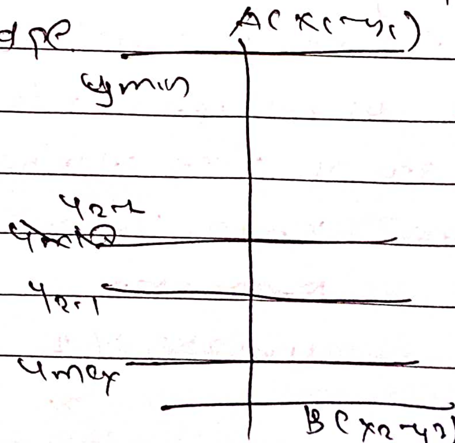
By tabularising it

Edge	ymax	ymin	xmax	xmin	Slope
AB	y_2	y_1	x_2	x_1	m_1
AD	y_4	y_1	x_4	x_1	m_2
CD	y_3	y_4	x_3	x_4	m_3
BC	y_2	y_3	x_2	x_3	m_4

- For edge AB which is formed by (x_2,y_2) and (x_1,y_1) y_2 is greater than y_1.
- As we are filling polygon line by line, at the same time we are not considering the edges. Out of those attribute we are selecting ymax and then we are sorting the ymax array. After sorting ymax array we will get the table.

Edge	ymax	ymin	xmax	xmin	Slope
AB	y_2	y_1	x_2	x_1	m_1
BC	y_2	y_3	x_2	x_3	m_4
CD	y_3	y_4	x_3	x_4	m_3
AD	y_4	y_1	x_4	x_1	m_2

- Now the array is sorted hence we can find the intercept of line of the first two edges for sorted attribute table which is AB and BC.
- Here important is to find out the intersection point of scan line with edge.



20WPE0Y33   3/01/22   2109000φ

CS   CSC305

Date __/__/__

slope (m) = $\frac{\Delta y}{\Delta x}$ = $\frac{y_{new} - y_{old}}{x_{new} - x_{old}}$

$m = \frac{1}{x_{new} - x_{old}}$

$= \frac{1}{m}$

$x_{new} = x_{old} + \frac{1}{m}$

3/01/22 21C30001

2020PE0493

CA RV CSC305

Date __/__/____

Q.4A) it is an active transformation used to convert a single row into multiple rows and vice-versa.

- It is a smart way of representing our data in more organized way/manner.
- In a single row there is repeating data in multiple row it can be split into multiple rows.
- Sometimes we have data in multiple occuring columns.

2020PE0473 3/01/22

21CA5001

Date __/__/____

CG Q:CSC205

~~4)3) Parallel projection: rule to display picture~~

(2)(1) Drawbacks of boundary fill algorithm

- Limitation of boundary fill algorithm is that it fills polygon having unique boundary color.
- If the polygon having different boundary color then this algorithm fails.
- It does not work for large polygon

(2) How are they overcomed by flood fill?

- It can process image containing more than one boundary
- Flood fill algorithm are simplest and efficient

(3) Flood fill algorithm using 8 connected approach

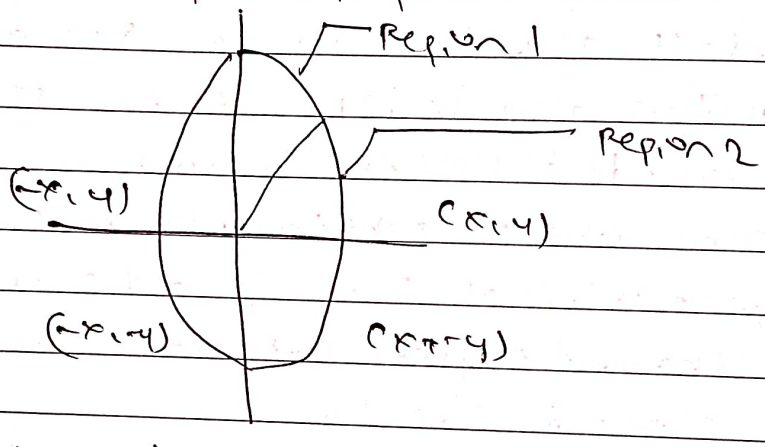
- The algorithm also begins with the seed ~~pixel~~ point which must be inside the polygon.
- Algorithm checks whether the pixel is having polygons original color or old color
- If yes change the pixel with new color and if no then return to its color
- When we want to fill the area that is not defined within a single boundary color. we can replaced them ~~by~~ with specified interior colors instead of searching for the boundary color
- Flood fill algorithm is ~~most~~ particularly useful if there is ~~not~~ no uniform color boundaries.

```
flood_fill (x,y,old_col,new_col)
{
    if (getpixel (x,y) = old_col)
    {
        putpixel(x,y,new_col);
        floodfill ( x+1,y,old_col,new_col);
        floodfill ( x-1,y,old_col,new_col);
```

Page No.

```
floodfill ( x, y+1, old_col, new_col);
floodfill ( x, y-1, old_col, new_col) ;
flood_fill ( x+1, y+1, old_col, new_col);
flood_fill ( x-1, y-1, old_col, new_col);
flood_fill ( x+1, y-1, old_col, new_col);
flood_fill ( x-1, y+1, old_col, new_col);
    }
}
```

Q.4 (i) Ans:- midpoint ellipse algorithm is used to draw an ellipse in computer graphics.



- This algorithm finds points of an ellipse on the first quadrant by dividing the quadrant into two regions
- Each point is projected into three quadrants.
- It uses four way symmetry.
- Algorithm for midpoint - ellipse algorithm

(i) Read radii rx and ry:-

(ii) Initialize starting point x=0 y=ry

(iii) $d_1 = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2$ (To calculate initial value)

(iv) $dx = 2r_y^2 x$

$dy = 2r_x^2 y$

→

Date ___/___/_____

Saathi

(v) do {plot (x,y)
if (d1 < 0)
{ x = x+1
y = y
dx = dx + 2r²y
d1 = d1 + dx + r²y
[d1 = d1 + 2r²x + 2r²y + r²y]
}
else
{
} x = x+1
y = y-1
dx = dx + 2r²x
dy = dy - 2r²x
d1 = d1 + dx - dy + r²y
[d1 = d1 + 2r²y x + 2r²y - (2r²xy - 2r²x) + r²y]
} while (dx < dy)

vi) $d_2 = r^2y\left[x+\frac{1}{2}\right]^2 + r^2x\left[y-1\right]^2 - r^2r^2$ [... no common initial value 2]

vii) do
{
Plot (x,y)
if (d2 > 0)
{ x = x
y = y-1
dy = dy - 2r²x
d2 = d2 - dy + r²x
[d2 = d2 - (2r²xy - 2r²x) + r²x]
}
else
{ x = x+1
y = y-1
dy = dy - 2r²x

Page No.

Saath

Date __/__/____

$d_2 = d_1 + 2r_y^2 x + 2y r^2 y - (2r^2 xy - 2r^2 x) + r^2 y]$

}

while ($y > 0$)

(iii) Determine the symmetrical points in other three quadrants

(iv) Stop