Q.1 Explain basic of Computer graphics with their terminology.

Ans – Computer Graphics is an art of drawing pictures on a computer screens with the help of programming. It involves computations, creations & manipulation of data.
CG is a rendering tool for generation & manipulation of images.

__Basic Terminology:-__

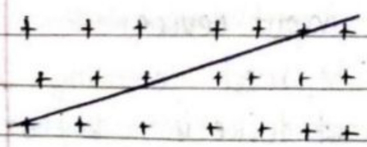
(i) Pixel (ii) Frame Buffe
(iii) Resolution (iv) Image Re
(v) Screen Resolution (vi)
(vii) Frame Aspect ratio
(viii) Pixel Aspect ratio

(i) __Pixel:-__ It is smallest unit of picture displayed on CS.

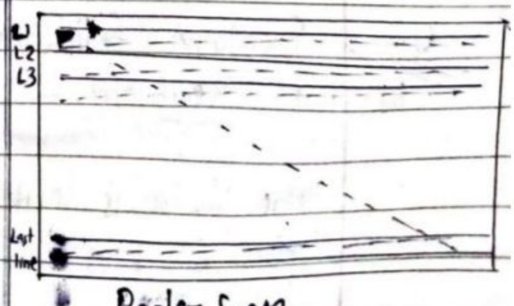
(ii) __Frame Buffer:-__ Before an image can be displayed on screen, it must be created by a computer program in a special part of computer Memory called __frame buffer.__

Q.4 (i) Difference b/w Random & Raster Scan

	Random Scan	Raster Scan
1.	It has high Resolution	low resoution
2.	It is more expensive	less expensive.
3.	Any modification is easy.	Modification is tough
4.	Solid pattern is tough to fill	Solid pattern is easy to fill
5.	Refresh rate depend on Resolution	Does not depend on picture
6.	Only screen with view on an area is displayed.	Whole screen is scanned.
7.	It does not use interlacing.	It uses interlacing method.
8.	Beam Peneration tech. comes under it.	Shadow mask technology comes under this
9.	It is restricted to line drawing application.	It is suitable for realistic display.

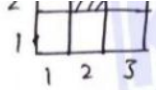


Random Scan



Raster Scan

(ii)	Beam Peneration	Shadow Mask
(i)	It is used with Random Scan system. to display color.	It is used with Raster Scan system.
(ii)	It can display only 4 color RGOY.	It can display millions of colors.
(iii)	less expensive	More Expensive
(iv)	High Resolution	Low Resolution
(v)	Less colors becaus BP depends on speed of electron beam.	Millions colors because SM depends on types of rays.
(vi)	Quality of picture	SM gives realism in



(1,2) is address of pixel.

Algorithm for line Drawing :- / (Scan Conversion Algorithm)

→ Direct use of line equation

→ DDA (Digital Differential Analyzer)

→ Bresenham's Algorithm

Direct use of line equation :-

$P_1 \rightarrow (x_1, y_1)$ $\qquad$ $P_2 \rightarrow (x_2, y_2)$

$\boxed{y = mx + b}$ $\qquad$ $m = \dfrac{y_2 - y_1}{x_2 - x_1}$ $\qquad$ $dy = y_2 - y_1$

$m = \dfrac{dy}{dx}$ $\qquad$ $dx = x_2 - x_1$

Step-1 Start Algo

Step-2 declare $x_1\ x_2\ y_1\ y_2\ dx\ dy\ m\ b$

Step-3 Enter values of x_1, x_2, y_1, y_2

Step-4 $dx = x_2 - x_1$

Step-5 $dy = y_2 - y_1$

Step-6 $m = dy/dx$

Step-7 $b = y_1 - m * x_1$

Step-8 ~~Step~~ set (x, y) and x_{end}

if $dx < 0$

then $x = x_2$

$y = y_2$, $x_{end} = x_1$

if $dx > 0$

then $x = x_1$

$y = y_1$

$x_{end} = x_2$

Step-9 Check whether the complete line has been drawn if $x = x_{end}$, stop

Step-10 Plot a point at (x, y)

Step-11 Increment, $x = x + 1$

Step-12 Complete next value of y from $y = mx + b$

Step-13 Go to step (9)

Eg- $P_1\ (0,0)$ $\qquad$ $P_2\ (6,18)$

$x_1 = 0,\ y_1 = 0,\ x_2 = 6,\ y_2 = 18$

$dx = 6 - 0 = 6$

$dy = 18 - 0 = 18$

$m = \dfrac{18}{6} = 3$

$b = 0 - 3 * 0 = 0$

$dx > 0$ $\quad$ $x = 0$, $y = 0$ $\quad$ $x_{end} = 6$

$x \neq x_{end}$

$(0,0)$

$x = 0 + 1 = 1$

$y = 3 \times 1 + 0 = 3$

$(1,3)$

$x \neq x_{end}$

$x = 1 + 1 = 2$

$y = 3 \times 2 + 0 = 6$ $\quad$ $(2,6)$

→ $x \neq x_{end}$

$x = 3$ $\quad$ $y = 9$ $\quad$ $(3,9)$

$(4,12)$ $\quad$ $(5,15)$

$x = 6$

$y = 18$ $\quad$ $x_{end} = x$ Stop

Bresenham's Line Algorithm :- It is used for scan converting a line.

→ It involves only integer addition, subtraction & multiplication operations

→ Next pixel selected is that one who has the least distance from true line

Step ① Input 2 endpoints & store left end point in (x_0, y_0)

Step ② Load $(x_0\ y_0)$ into frame buffer & it's first point.

Step ③ calculate $\Delta x, \Delta y, 2\Delta y$ & $2\Delta y - 2\Delta x$ and obtain $p_0 = 2\Delta y - \Delta x$

Step ④ At each x_K, starting at $k = 0$

if $p_K < 0$ then $(x_K + 1, y_K)$

and $p_{K+1} = p_K + 2\Delta y$

else $(x_K + 1, y_K + 1)$

and $p_{K+1} = p_K + 2\Delta y - 2\Delta x$

Step ⑤ Repeat step ④ Δx times

$P_1(5, 5)$ $P_2(13, 9)$ — $x_0 = 5$, $y_0 = 5$

$dx = 13 - 5 = 8$, $dy = 9 - 5 = 4$

$p_0 = 2 \times 4 - 8 = 0$

At (5, 5) :-

$P_1 = 0 + 2 \times 4 - 2 \times 8 = -8$ ✓

(6, 6)

At (6, 6) :-

$P_K < 0$ (7, 6)

$P = -8 + 2 \times 4 = 0$

At (7, 6)

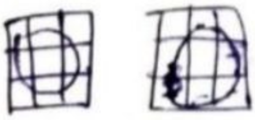
(8, 7)

$P = -8$

Plot	x	y	P
	5	5	0
(5,5)	6	6	−8
(6,6)	7	6	0
(7,6)	8	7	−8
(8,7)	9	7	0
(9,7)	10	8	−8
(10,8)	11	8	0
(11,8)	12	9	−8
(12,9)	13	9	0
~~(13,9)~~ ✓	~~14~~	~~10~~	~~−8~~

Circle Generating Algorithms:-

center (x_c, y_c)



$$(x-x_c)^2 + (y-y_c)^2 = r^2$$

→ Polar coordinates (r & θ) are used to eliminate unequal space.

$$x = x_c + r\cos\theta$$
$$y = y_c + r\sin\theta$$

Mid Point circle Algorithm:-

A method for direct distance comparison is to test the halfway position b/w 2 pixels to determine if this midpoint is inside or outside the circle boundary.

$$\boxed{f_{cir}(x,y) = x^2+y^2-r^2}$$

→ $f<0$, point is inside

→ $f>0$, point is outside.

→ $f=0$, point is on boundary.

Step-1 r, (x_c, y_c) & $(x_0, y_0) = (0, r)$

Step-2 Calculate $p_0 = 1-r$

Step-3 if $p_k < 0$ then

(x_k+1, y_k) &

$$p_{k+1} = p_k + 2x_{k+1} + 1$$

else

(x_k+1, y_k-1) &

$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$$

where

$$2x_{k+1} = 2x_k + 2$$
$$2y_{k+1} = 2y_k - 2$$

At(7,7) R ③ x 7,4.

eg- $r = 10$

$(x_0, y_0) =$

$p_0 = 1 - 10 = -9$

At (0,10):- (1,10)

$P_1 = -9 + 2 + 1 = -6$

At (1,10):- (2,10)

$P_2 = -6 + 4 + 1 = -1$

At (2,10):- (3,10)

$P_3 = -1 + 6 + 1 = 6$

At (3,10):- (4,9)

$P_4 = 6 + 8 + 1 - 18 = -3$

At (4,9):- (5,9)

$P_5 = -3 + 10 + 1 = 8$

At (6,8):-

Mid Point circle Algorithm :-

A method for direct distance comparison is to test the halfway position b/w 2 pixels to determine if this midpoint is inside or outside the circle boundary.

$$f_{cir}(x,y) = x^2 + y^2 - r^2$$

- $f < 0$, point is inside
- $f > 0$, point is outside.
- $f = 0$, point is on boundary.

Step-1 r, (x_c, y_c) & $(x_0, y_0) = (0, r)$

Step-2 Calculate $p_0 = 1 - r$

Step-3 if $p_k < 0$ then

$(x_k + 1, y_k)$ &

$p_{k+1} = p_k + 2x_{k+1} + 1$

else

$(x_k + 1, y_k - 1)$ &

$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$

where

$2x_{k+1} = 2x_k + 2$

$2y_{k+1} = 2y_k - 2$

At (7,7) R (3) $x \geq y$

eg- $r = 10$

$(x_0, y_0) = (0, 10)$

$p_0 = 1 - 10 = -9$

At (0,10):- (1,10)

$P_1 = -9 + 2 + 1 = -6$

At (1,10):- (2,10)

$P_2 = -6 + 4 + 1 = -1$

At (2,10):- (3,10)

$P_3 = -1 + 6 + 1 = 6$

At (3,10):- (4,9)

$P_4 = 6 + 8 + 1 - 18 = -3$

At (4,9):- (5,9)

$P_5 = -3 + 10 + 1 = 8$

At (6,8):- (7,7)

$P_6 = 8 + 14 + 1 - 14 = 9$

Step-4 Determine symmetry points on other seven octants.

Step-5 Move each calculated pixel position (x,y) onto (x_c, y_c) &

$x = x + x_c$

$y = y + y_c$

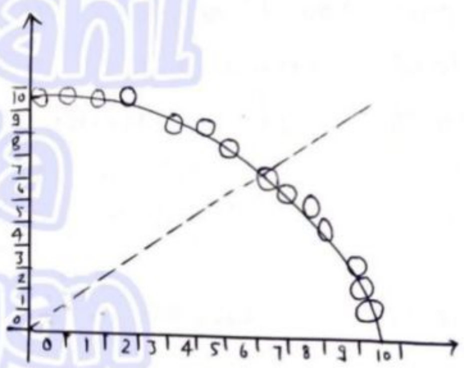
Step-6 Repeat step-3 until $x \geq y$

Step-7 For all points add the center point (x_c, y_c)

K	P_K	(x_{k+1}, y_{k+1})	$2x_{k+1}$	$2y_{k+1}$
0	-9	(1,10)	2	20
1	-6	(2,10)	4	20
2	-1	(3,10)	6	20
3	6	(4,9)	8	18
4	-3	(5,9)	10	18
5	8	(6,8)	12	16
6	9	(7,7)	14	14

Octant-1

Octant-1	Octant-2
(0,10)	(8,6)
(1,10)	(9,5)
(2,10)	(9,4)
(3,10)	(10,3)
(4,9)	(10,2)
(5,9)	(10,1)
(6,8)	(10,0)



Aliasing :- When a line in a digital image runs at an angle, then it will ...

Cohen Sutherland line Clipping Algorithm:-

It is detected whether line lies inside the screen or it is outside the screen. All lines come under visibe, not visible or clipping case.

- 2 phases Algo

(i) Identification Phase :- World space is divided into regions based on window boundary

Top Btm R L

1001	1000	1010	Y_{max}
0001	0000	0010	Y_{min}
0101	0100	0110	
	X_{min}	X_{max}	

$x < X_{min}$ → left of window

$x > X_{max}$ → right of window

$Y < Y_{min}$ → Bottom of window

$Y > Y_{max}$ → Top of window

Algo:-

1. Assign Region code to both end points. P_0 & P_1 AND
2. If P_0 & P_1 == 0000 then assigned ✓

else

if line crossed X_{min} or X_{max} then

$y = y_1 + m(x - x_1)$

else

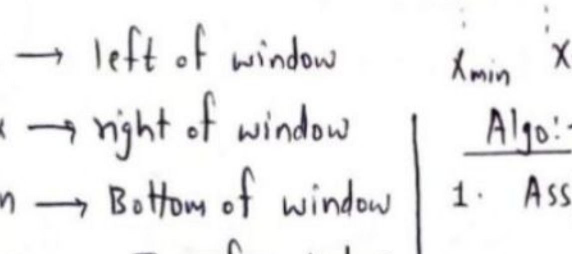
$x = x_1 + \frac{1}{m}(y - y_1)$

3. Verify $X_{min} \le X \le X_{max}$

$Y_{min} \le Y \le Y_{max}$ ✓

Q. ** A(20,20) B(90,20) C(90,70) D(20,70)

find region code for line P_1(10,30) & P_2(80,90) & clip the line P_1P_2



$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{90-30}{80-10}$

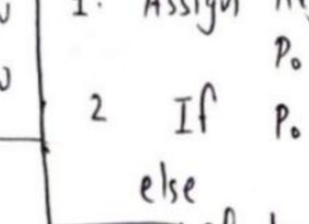
$= \frac{60}{70}$

$= 0.86$

P_1 = 0001

P_2 = 1000 &

0000 ✓ assigned



$|A| = 1$

0

$i_1(X_{min}, Y)$

$Y = Y_1 + m(X_{min} - X_1)$

$Y = 30 + 0.86(20-10)$

$Y = 38.57$

$X = X_1 + \frac{1}{m}(Y - Y_1)$

$X = 80 + \frac{1}{0.86}(70 - 90)$

$X = 56.67$

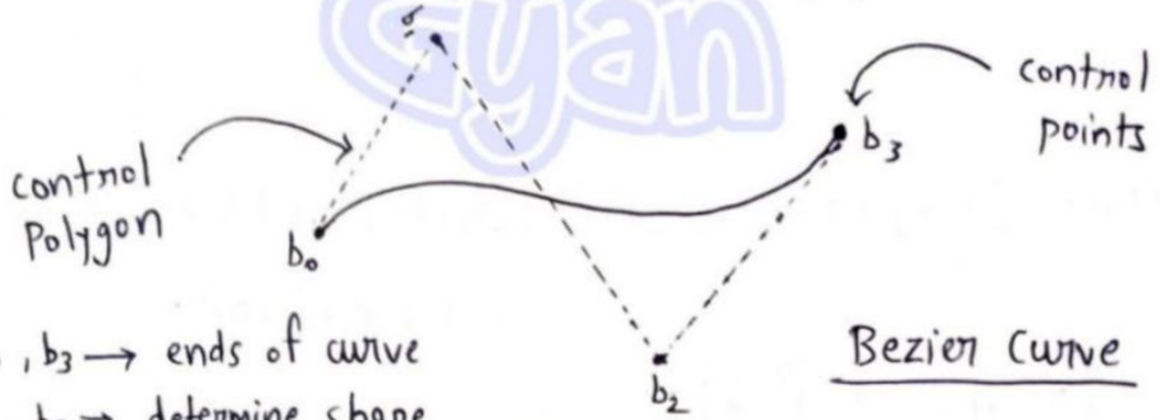
★ Bezier Curve :-

It is a parametric curve defined by a set of control points. The no of control points to be approximated and their relative position determine the degree of Bezier ~~curve~~ polynomial. $D = 4 - 1 = 3$

→ Two points are ends of curve. Other points determine shape of curve. It was given by Pierre Bezier.

→ Bezier curve generally follows the shape of the defining polygon.

→ ★ No straight line intersects a bezier curve more times than it intersects it control polygon.



$b_0, b_3 \rightarrow$ ends of curve

$b_1, b_2 \rightarrow$ determine shape of curve

Bezier Curve

$$P(t) = \sum_{i=0}^{n} B_i J_{n,i}(t)$$

- $P(t)$ → Any point lying on bezier curve
- B_i → ith control point of bezier curve
- $J_{n,i}(t)$ → Blending f^n
- n → degree of curve

$$J_{n,i}(t) = C(n,i)\, t^i (1-t)^{n-i} \Rightarrow \frac{n!}{i!\,(n-i)!} t^i (1-t)^{n-i}$$

$$P(t) = \sum_{i=0}^{3} B_i J_{3,i}(t) = B_0 J_{3,0}(t) + B_1 J_{3,1}(t) + B_2 J_{3,2}(t) + B_3 J_{3,3}(t)$$

$$J_{3,0}(t) = \frac{3!}{0!\,3!} t^0 (1-t)^{3-0} = (1-t)^3$$

$$J_{3,1}(t) = \frac{3!}{1!\,2!} t^1 (1-t)^{3-1} = 3t(1-t)^2 \quad , \quad J_{3,3}(t) = \frac{3!}{3!\,0!} t^3 (1-t)^0 = t^3$$

$$J_{3,2}(t) = \frac{3!}{2!\,1!} t^2 (1-t)^{3-2} = 3t^2(1-t)$$


Purchase the Notes
100₹
Per semester
(All subjects)
Notes (Hand written) ✅
Most Questions ✅
All Branches
Min 100%
amount will go
into charity 🌟
For specific
Subject - 50₹
UPI ID -
sahilkagyan337@ybl

Scan & Pay Using PhonePe App

SAHIL KHAN