COMPUTER GRAPHICS

Date _____

It is an area of computer science & engineering which plays a very important role in almost every application of computer software & use of computer science.

Computer Graphics involves display, manipulation and storage of pictures & experimental data for proper visualization using a computer.

Typically Graphics system comprises of a host computer with support of fast processor, large m/m, frame buffer and few other component
Other components :- 1) Display devices (monitor)
2) Input Devices (mouse, keyboard, joystick)
3) Output Devices (LCD panels, laser printers)
4) Interfacing Devices (video I/O, TV interface etc)

Computer Graphics system could be <u>active</u> or <u>passive</u>.

In both cases, the input to the system is the scene description and output is a static or animated scenes to be displayed.

In case of active systems, the user controls the display with the help of a GUI, using an input device.
In passive systems, however there is no control, you have to watch whatever is shown to you.
eg: TV :- you can change the channel but cannot control what is being shown.
Video Games :- Active System.

Date

Various fundamental concepts and principles in Computer Graphics are:

1) Display System

Storage displays, random scan, raster scan, CRT basics, video basics flat panel displays.

2) Transformations

Affine (2D & 3D) : rotation, translation, scale, reflection and shear.
Viewing : The camera transformations perspective, orthographic, isometric and stereographic views

3) Scan conversion and Clipping

drawing of pts, lines, markers, curves circles, ellipse, polyline, polygon, area-fill, fill pattern, clipping algorithm, anti-aliasing

4) Hidden Surface Removal

back face culling, painter's algorithm, scan-line algo., BSP trees, z-buffer, ray tracing etc.

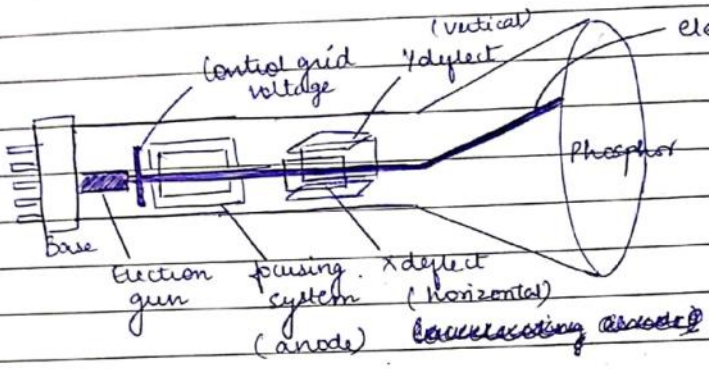
5) Shading & Illumination

Phongs shading model, texture mapping, bump mapping, Gouraud shading, shadows and background, color models etc.

Date _____

Cathode Ray Tube

The primary output device in a graphical system is the video monitor. The main element of a video monitor is the CRT.



It is an operation which shows how an e⁻ gun with an accelerating anode is used to generate an electron beam which is used to display point or pictures in the screen.

Heating filament is responsible to heat up the cathode element of CRT and that is what generates the e⁻ elections.

Control Grid : manages the intensity
When we observe a picture on the screen some part of picture may be bright or some part may be dark.
This brightness and darkness or illumination or the intensity on the screen is basically controlled by the intensity of the beam which strikes a particular point on the screen.
This intensity of the beam is controlled by controlling the intensity of the electron

beam which is coming out of cathode.

Election beam consists of e^- which are ~~negatively~~ -vely charged. The control grid is a cylindrical device which is also -vely charged. It is a high -ve voltage which is applied to the control grid. Now, if both e^- & control grid are -vely charged both of them will repel each other.

So the amount of voltage at the control grid will essentially allow a certain amount of e^- of the election beam to pass through it & the amt. of e^- beam or amt. of e^- which pass through cylindrical control grid will be controlled by -ve voltage in control grid.

⇒ reduce the amt. of voltage in control grid or that means -ve voltage is reduced you are allowing more e^- to pass through it & the intensity of the beam will be higher.

If ↑ voltage of control grid less intensity.

Unlike the control grid the other two parts ie. accelerating anode and focusing anode will have positive voltage.

Date _____

Essentially the focusing anode is responsible to focus the beam on to a particular point on the screen. (e^- are focused on a particular point on the screen).
Accelerating anode is necessary because we want the e^- to strike the screen at a very high speed to emit light. (provides acceleration)

Horizontal deflection and vertical deflection are needed because the screen is a square matrix and we want ~~our~~ to implement the deflection of the beam so that the beam not only go and hit the centre of the screen to create a point but we need this beam to move all around this image to create a picture.

Two techniques for producing images on the CRT screen.

1) Random Scan Display
2) Raster Scan Display

RASTER SCAN DISPLAY

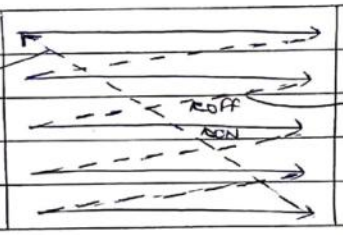
Electron beam is swept across the screen one row at a time from top to bottom. As it moves across each row, the beam intensity is turned on and off to create a pattern of illuminated spots.
The scanning process is called refreshing. Each complete scanning of a screen is normally called a frame.

Date

Refreshing rate, is called the frame rate
Picture definition is stored in a m/m area called frame buffer.
Frame buffer stores the intensity value for all the screen points (pixel) or picture element.

On black & white system, frame buffer storing the value of pixels is called bitmap.
1- on 0- off determines intensity of pixel.

On color system, frame buffer storing pixel value is called pixmap.



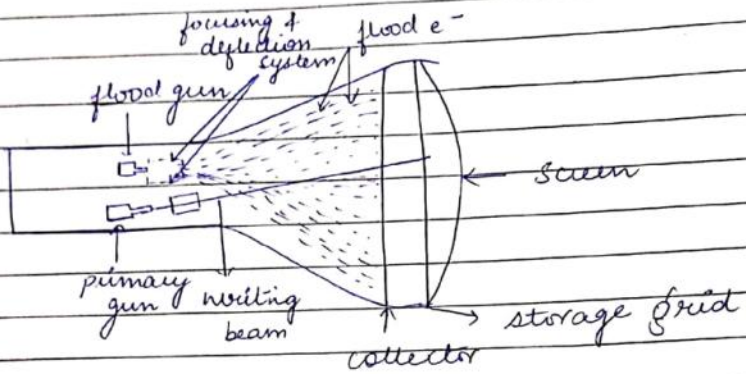
RANDOM SCAN DISPLAY

The CRT's e⁻ beam is directed only to the parts of the screen where a picture is to be drawn. The picture definition is stored as a set of line-drawing commands in a refresh display file or a refresh buffer in memory.

Date _____

Base of Difference	Raster Scan System	Random Scan System
Election Beam	1) The e⁻ beam is swept across the screen, one row at a time, from top to bottom	1) The e⁻ beam is directed only to the parts of screen where a picture is to be drawn.
Resolution	2) Poor because it produces zigzag lines that are plotted as discrete point sets.	2) Good because it produces smooth lines drawings because CRT beams directly follows the line path
Picture Definition	3) Stored as a set of intensity values for all screen pts, called pixels in a refresh buffer area.	3) Stored as a set of line drawing instructions in a display file
Realistic Display	4) The capability of this system to store intensity values for pixel makes it well suited for realistic display of scenes contain shadow & color pattern	4) System is designed for line-drawing & can't display realistic shaded scenes.
Draw an image	5) screen pts./pixels are used to draw an image.	5) mathematical functions are used to draw an image

DIRECT - VIEW STORAGE TUBES (DVST)



It is an alternative method for maintaining the screen image. It ~~does~~ uses the storage grid which stores the pictures information as a charge distribution just behind the phosphor coated screen. DVST consists of two e⁻ guns a primary gun and a flood gun.

A primary gun stores the picture pattern & the flood gun maintains the picture display. Primary gun emits high speed e⁻ which strikes on the storage grid to draw the picture pattern. As e⁻ beam strikes on the storage grid with high speed, it knocks out e⁻ from the storage grid keeping the net positive charge.

The knocked out e⁻ are attracted towards the collector. The net positive charge on the storage grid is nothing but the picture pattern. The continous low speed of e⁻ from flood gun pass through the control grid & are attracted to the +ve charged area of the storage grid.

Date _____

The low speed e^- then penetrate the storage grid & skrike the phosphor coating without affecting the the charge pattern on the storage grid.
During this process the collector just behind the storage grid smooth out the flow of flood e^-.

Advantage

1) No refreshing is needed
2) very complex pictures can be displayed at very high resolution without flicker.

Disadvantage

1) Do not display color.
2) Selected parts of picture cannot be erased.
3) To eliminate a picture selection, entire screen must be erased & modified picture is redrawn; which is time consuming for complex pictures.

Date

DIGITAL DIFFERENTIAL ANALYZER (DDA)

DDA line Drawing Algorithm.

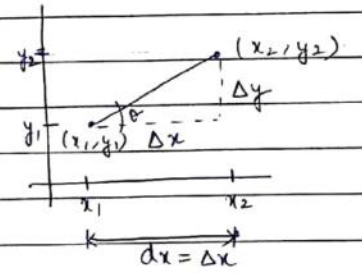
Line drawing is done by calculating intermediate positions along the line path between two specified endpoints positions.
The output device is then directed to fill in those positions between the end points with some color.

Line Equations

$$y = mx + c$$

↳ y intercept.

Slope of line



$$m = \frac{\Delta y}{\Delta x}$$

$$\tan\theta = m = \frac{\Delta y}{\Delta x}$$

$$\Delta y = m \Delta x$$

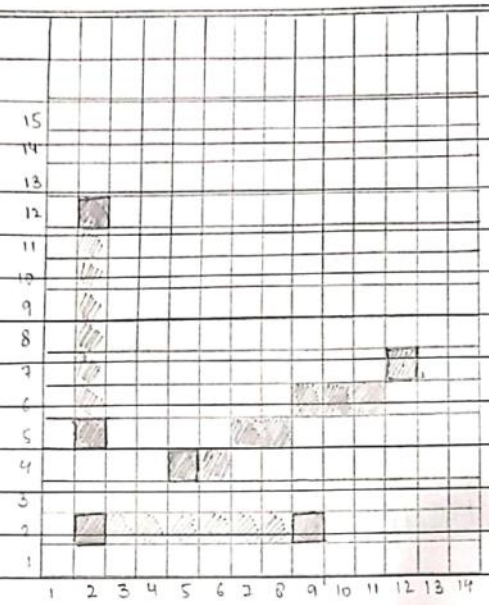
Now,

$|m|$ can be :

$\frac{\Delta y}{\Delta x} > 1$; $\frac{\Delta y}{\Delta x} < 1$ and $\frac{\Delta y}{\Delta x} = 1$

$\theta > 45°$ $\theta < 45°$ $\theta = 45°$

Date _____



(x_1, y_1) (x_2, y_2)

$(2, 2)$ $(9, 2)$

$\Delta x = 9 - 2 = 7$

$\Delta y = 2 - 2 = 0$

Steps = $\Delta x > \Delta y$

$\therefore \Delta x$

$m = \frac{\Delta y}{\Delta x} = \frac{0}{7} = 0$; Steps = 7

Steps : from starting pt. how many pts are needed to reach end pt.

$x_{inc} = \frac{7}{7} = 1$ $y_{inc} = \frac{0}{7} = 0$

$= \frac{\Delta x}{steps}$ $\frac{\Delta y}{steps} = 0$

x	2	3	4	5	6	7	8	9
y	2	2	2	2	2	2	2	2

Date

Eg: (5, 4) and (12, 7) $m<1$

$\Delta x = 12 - 5 = 7$

and $\Delta y = 7 - 4 = 3$

$\Delta m = \frac{\Delta y}{\Delta x} = \frac{3}{7}$

$\therefore$ Steps $= \Delta x = 7$

$x_{inc} = \frac{\Delta x}{steps} = \frac{7}{7} = 1$

$y_{inc} = \frac{\Delta y}{steps} = \frac{3}{7} = 0.4$

x	5	6	7	8	9	10	11	12
y	4	4.4	4.8	5.2	5.6	6	6.4	6.8

There is no decimal pixel

∴ round off the pixel value.

known as "Jaggies"

x	5	6	7	8	9	10	11	12
y	4	4	5	5	6	6	6	7

As a result some curved line is obtained

Here, $m < 1$

then

x_{K+1}

$x_{K+1} = x_K + 1$

and

y_{K+1}

$y_{K+1} = y_K + m$

Date

Eg: (12, 9) (17, 14) $m = 1$

$\Delta x = 17 - 12 = 5$

$\Delta y = 14 - 9 = 5$

Steps = 5

$x_{inc} = \frac{5}{5} = 1$ $y_{inc} = \frac{5}{5} = 1$

x	12	13	14	15	16	17
y	9	10	11	12	13	14

ALGORITHM

1. Get the input two end pts. (x_1, y_1) & (x_2, y_2)
2. Calculate the difference b/w the two pts.

$dx = \Delta x = x_2 - x_1$

$dy = \Delta y = y_2 - y_1$

3. Based on step2, Identify the no. of steps to putpixel.

If abs(dx) > abs(dy) (more steps are needed in x coordinate)

Steps = absolute (dx)

else Steps = absolute (dy)

Calculate the increment in x coordinate & y coord

$x_{inc} = dx / float(steps)$

$y_{inc} = dy / float(steps)$

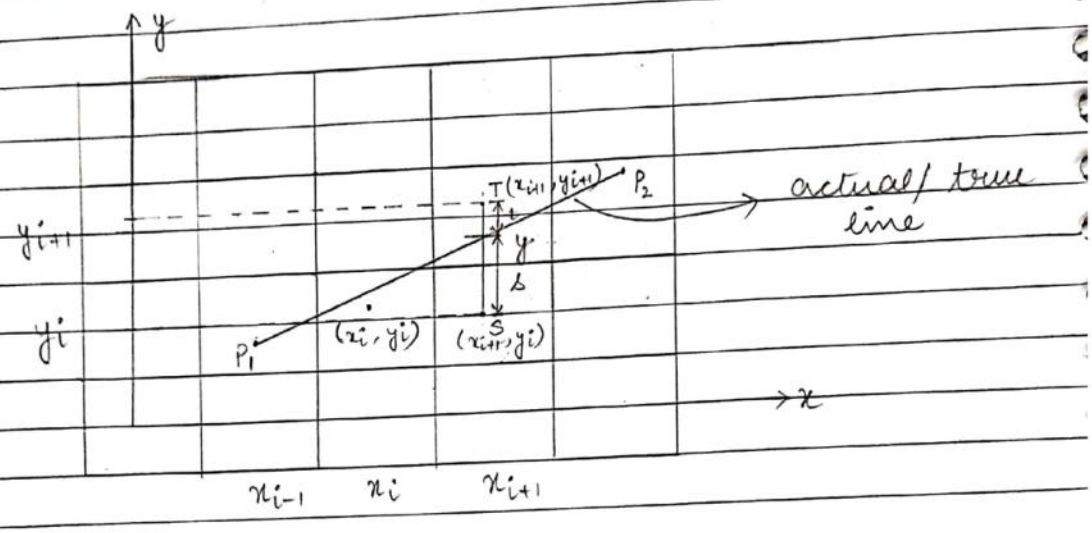
Date _____

5. Put the pixel by successfully incrementing x & y coordinates.

```
for ( i= 0 ; i < steps ; i++)
    {
        x=  x + x inc
        y =  y + y inc
        putpixel (x, y);
    }
```

Date

BRESENHAM'S LINE ALGORITHM



Last chosen pixel is (x_i, y_i)
Task is to chose the next one between the bottom pixel S and top pixel T.
If S is chosen coordinates will be (x_{i+1}, y_i) and if T is chosen coordinates will be (x_{i+1}, y_{i+1}).

$$x_{i+1} = x_i + 1 \quad ; \quad y_{i+1} = y_i + 1$$

$$s = y - y_i$$
$$t = y_{i+1} - y$$

Actual line, $y = mx + c$

$$= m(x_{i+1}) + c$$
$$= m(x_i + 1) + c$$
$$= mx_i + m + c \quad \text{— (1)}$$

Date _____

Now, difference b/w these two pixel values:

$$s - t = (y - y_i) - (y_{i+1} - y)$$

$$= y - y_i - (y_i + 1 - y)$$

$$= y - y_i - y_i - 1 + y$$

$$= 2y - 2y_i - 1 \quad \text{---} \quad (2)$$

Now, Put value of y from (1) in (2)

$$s - t = 2[mx_i + m + c] - 2y_i - 1$$

* using $m = \frac{\Delta y}{\Delta x}$

$$s - t = 2\left[\frac{\Delta y}{\Delta x} x_i + \frac{\Delta y}{\Delta x} + c\right] - (2y_i + 1)$$

$$s - t = 2\left[\frac{\Delta y x_i + \Delta y + c}{\Delta x}\right] - (2y_i + 1)\frac{\Delta x}{\Delta x}$$

$$(s - t)\Delta x = 2\Delta y x_i + 2\Delta y + 2c - 2y_i \Delta x - \Delta x$$

Assume $d_i = (s - t)\Delta x$ a decision parameter

$$d_i = 2\Delta y x_i + 2\Delta y + 2c - 2y_i \Delta x - \Delta x$$

$$= 2\Delta y x_i - 2y_i \Delta x + \underbrace{2\Delta y + 2c - \Delta x}_{c}$$

$$d_i = 2\Delta y x_i - 2y_i \Delta x + c \quad \text{— (3)}$$

For next step $i = i+1$

$$d_{i+1} = 2\Delta y x_{i+1} - 2y_{i+1} \Delta x + c \quad \text{— (4)}$$

Subtract (3) from (4)

$$d_{i+1} - d_i = 2\Delta y x_{i+1} - 2y_{i+1} \Delta x + \cancel{c}$$
$$- 2\Delta y x_i + 2y_i \Delta x - \cancel{c}$$

$$d_{i+1} - d_i = 2\Delta y (x_{i+1} - x_i) - 2\Delta x (y_{i+1} - y_i)$$

Now, Since $x_{i+1} = x_i + 1$

$$d_{i+1} - d_i = 2\Delta y (x_i + 1 - x_i) - 2\Delta x (y_{i+1} - y_i)$$

$$d_{i+1} - d_i = 2\Delta y - 2\Delta x (y_{i+1} - y_i)$$

$$d_{i+1} = d_i + 2\Delta y - 2\Delta x (y_{i+1} - y_i)$$

if the chosen pixel is S

$$\Rightarrow d_i < 0$$

then, $y_{i+1} = y_i$

$$\therefore \quad d_{i+1} = d_i + 2\Delta y - 2\Delta x (y_i - y_i)$$

Date _____

$$\boxed{d_{i+1} = d_i + 2\Delta y}$$

If the chosen pixel is T $\Rightarrow d_i \geq 0$

then, $y_{i+1} = y_i + 1$

$$d_{i+1} = d_i + 2\Delta y - 2\Delta x(y_{i+1} - y_i)$$

$$\boxed{d_{i+1} = d_i + 2\Delta y - 2\Delta x}$$

$$d_{i+1} = \begin{cases} d_i + 2(\Delta y - \Delta x) & \text{if } d_i \geq 0 \\ d_i + 2\Delta y & \text{if } d_i < 0 \end{cases}$$

Now, for d_i

$$d_i = (s-t)\Delta x$$

$$= (2y - 2y_i - 1)\Delta x \quad (\text{from } ①)$$

$$= [2(mx_i + c) - 2y_i - 1]\Delta x \quad [\text{Eq. of line } y]$$

$$= [2(mx_i + m + c) - 2y_i - 1]\Delta x$$

as $(x_{i+1} = x_i + 1)$

$$= [2mx_i + 2m + 2c - 2y_i - 1]\Delta x$$

$$= [2(mx_i + c - y_i) + 2m - 1]\Delta x$$

Now, $y = mx + c \Rightarrow mx + c - y = 0$

Date _____

$$= (2x0 + 2m - 1)\,\Delta x$$

$$= (2m - 1)\,\Delta x$$

$$= \left(2\frac{\Delta y}{\Delta x} - 1\right)\Delta x$$

$$\boxed{d_i = 2\Delta y - \Delta x}$$

$$d_{i+1} = \begin{cases} d_i + dT & d_i \geq 0 \\ d_i + dS & d_i < 0 \end{cases}$$

$$d = \begin{cases} d + dT & d \geq 0 \\ d + dS & d < 0 \end{cases}$$

ALGORITHM

Bresenham's algorithm for scan-converting a line from $P_1(x_1, y_1)$ to $P_2(x_2, y_2)$ with $x_1 < x_2$ and $0 < m < 1$

```
int x = x1 , y = y1
int dx = x2 - x1 and dy = y2 - y1

int dT = 2( dy - dx) , dS = 2dy ;
int d = 2dy - dx
setPixel (x,y);
```

Date

```
while ( x < x2 )
  { x++ ;
    if ( d < 0 )
        d = d + dS ;
    else
      { y++ ;
        d = d + dT ;
      }
    setpixel (x, y) ;
  }
```

Example : (1, 1) and (8, 5)

$\Delta x = 7$; $\Delta y = 4$

$d_i = 2\Delta y - \Delta x$

$= 8 - 7 = 1 \geq 0$

∴ Select T and x++ and y++

$d_{i+1} = d_i + 2\Delta y - 2\Delta x$

$= 1 + 8 - 14 = -5 < 0$

∴ Select S

y++ x++

$d < 0$

∴ $d = d + dS$

$= -5 + 2\Delta y$

$= -5 + 2(4) = -5 + 8 = +3 \geq 0$

∴ Select T

x++ and y++

di	x	y
	1	1
d=1	2	2
d=-5	3	2
d=3	4	3
d=-3	5	3
d=+5	6	4
d=-1	7	4
d=7	8	5

Date

Example : (1,1) and (8,5)

$\Delta x = 7 \qquad \Delta y = 4$

$d = 2\Delta y - \Delta x$

$= 8 - 7 = 1 \geq 0$

$\therefore \quad x{+}{+} \text{ and } y{+}{+}$

and

new $d = d + dT$

$= 1 + 2\Delta y - 2\Delta x$

$= 1 + 2(4) - 14$

$= 1 + 8 - 14$

$= -5 < 0$

$\therefore \quad x{+}{+}$

and

new $d = d + dS$

$= -5 + 2\Delta y$

$= -5 + 2(4) = -5 + 8$

$= 3 \geq 0$

$\therefore \quad x{+}{+} \text{ and } y{+}{+}$

and new $d = d + dT$

$= 3 + 2\Delta y - 2\Delta x$

$= 3 + 8 - 14 = -3 < 0$

$x{+}{+}$

new $d = d + dS$

$= -3 + 2\Delta y = -3 + 8 = 5 \geq 0$

$x{+}{+} \text{ and } y{+}{+}$

x	y
1	1
2	2
3	2
4	3
5	3
6	4
7	4
8	5

d = 1, d = -5, d = 3, d = -3, d = 5, d = -1, d = 7

Date

new $d = d + dT$

$= 5 + 2(4) - 2(7) = 5 + 8 - 14 = -1$

$\therefore$ $x++$

new $d = d + dS$

$= -1 + 2\Delta y = -1 + 2(4) = -1 + 8 = 7$

$x++$ and $y++$

SCAN CONVERTING A CIRCLE

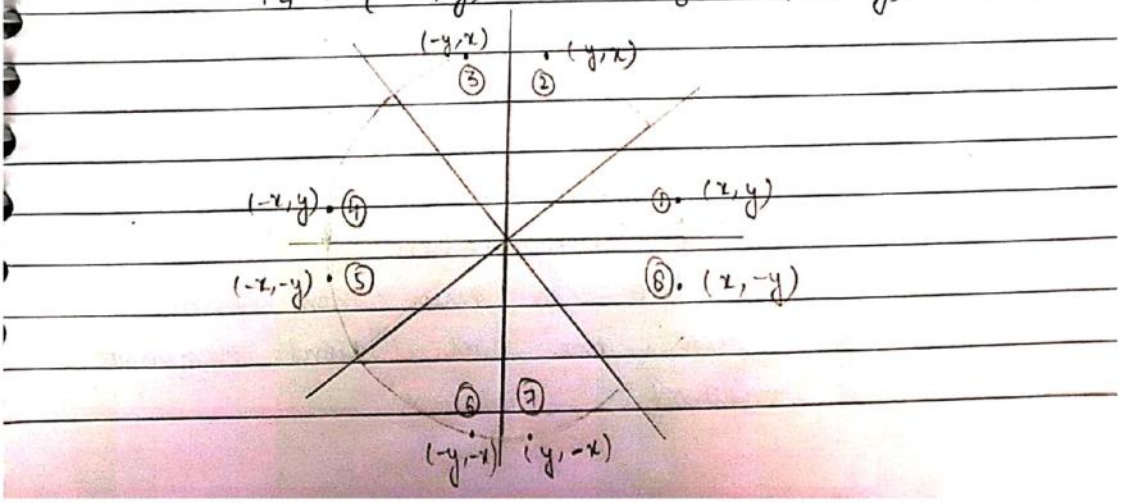
A circle is a symmetric figure. Any circle-generating algorithm can take advantage of the circle's symmetry to plot eight points for each value that the algorithm calculates. Eight-way symmetry is used by reflecting each calculated point around each 45° axis.

$P_1 = (x, y)$ $\quad$ $P_5 = (-x, -y)$

$P_2 = (y, x)$ $\quad$ $P_6 = (-y, -x)$

$P_3 = (-y, x)$ $\quad$ $P_7 = (y, -x)$

$P_4 = (-x, y)$ $\quad$ $P_8 = (x, -y)$



Defining a Circle

1. Second-order polynomial equation

$$x^2 + y^2 = r^2$$

$$y^2 = r^2 - x^2$$

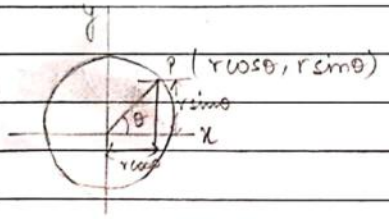
each x coordinate in the sector, from 90° to 45° is found by stepping x from 0 to $r/\sqrt{2}$ and y coordinate is found by evaluating $\sqrt{r^2 - x^2}$ for each step of x.

This method is inefficient, because for each point both x and r must be squared and subtracted from each other; then square root of the result must be found.

Point will $P(x, \sqrt{r^2 - x^2})$

2. Trigonometric function

$$x = r\cos\theta \quad ; \quad y = r\sin\theta$$



θ is stepped from 0 to $\pi/4$

This method is not feasible because computation of the values of $\sin\theta$ and $\cos\theta$ is even more time-consuming than the calculations required by first method.

Date _____

BRESENHAM'S CIRCLE ALGORITHM

If the eight-way symmetry of a circle is used to generate a circle, points will only have to be generated through a 45° angle.

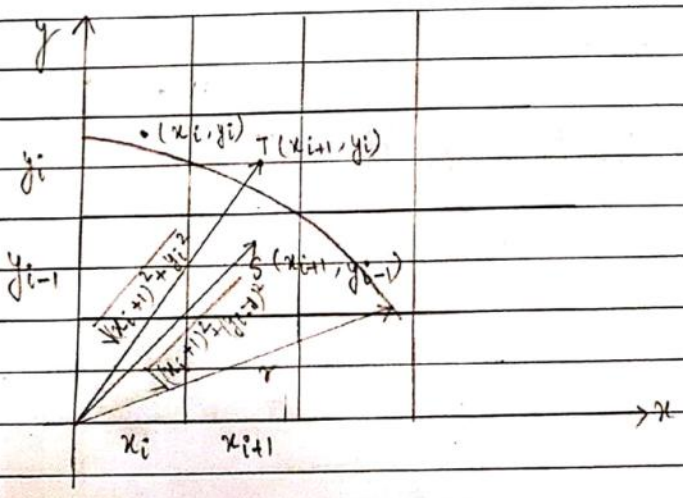
Points are generated from 90° to 45°, moves will be made only in the +x and -y directions.

If points are generated from 90° & 45°, each new point closest to the true circle can be found by taking either of two actions

i) move in the x direction one unit

ii) move in the x direction one unit and move in the negative y direction one unit.

∴ A method for selecting b/w these two points is necessary to find the points closest to the true circle.



(x_i, y_i) coordinates of the last scan-converted pixel upon entering step i.

$T(x_{i+1}, y_i)$ and $S(x_{i+1}, y_{i-1})$

$x_{i+1} = x_i + 1$ and

$y_{i-1} = y_i - 1$

$D(T)$ = distance from origin to T pixel squared minus distance to the true circle squared

$= (x_{i+1})^2 + y_i^2 - r^2$

$= (x_i + 1)^2 + y_i^2 - r^2$

$D(S)$ = distance from origin to S pixel squared minus distance to the true circle squared.

$= (x_{i+1})^2 + (y_{i-1})^2 - r^2$

$= (x_i + 1)^2 + (y_i - 1)^2 - r^2$

$r^2 - [(x_i + 1)^2 + (y_i - 1)^2]$

Date _____

d_i = decision parameter

$d_i = D(T) + D(S)$ / $D(T) - D(S)$

$d_i = 2(x_i+1)^2 + y_i^2 + (y_i-1)^2 - 2r^2$

for next step

$d_{i+1} = 2(x_{i+1}+1)^2 + y_{i+1}^2 + (y_{i+1}-1)^2 - 2r^2$

Now,

$d_{i+1} - d_i = 2(x_{i+1}+1)^2 + y_{i+1}^2 + (y_{i+1}-1)^2 - 2r^2$
$- 2(x_i+1)^2 - y_i^2 - (y_i-1)^2 + 2r^2$

Since $x_{i+1} = x_i + 1$

$\therefore d_{i+1} - d_i = 2(x_i+1+1)^2 + y_{i+1}^2 + (y_{i+1}-1)^2$
$- 2(x_i+1)^2 - y_i^2 - (y_i-1)^2$

$d_{i+1} - d_i = 2(x_i+2)^2 + y_{i+1}^2 + (y_{i+1}-1)^2 - 2(x_i+1)^2$
$- y_i^2 - (y_i-1)^2$

$= 2[x_i^2 + 4x_i + 4] + y_{i+1}^2 + y_{i+1}^2 + 1 - 2y_{i+1}$
$- 2[x_i^2 + 1 + 2x_i] - y_i^2 - [y_i^2 + 1 - 2y_i]$

Date

$$= 2x_i + 8 + 8x_i + y_{i+1}^2 + y_{i+1}^2 + 1 - 2y_{i+1}$$

$$- 2x_i - 2 - 4x_i - y_i^2 - y_i^2 - 1 + 2y_i$$

$$= 4x_i + 2(y_{i+1}^2 - y_i^2) - 2(y_{i+1} - y_i) + 6$$

$$d_{i+1} = d_i + 4x_i + 2(y_{i+1}^2 - y_i^2) - 2(y_{i+1} - y_i) + 6$$

When $d_i < 0$

$\Rightarrow$ $|D(T)| < |D(S)|$ and T pixel is chosen

$\therefore$ $y_{i+1} = y_i$ and so.

$$d_{i+1} = d_i + 4x_i + 2(y_i^2 - y_i^2) - 2(y_i - y_i) + 6$$

$$d_{i+1} = d_i + 4x_i + 6$$

When $d_i \geq 0$

$\Rightarrow$ $|D(T)| \geq |D(S)|$ and S pixel is chosen

$\therefore$ $y_{i+1} = y_i - 1$

$$d_{i+1} = d_i + 4x_i + 2((y_i - 1)^2 - y_i^2) -$$

$$2(y_i - 1 - y_i) + 6$$

$$= d_i + 4x_i + 2[y_i^2 + 1 - 2y_i - y_i^2] - 2(-1) + 6$$

$$d_{i+1} = d_i + 4x_i + 2 - 4y_i + 2 + 6$$

$$d_{i+1} = d_i + 4(x_i - y_i) + 10$$

$$d_{i+1} = \begin{cases} d_i + 4x_i + 6 & \text{if } d_i < 0 \\ d_i + 4(x_i - y_i) + 10 & \text{if } d_i \geq 0 \end{cases}$$

Set $(0, r)$ to be starting pixel coordinates & compute the base case value d_1

$$d_1 = 2(0+1)^2 + r^2 + (r-1)^2 - 2r^2$$

$$= 2 + r^2 + r^2 + 1 - 2r - 2r^2$$

$$= 3 - 2r$$

Algorithm for pixel coordinates 90° to 45° octant

```
int x=0, y=r, d=3-2r
while (x<=y)
  setPixel (x,y);
  if (d<0)
    d= d+4x+6
  else {  d= d+4(x-y)+10;
          y--;
  }
```

Date _____

x++
}

Ques: initial point (0, 7)

MIDPOINT CIRCLE ALGORITHM

$$f(x,y) = x^2 + y^2 - r^2 \begin{cases} < 0 & (x,y) \text{ inside the circle} \\ = 0 & (x,y) \text{ on the circle} \\ > 0 & (x,y) \text{ outside the circle} \end{cases}$$

Consider the coordinates of the point halfway between pixel T and pixel S

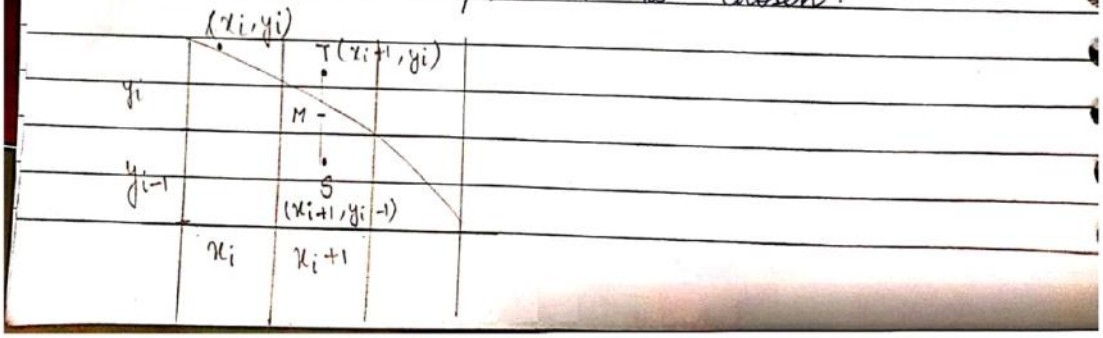
$$M\left(x_i+1, y_i-\frac{1}{2}\right)$$

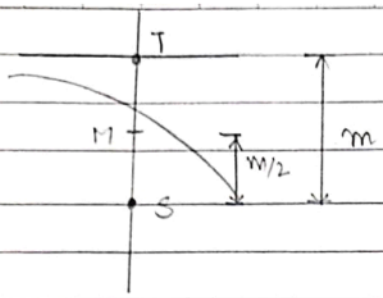
$$T(x_i+1, y_i) \qquad S(x_i+1, y_i-1)$$

$$x^2 + y^2 - r^2 < 0$$

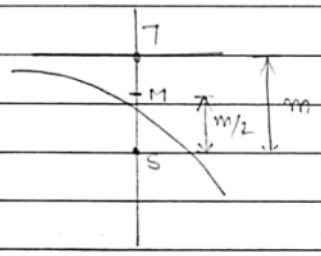
$$\Rightarrow x^2 + y^2 < r^2$$

∴ pixel T is chosen.





S pixel is more than $m/2$ distance away from circle boundary &
T pixel is less than $m/2$ distance away from circle boundary
∴ T pixel is chosen.



here S pixel is chosen

d_i = decision parameter

$d_i = f(x_{i+1}, y_i - 1/2)$

$d_i = (x_i + 1)^2 + (y_i - \frac{1}{2})^2 - r^2$

Now, $d_{i+1} = (x_{i+1} + 1)^2 + (y_{i+1} - \frac{1}{2})^2 - r^2$

$d_{i+1} - d_i = (x_{i+1} + 1)^2 + (y_{i+1} - \frac{1}{2})^2 - r^2$

$- \left[(x_i + 1)^2 + (y_i - \frac{1}{2})^2 - r^2 \right]$

$= (x_{i+1}+1)^2 + \left(y_{i+1} - \frac{1}{2}\right)^2 - (x_i+1)^2 - (y_i - \frac{1}{2})^2$

Since $x_{i+1} = x_i + 1$

$d_{i+1} - d_i = (x_i+2)^2 + \left(y_{i+1} - \frac{1}{2}\right)^2 - (x_i+1)^2 - \left(y_i - \frac{1}{2}\right)^2$

$= x_i^2 + 4 + 4x_i + y_{i+1}^2 + \frac{1}{4} - y_{i+1} - x_i^2 - 1 - 2x_i - y_i^2 - \frac{1}{4} + y_i$

$d_{i+1} - d_i = 2x_i + 3 + y_{i+1}^2 - y_i^2 - y_{i+1} + y_i$

$d_{i+1} = d_i + 2x_i + 3 + (y_{i+1}^2 - y_i^2) - (y_{i+1} - y_i)$

If $d_i < 0$

$\Rightarrow$ T pixel is chosen

$\Rightarrow y_{i+1} = y_i$

$\therefore d_{i+1} = d_i + 2x_i + 3 + (y_i^2 - y_i^2) - (y_i - y_i)$

$d_{i+1} = d_i + 2x_i + 3$

If $d_i \geq 0$

S pixel is chosen

$y_{i+1} = y_i - 1$

$d_{i+1} = d_i + 2x_i + 3 + ((y_i - 1)^2 - y_i^2) - (y_i - 1 - y_i)$

$$d_{i+1} = d_i + 2x_i + 3 + y_i^2 + 1 - 2y_i - y_i^2 - y_i + 1 + y_i$$

$$d_{i+1} = d_i + 2x_i - 2y_i + 5$$

$$d_{i+1} = d_i + 2(x_i - y_i) + 5$$

$$d_{i+1} = \begin{cases} d_i + 2x_i + 3 & \text{if } d_i < 0 \\ d_i + 2(x_i - y_i) + 5 & \text{if } d_i \geq 0 \end{cases}$$

Initial value for decision parameter d_i

Set $(0, r)$ be the starting pixel.

$$d_1 = (0+1)^2 + \left(r - \frac{1}{2}\right)^2 - r^2$$

$$d_1 = \frac{5}{4} - r$$

$$d_1 = 1 - r$$

error of being $\frac{1}{4}$ less than the precise value does not prevent d_1 from getting the appropiate sign.

ALGORITHM:

```
int x=0 , y=r , d=1-r
while ( x<=y)
{ setpixel (x,y);
  if (d<0)
    d = d + 2x + 3;
  else
    { d = d + 2(x-y) +5;
      y--;
    }
  x++;
}
```

Date _____

TWO-DIMENSIONAL VIEWING

Graphics package allows a user to specify which part of a defined picture is to be displayed and where that part is to be placed on the display device.

Viewing Pipeline

Window

A world coordinate area selected for display.

Viewport

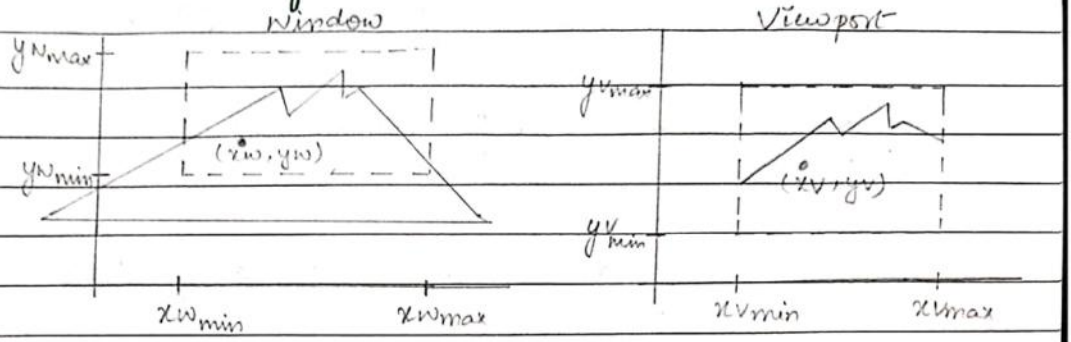
An area on a display device to which a window is mapped.

The window defines what is to be viewed; the viewport defines where it is to be displayed.

Mapping of a part of a world-coordinate scene to device coordinates is referred to as a viewing transformation.
2D viewing transformation is simply referred to as the window-to-viewport transformation or the windowing transformation.

Date _____

Window - To- Viewport Coordinate Transformation



A point (x_w, y_w) in the window is mapped into position (x_v, y_v) in the associated viewport.

To maintain same relative replacement in viewport as in the window — Same proportion ∴ Equal

$$\frac{x_v - x_{vmin}}{x_{vmax} - x_{vmin}} = \frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}}$$

and,

$$\frac{y_v - y_{vmin}}{y_{vmax} - y_{vmin}} = \frac{y_w - y_{wmin}}{y_{wmax} - y_{wmin}}$$

Now,

$$x_v - x_{vmin} = \frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}} \times (x_{vmax} - x_{vmin})$$

$$x_v = x_{vmin} + (x_w - x_{wmin}) \left(\frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}} \right)$$

Date _____

Similarly,

$$y_v = y_{v_{min}} + (y_w - y_{min})\left(\frac{y_{v_{max}} - y_{v_{min}}}{y_{w_{max}} - y_{w_{min}}}\right)$$

$$y_v = y_{v_{min}} + (y_w - y_{w_{min}})\, S_y$$

where S_x and S_y are scaling factors