# Computer Graphics & Animation

27/3/22

## Class - 1

Google Classroom Code: yizyto4

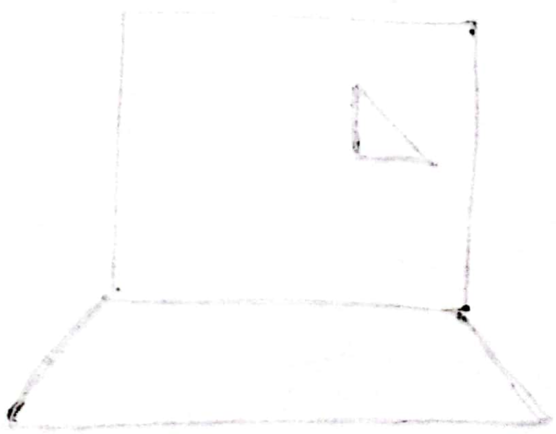
Book: Theory & Problems of Computer Graphics

ZHIGANG XIANG

ROY A. PLASTOCK

→ OLED vs LED?? why?

→ What is Computer Graphics?



RH

## Class-2

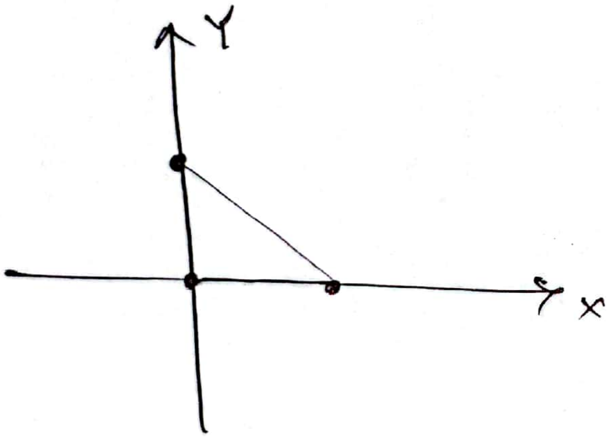
29/3/22

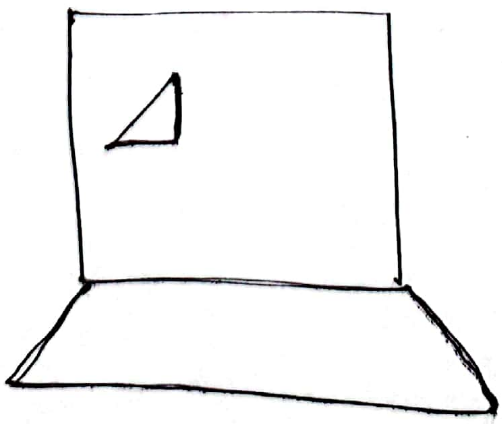
### Object Space:

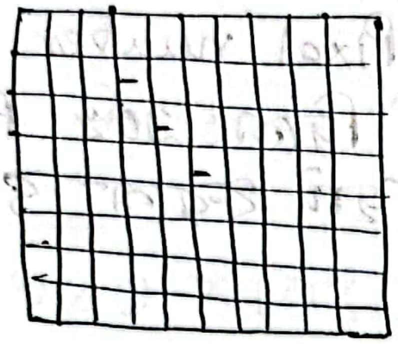
Subconcious mind এ যেখানে Draw কর হয়.

### Object Definition:

যেটা Draw করব সেটা Structurally দেখতে কেমন হবে.



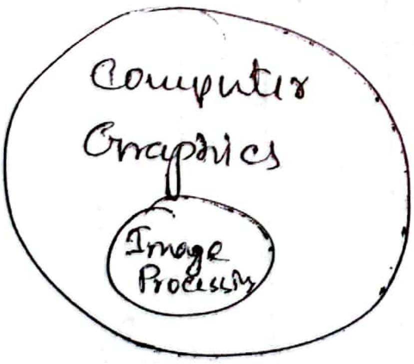


Scan Conversion → Continuous Space থেকে Discrete Space এ আসার Process.

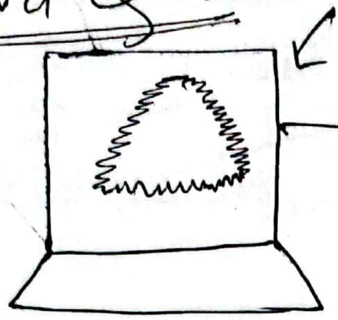
* Class-2 এবং Class-3 এর Syllabus এর উপর CT-01 এ আসবে

# Computer Graphics vs Image Processing

→ Semester Final এ দেখ.



## Board Viva Ques: কখন এটা Smooth হবে?



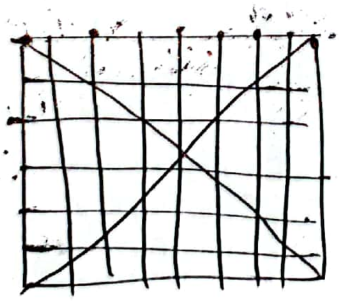
→ Pixel number বাড়িয়ে দিলে এবং Pixel Size ছোট করলে Smooth হবে।

## Image Space:

- → যেখানে Draw করব।
- → Physical Existence আছে।
- → এই Space এ Image Processing হয়।

## Image Representation:

Digital Image → A set of discrete pixels



Rectangular ~~[illegible]~~ আকারে হলো Raster

Digital Image হলো Raster Image

Resolution → No. of pixels per unit length

Aspect Ratio → Image width : Image Height

1920 × 1080 resolution এ

$\frac{1920}{1080} = \frac{16}{9} = 16:9$ aspect ratio

Image Coordinate:

↳ Coordinate কে প্রকাশ করতে একটা Matrix দিয়ে।

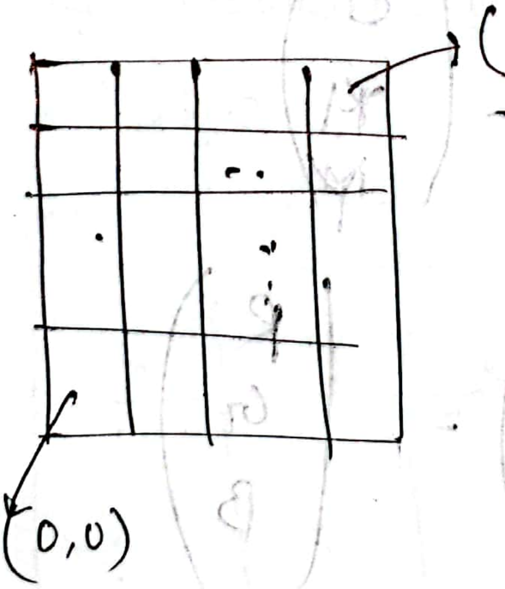



Image Coordinate এ (0,0) left bottom থেকে শুরু

## RGB color Space:

$\square \rightarrow (R, G, B)$

→ RGB is an additive Process
(এখন থেকে Math আসবে)

## CMY color Model:

↳ Printing এ use হয়

RGB ⟹ CYM

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} C \\ M \\ Y \end{pmatrix}$$

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

## Direct Coding:

→ Color গুলো Memory তে store করার way.

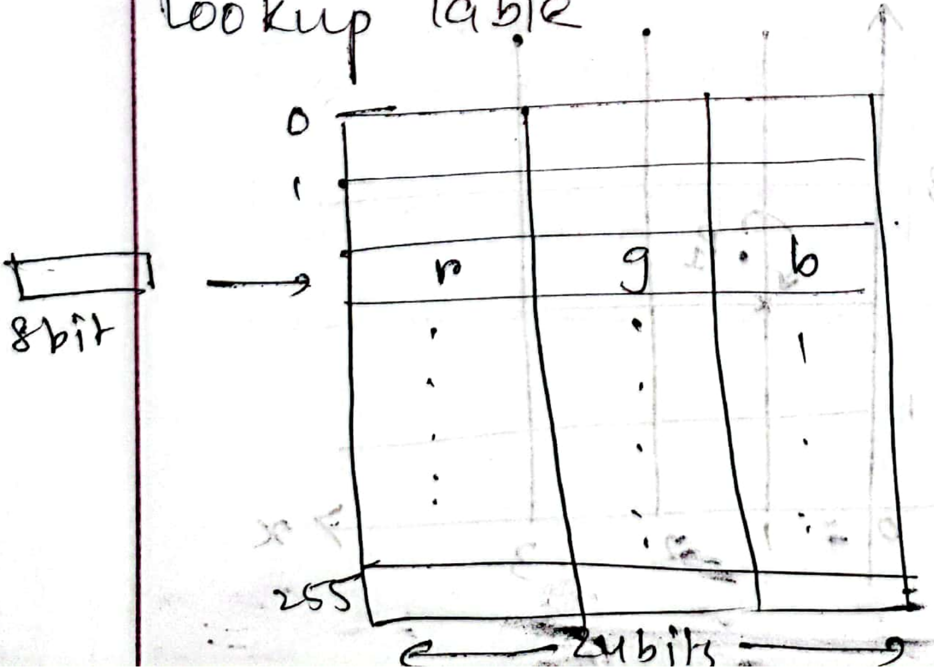
* $\boxed{\text{Bit Limitation}}$ থাকবে. থাকে

True Color Representation → প্রতিটা pixel এর channel এ
8 bit দেয়া হয়

So, ১ pixel এ (8+8+8)
= 24 bit

8 bit for R
8 " " G
8 " " B

আমাদের Problem Solve করতে আসছে
Lookup Table.




Ques: Why Direct Coding এর পরিবর্তে Lookup table?

MZI

# Class-03

2/4/22

## Computer Graphics

### Chapter-3

### "Scan Conversion"

Scan Conv. → Method to draw 1D/2D/3D.... objects

**Objective:**

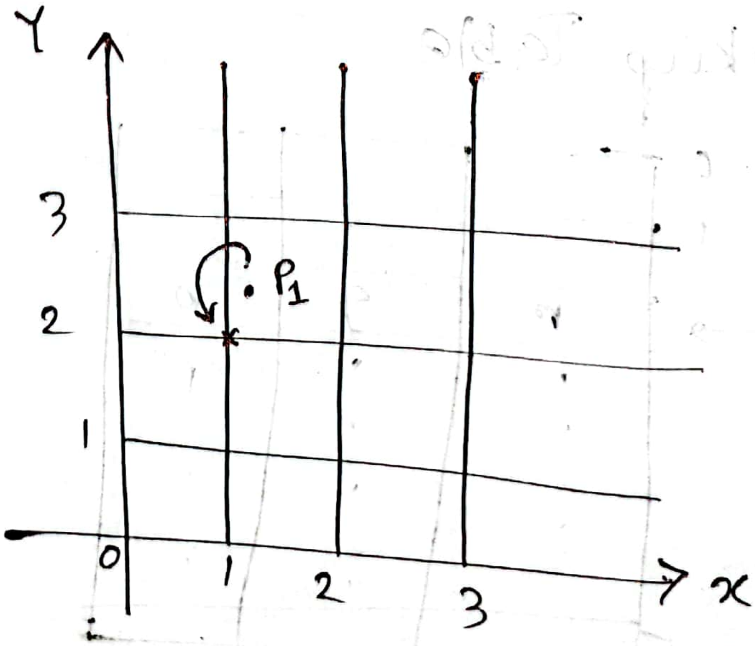
- To draw various 2D objects.

**1) Point Drawing:**

Let, the point be $(x,y)$

$(x,y) \longrightarrow (1.1, 2.3)$

$(x,y) \in \mathbb{R}$



# যেহেতু, Programming index এ fraction নাই
So, point গুলোকে আমরা mapping করব

## Mapping:

$$(x, y) \longrightarrow (x', y')$$

$x' = \lfloor x \rfloor$ (Floor of $x$) $= 1$

$y' = \lfloor y \rfloor$ (Floor of $y$) $= 2$

(*) Mapping এ floor এবং ceiling করতেছি
as 10, 100, ... etc. দিয়ে গুন করলে
domain range থেকে যাবে।

$$\boxed{\begin{aligned} x' &\le x < x'+1 \\ y' &\le y < y'+1 \end{aligned}}$$

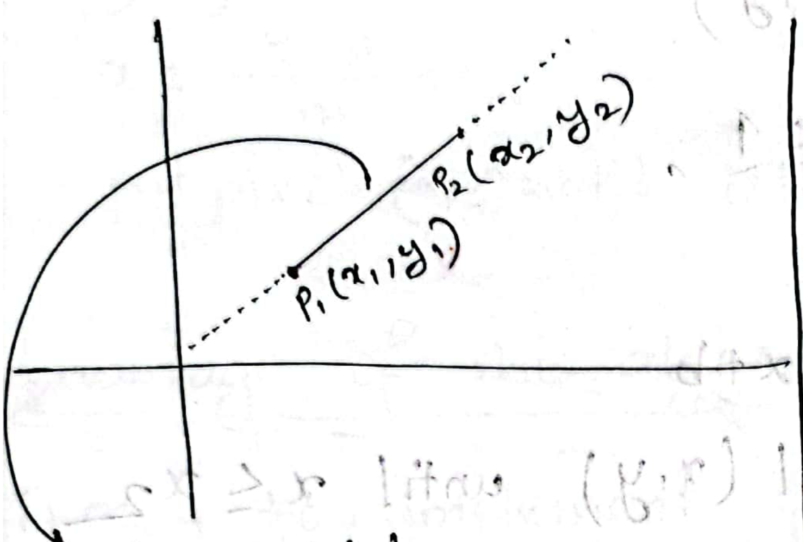
$x' = \lfloor x + 0.5 \rfloor$

$y' = \lfloor y + 0.5 \rfloor$ } এরকম হলে

$$\boxed{\begin{aligned} x' - 0.5 &\le x < x' + 0.5 \\ y' - 0.5 &\le y < y' + 0.5 \end{aligned}}$$

⊛ Floor use করি discontinuity দূর করার জন্য।

## 2) Line drawing using direct equation:




$y = mx + b$

$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$

$b = y_1 - mx_1 = y_2 - mx_2$

**input:**

two points

$(x_1, y_1)$, $(x_2, y_2)$

**Output:**

Draw the line using input

**Method:**

i) Find $m$

ii) Find $b$

iii) If $|m| \leq 1$

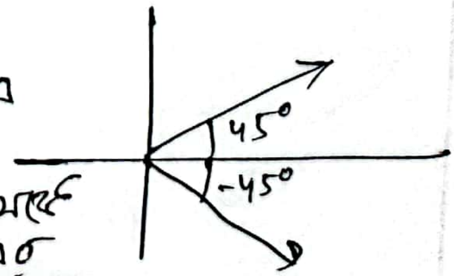
$-1 \leq m \leq 1$

$-1 \leq \tan\theta \leq 1$

$\tan^{-1}(-1) \leq \theta \leq \tan^{-1}(1)$

$\Rightarrow -45^\circ \leq \theta \leq 45^\circ$

অর্থাৎ হলো, slope -1 থেকে +1 এর মধ্যে



iii) initially set $x = x_1$, $y = y_1$

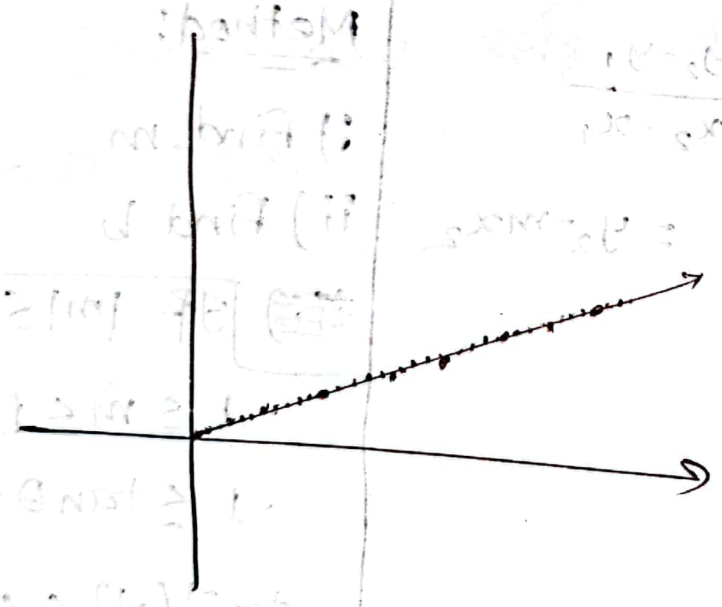
put pixel $(x, y)$

iv) if $|m| \le 1$,

$x = x + 1$

$y = mx + b$

put pixel $(x, y)$ until $x \le x_2$



else $(|m| > 1)$

$y = y+1$

$x = \dfrac{y-b}{m}$

put pixel $(x,y)$ until $y \leq y_2$

Advantage of this algo: (PROS)

i) easy to implement

Disadvantage: (CONS)

i) Floating point এর operation এর জন্য time complexity বাড়ে। (Floating Point Multiplication & Addition)

RH

# Class - 04

3/4/22

(*) Chapter-1,2 থেকে CT-01 এ থাকবে। আজকে class এ chapter-2 পর্যন্ত পড়ানো হয়ে যাবে।

## CRT

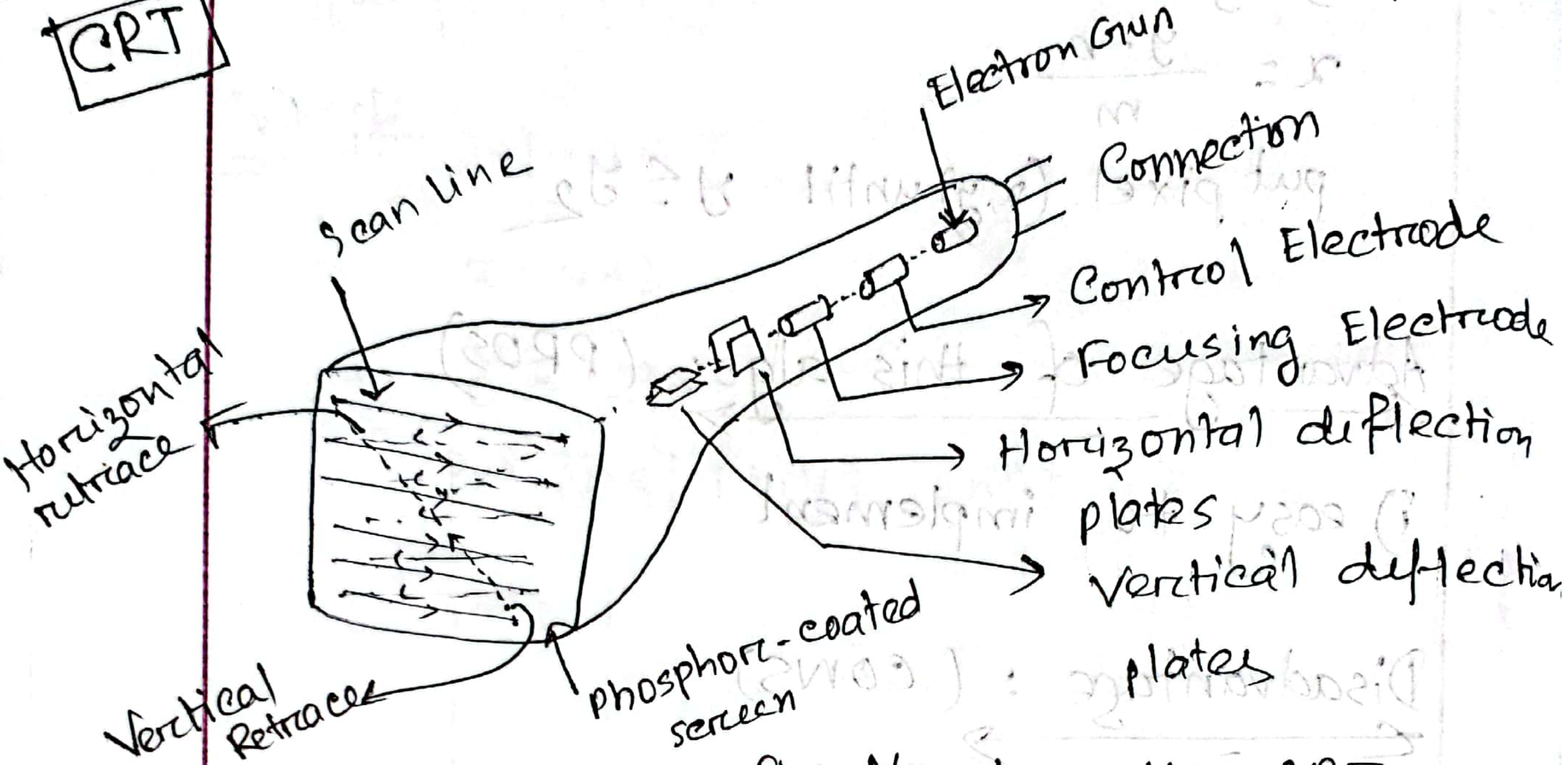



Fig: Monochromatic CRT

### Control Electrode:

Brightness control করে by controlling electron beam.

### Focusing Electrode:

দুর্বিয়ে পড়া electron কে Focused করে।

CRT ⇒ electric signal কে visual signal এ convert করে।

H.D.P & V.D.P:

→ Works for scan line's horizontal retrace (H.D.P)

→ " " vertical retrace এর (V.D.P)

~~horizontal retrace এর~~

Fluorescence:

Electron beam phosphorous কে hit করলে light emit হবে; কিন্তু, electron beam সরার পরেও কিছুক্ষণ light emit হবে এটাকে বলে Phosphorence.

Circuit (Content (0111....0..) কে প্রথমে electric signal এ convert করে)

↑

Memory

Frame Buffer/ Refresh Buffer

⇒ যেই Image/ Video টাকে দেখাতে চাইবো সেটা store করে রাখে F.B বা R.B তে.

→ এই electric signal টাই CRT তে processed হয়।

⊞ Big Electron beam সরিয়ে নিলেও Phosphorous টা কতক্ষণ আলো emit করবে

What is the condition here?

⇒ At least ততক্ষণ পর্যন্ত হতে হবে যতক্ষণ আমি দেখতে পাব, অর্থাৎ ততক্ষণই Next Frame আসার আগে চলে যেতে হবে।

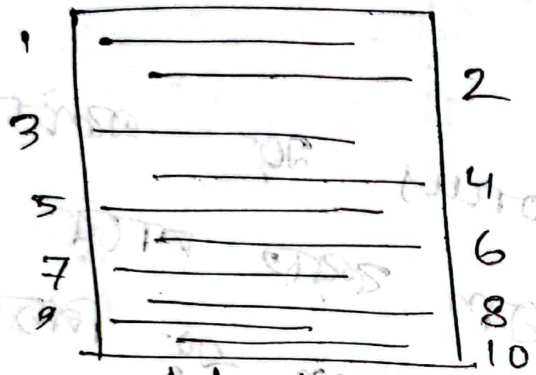
⊞ Refresh Rate:

১ sec এ কতগুলো Content আনব,

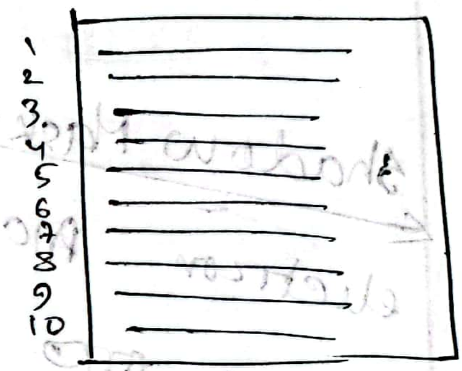
60Hz / 60 FPS → 60 Frame per sec.

Blinking Problem:

→ Low Refresh rate এ, অনেক frame loss হয়। ফলাফল দেখায় Screen. এটা দূর করতে এসে গেছে Interlacing.




Interlacing




Progressive Screen

Interlacing:

→ Screen Refresh হওয়ার সময় কমিয়ে দেয়।

Progressive Screen → প্রতিটি Scan line একবার Refresh করা হয়। Full screen Refresh করতে time বেশি লাগে, But, Interlacing এ Odd line First এ refreshed হয়, এতে Overall refresh

refresh
rate কমে না। But, speed বাড়িয়ে বাড়ে।

# Monochromatic এ electron Gun ১টা, RGB তে ৩টা Gun.

Shadow Mask:→ electron phosphorus এর একটা নির্দিষ্ট জায়গায় যাতে আঘাত করতে পারে তার জন্য screen এর পিছনে shadow থাকে, এর জন্য plate থাকে which creates shadow mask

RH

Class - 05

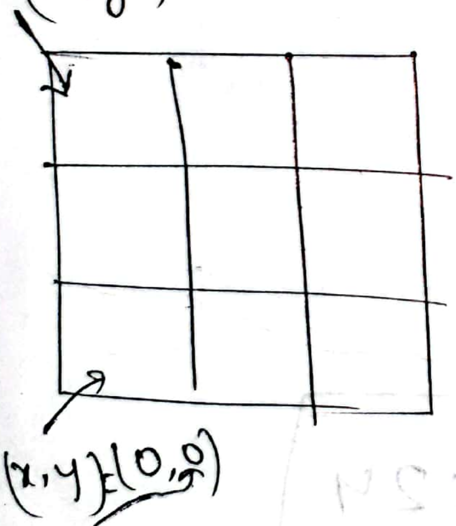
5/4/22

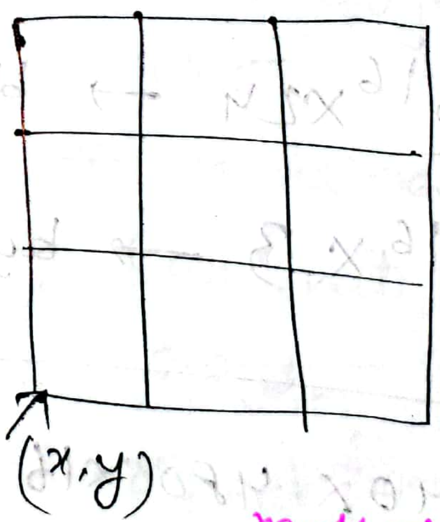
Maths: Chapter-2 (থেকে Ques. আসবে।

□ Compute the size of 640 × 480 image at 240 pixels per inch.

Ans: $\frac{640}{240} \times \frac{480}{240}$ inches.

■ ~~2×2~~




m − y − 1

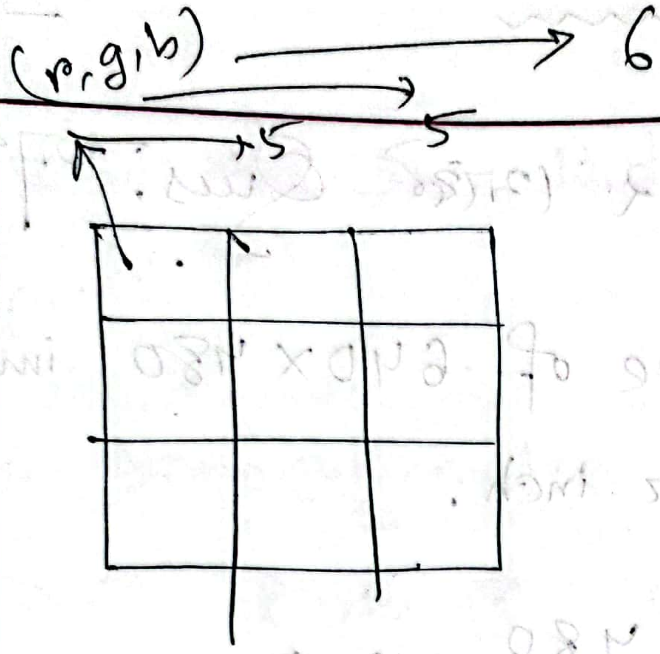
$x' = x$

$y' = 3 - 0 - 1 \rightarrow$ formulated $= m - y - 1$

3 → no. of vertical pixels

0 → y value

Slide 20 Math Solve




$2^5 \times 2^5 \times 2^6$

$= 2^{16}$ possible colour combinations

$\boxed{640 \times 480 \times (5+5+6)}$ → Image Size

$2^{16} \times 24$ → bit

$2^{16} \times 3$ → byte

$\boxed{640 \times 480 \times 16 + 2^{16} \times 24}$

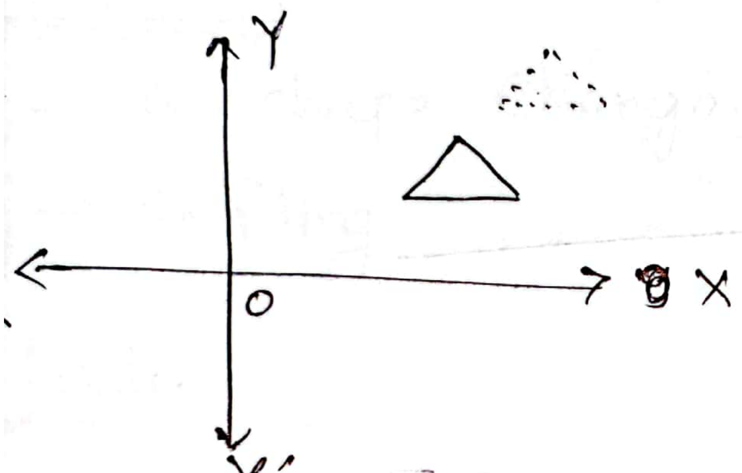
# Example 2.1 - 2.24

chapter-2

# Geometric Transformation

Transformation

Geometric     Coordinate

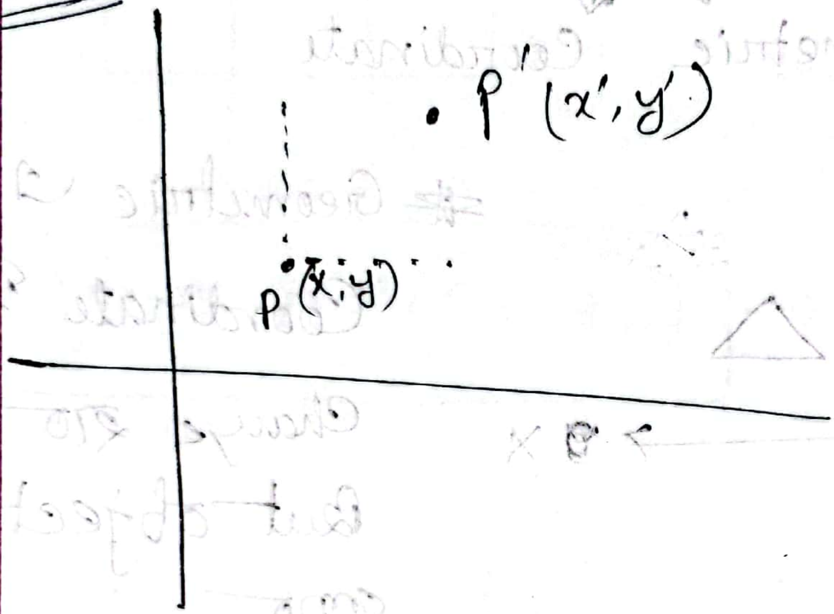



# Geometric এ
Coordinate Sys.
Change হয় না,
But object move
করে।

# Coordinate এ
Coordinate Sys move
করে। But
object যেখানে
আছে সেখানেই থাকে।

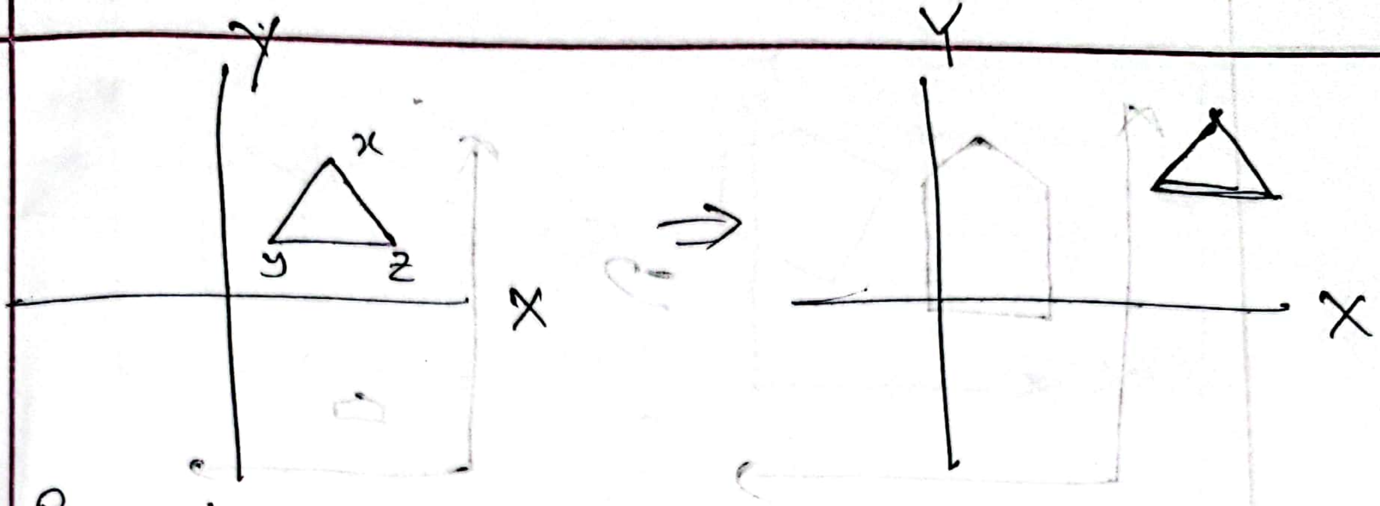
G.T. →
- Translation
- Scaling
- Rotation

## Translation:




$$P = \begin{bmatrix} x \\ y \end{bmatrix} \qquad T = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$$

$$P' = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$$




**Property:**

→ No shape changes

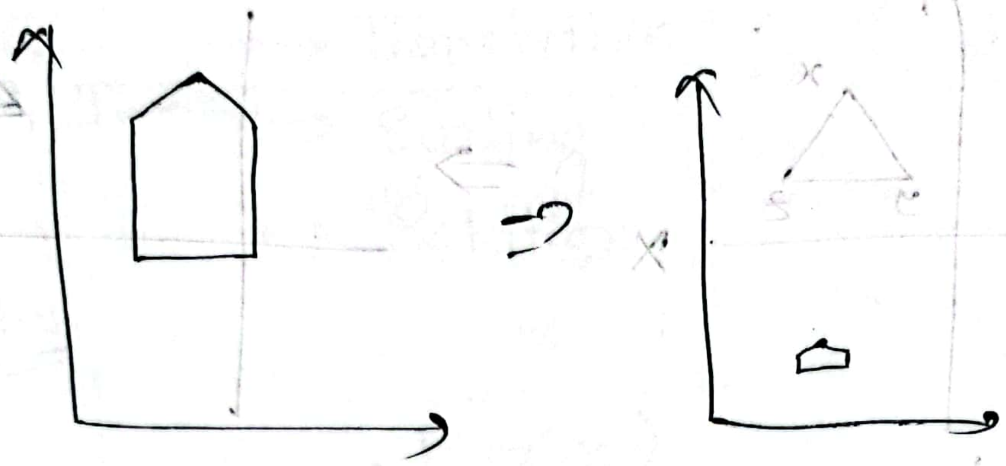
→ Shifting

## Scaling:

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

$x' = S_x . x$

$y' = S_y . y$



$S_x, S_y < 1$ হলে Shrinking

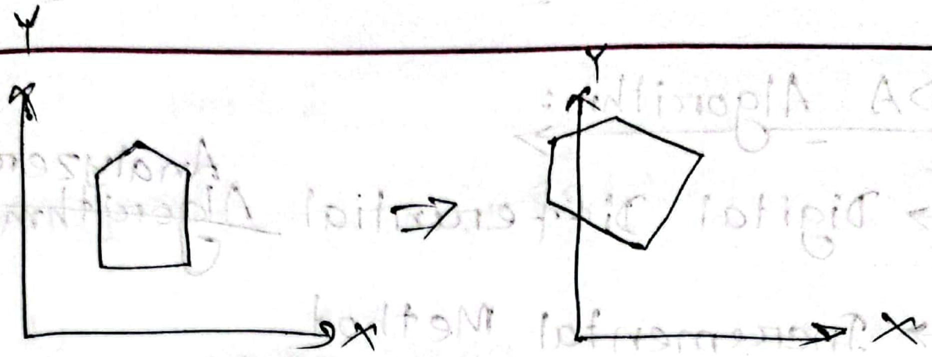
$S_x, S_y > 1$ হলে zooming.

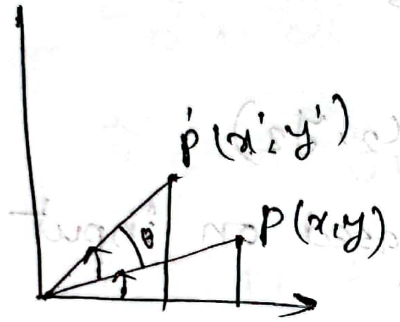
## Rotation: (After $\theta^\circ$ counter-clockwise rotation)

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

$$\cos(-\theta) = \cos\theta$$

$$\sin(-\theta) = -\sin\theta$$




chapter-5

Computer Graphics Principle and Practice (Foley)

→ Geometric Transformation এখান থেকে পড়ানো হয়েছে।

Class - 06 9/4/22

DDA Algorithm:

→ Digital Differential ~~Algorithm~~ Analyzer

→ Incremental Method

→ Use previous calculation

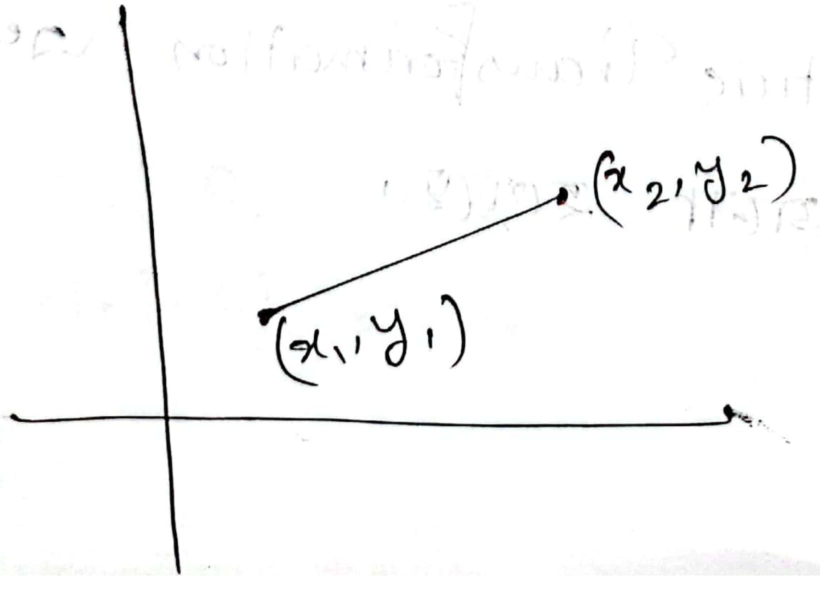
input: $(x_1, y_1), (x_2, y_2)$

Output: A line based on input

Method: i) Find $m = \frac{\Delta y}{\Delta x}$

ii) $x_i = x_1, y_i = y_1$ $(x_1 \le x_2$ & $y_1 \le y_2)$

put pixel $(x_i, y_i)$



iii). if $|m| \leq 1$

$$x_{i+1} = x_i + 1 , \ (\Delta x = 1)$$

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

$$\Rightarrow y_{i+1} - y_i = m \Delta x$$

$$\Rightarrow y_{i+1} - y_i = m \cdot 1$$

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

put pixel $(x_{i+1}, y_{i+1})$

Repeat until $x_{i+1} \leq x_2$

iv) if $|m| > 1$

$y_{i+1} = y_i + 1 \quad (\Delta y = 1)$

We know,

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

$\Rightarrow \Delta x = \Delta y / m$

$\Rightarrow x_{i+1} - x_i = \frac{1}{m}$

$\Rightarrow x_{i+1} = x_i + \frac{1}{m}$

putpixel $(x_{i+1}, y_{i+1})$

Repeat until $y_{i+1} \leq y_2$

Disadv:

→ floating point additim at $y_{i+1} = y_i + m$

or $x_{i+1} = x_i + \frac{1}{m}$

CamScanner

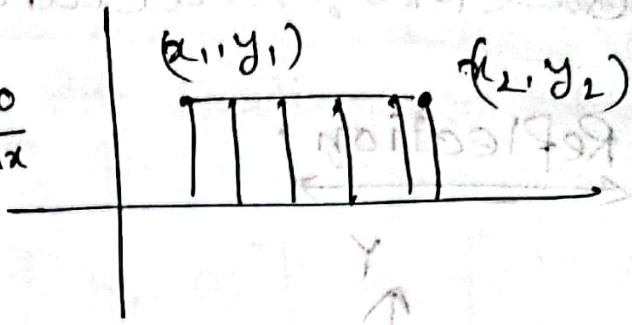
# Special Cases:

(No algorithm needed for special cases)

* $m=0$ হলে,

$x \leftarrow x+1$, $\frac{\Delta y}{\Delta x} = \frac{0}{\Delta x}$

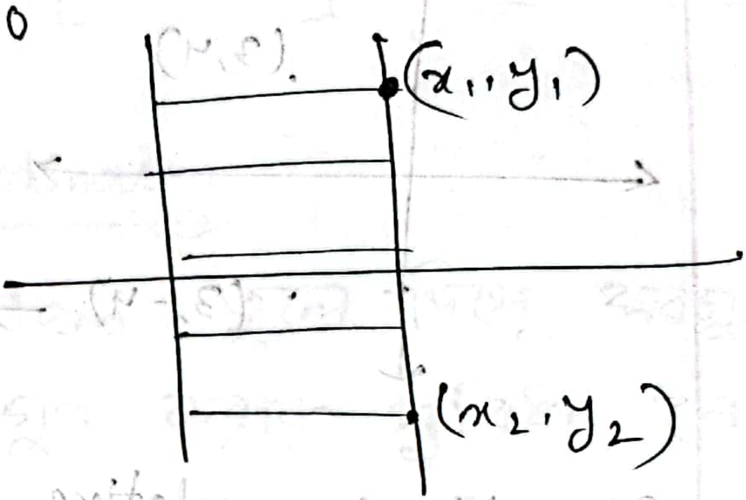
$y = y_1 = y_2$




* $m=\infty$ হলে, $\Delta x = 0$

$y \leftarrow y+1$

$x = x_1 = x_2$




* $m=1$ হলে,

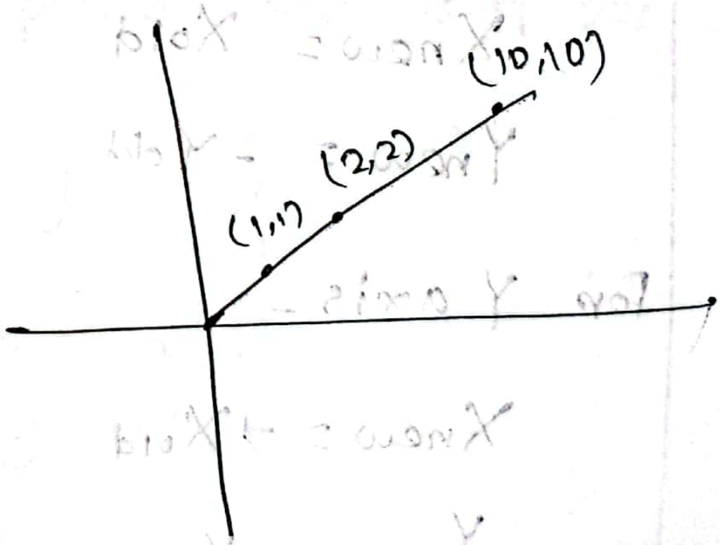
$x \leftarrow x+1$

put pixel $(x, y)$

$\downarrow$

$(x, x)$

নতুন করে y বের করার দরকার নাই। কারণ, সমান হবে তো x এবং.

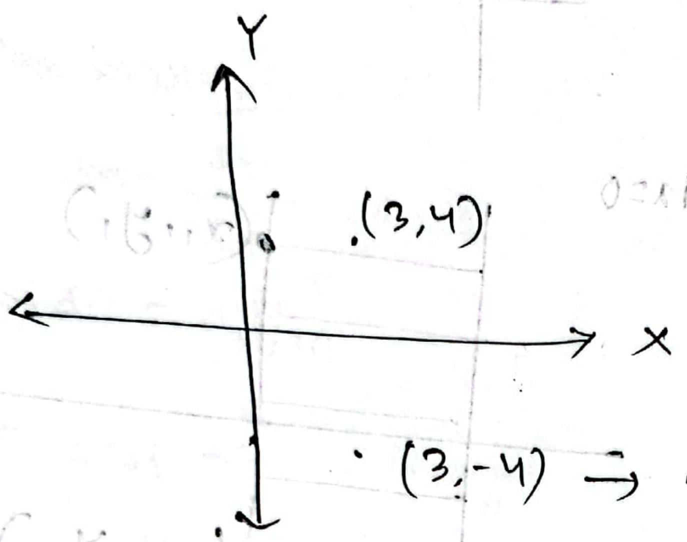


10/4/22

## Class - 07

✱ আজকের Lecture অনেক Important.

**Reflection:**




(3,-4) ⇒ x axis এর সাপেক্ষে reflected point

For X-axis relative reflection-

$X_{new} = X_{old}$

$Y_{new} = -Y_{old}$

For Y axis -

$X_{new} = -X_{old}$

$Y_{new} = Y_{old}$

⇒ Reflection is a variant of scaling.
কারণ এটার Matrix টা Scaling এর Matrix এর মতোই। Like -

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

## Homogeneous Coordinate:

Addition এর system তুলে দিয়ে সবগুলো Operation এই গুণ করার সুবিধা নেয়ার জন্য use করব Homogeneous Coordinate.

এখানে একটা 3rd plane add করা হয়,

$$(x, y) \longrightarrow (x', y', w)$$

$w = 1$ হলে, $x = x'$,
$y = y'$

⊛ $w = 1$ হলে Homogeneous coordinate টা 2-D coordinate বলতে পারব,

Ex- $\underline{(2,3,6)}$ and $\underline{(4,6,12)}$ are same in homogeneous coordinate.

$$\therefore \left(\frac{2}{6}, \frac{3}{6}, 1\right) \rightarrow \left(\frac{4}{12}, \frac{6}{12}, 1\right)$$

$$(x', y', w) \longrightarrow (x'/w, y'/w, 1)$$

$$\downarrow$$

$$(x'/w, y'/w)$$

# (0, 0, 0) doesn't exist at H.C.

$W = 1$ হ'ব লাগিব at least.

**H.C to 2-D Coordinate:**

$$(2, 3, 6)$$

$$\Rightarrow \left(\frac{2}{6}, \frac{3}{6}, 1\right)$$

$$\Rightarrow \left(\frac{1}{3}, \frac{1}{2}\right) \longrightarrow \text{2D coordinate}$$

$$(x^*, y^*) \to (x', y')$$

old point → $(x^*, y^*)$; new point → $(x', y')$

in

e .

3 primary basic transformations of H.C.

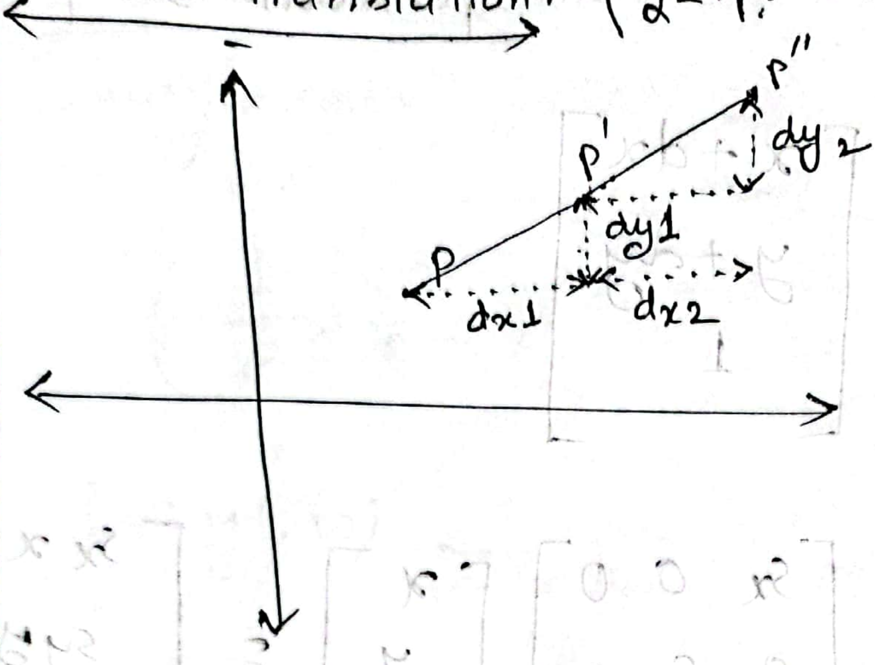
① $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x + dx \\ y + dy \\ 1 \end{bmatrix}$$

② $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} Sx\,x \\ Sy\,y \\ 1 \end{bmatrix}$$

③ $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \\ 1 \end{bmatrix}$$

## Double Translation: (দুইবার Translate করা হ




$$P' = T(dx_1, dy_1) \cdot P$$

$$P'' = T(dx_2, dy_2) \cdot P'$$

$$= T(dx_2, dy_2) \cdot T(dx_1, dy_1) \cdot P$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx_2 \\ 0 & 1 & dy_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & dx_1 \\ 0 & 1 & dy_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 + 0 + 0 \\ 0 + 0 + dx_1, dy_2 \\ 0 + \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & dx_1 + dx_2 \\ 0 & 1 & dy_1 + dy_2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scaling:

তাহলে Similarly Double Scaling করলে,

$$\begin{bmatrix} Sx_1 \cdot Sx_2 & 0 & 0 \\ 0 & Sy_1 \cdot Sy_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# Derivation এর same as Double Translation

H.W Double Rotation এর Derivation

( এইটা important for exam )

For Double Rotation -

$$\begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

এখান থেকে Derive করব।

⊕ Proof করতে বলতে পারে Double Rotation is an additive process.
Additive: result বলতে end result এর $\sin(\theta_1+\theta_2)$, $\cos(\theta_1+\theta_2)$ আকারে আসবে।

⊕ Double Rotation এর Derivation proof প্রায়ই আসে।

Same ই Derive করে দেখাই দিতে হবে যে additiv

Class - 08 12/4/22

## Rotate about any arbitrary Point :

$P_1$ এর সাপেক্ষে rotate করতে হবে,

So, First এ $P_1$ কে origin এ নিয়ে আসি

$P_1(x, y)$

$$T_{P_1} \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix}$$

$R_{\theta_1}$ এ Rotate করাতে হবে,

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

আবার $P_1$ এ ফিরে যাব.

$$T_{P_1} \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

এই 3টের Matrix multiply করলে: এ ultimate $P_1$ এর সাপেক্ষে Rotation পাব.

$$T_{P_1} \times R_{\theta_1} \times T_{-P_1}$$

**Ques:** (V.V.I for exam this topic)

→ Derive করতে বলতে পারে.

→ Value দিয়ে Calculation করতে বলতে পা

⊛ Multiplication টা অনেকখানি (যেহে হয়ে আসবে.

## Scale about any arbitrary Point:

$$T(x_1, y_1) \cdot S(s_x, s_y) \cdot T(-x_1, -y_1)$$

$$= \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

Derivation

example

$$= \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & -s_x \cdot x_1 \\ 0 & s_y & -s_y \cdot y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} s_x & 0 & -s_x \cdot x_1 + x_1 \\ 0 & s_y & -s_y \cdot y_1 + y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

## Scale and Rotate about any arbitrary Point:

**Step 1:** $P_1$ কে origin এ আনব, $T_{-P_1}$

**Step 2:** Scale করব $S_{x,y}$

**Step 3:** Rotate করব $R_\theta$

**Step 4:** Translate to final position $P_2$

$$\boxed{T_{P_1} \; R_\theta \; S_{x,y} \cdot T_{-P_1}}$$

Chap-4 → SCHAUM's outline

MATH

## Order of Operation:

Matrix এ $AB \neq BA$ always.

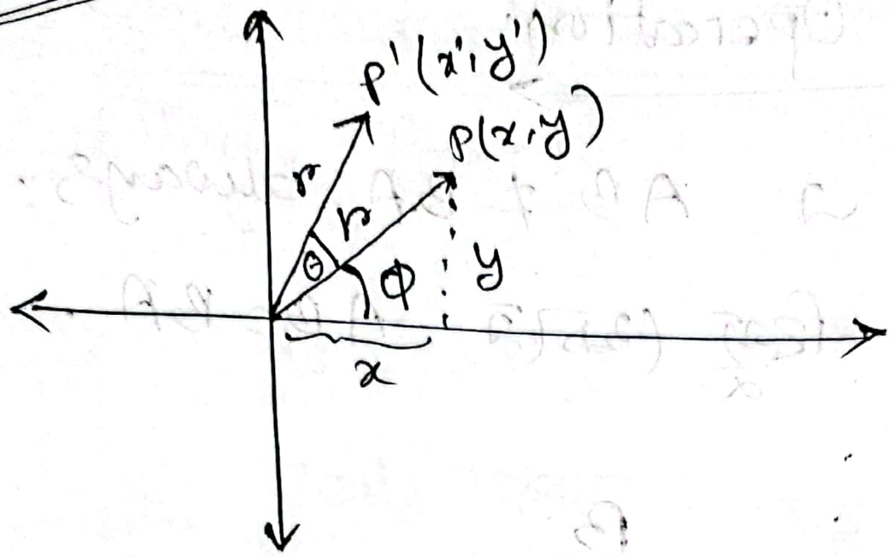
But কিছু কিছু ক্ষেত্রে $AB = BA$.

| A | B |
|---|---|
| Translate | Translate |
| Scale | Scale |
| Rotate | Rotate |
| Scale (with $S_x = S_y$) | Scale (with $S_x = S_y$) |

এক্ষেত্রে হবেই $AB = BA$.

Problem Related to 2D Transformation (Slide

Prob!:




For, P,

$$\vec{r} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r\cos\phi \\ r\sin\phi \end{bmatrix}$$

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

$\therefore x = r\cos\phi$

Simillarly,

$y = r\sin\phi$

For $P'$,

$$\vec{r}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} r\cos(\theta+\phi) \\ r\sin(\theta+\phi) \end{bmatrix}$$

$$\cancel{\cos} \quad = \begin{bmatrix} r\cos\theta\cos\phi - r\sin\theta\sin\phi \\ r\sin\theta\cos\phi + r\sin\phi\cos\theta \end{bmatrix}$$

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

$\left[\because x = r\cos\phi, \; y = r\sin\phi\right]$

$\therefore$ Matrix representation for this rotation

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \cancel{x}\sin\theta & +\cos\theta \end{bmatrix}_{2\times 2}$$

ঠিক answer হলনা

(Ans).

Problem 2

$$\begin{bmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{bmatrix}$$

Problem 3

$$\begin{bmatrix} \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix} \begin{bmatrix} 2 \\ -4 \end{bmatrix}$$

Problem-6

i) $$\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

গুণ হবে আমাদের এরকম

$$\begin{bmatrix} 2 & 5 & 7 \\ 3 & 6 & 8 \\ 1 & 1 & 1 \end{bmatrix}$$

So, rotation হবে এর point (2,3)
(5,6)
(7,8)

But,

$$\begin{bmatrix} 2 & 5 & 7 \\ 3 & 6 & 8 \\ 6 & 12 & 14 \end{bmatrix}$$ হলেও আমার নিচে 1 1 1 আনতে হবে,

Then, point নিবো,

আসবে হবে $(2/6, 3/6)$, $(5/12, \frac{6}{12})$, $(7/14, \frac{8}{14})$

RH

# LAB-01

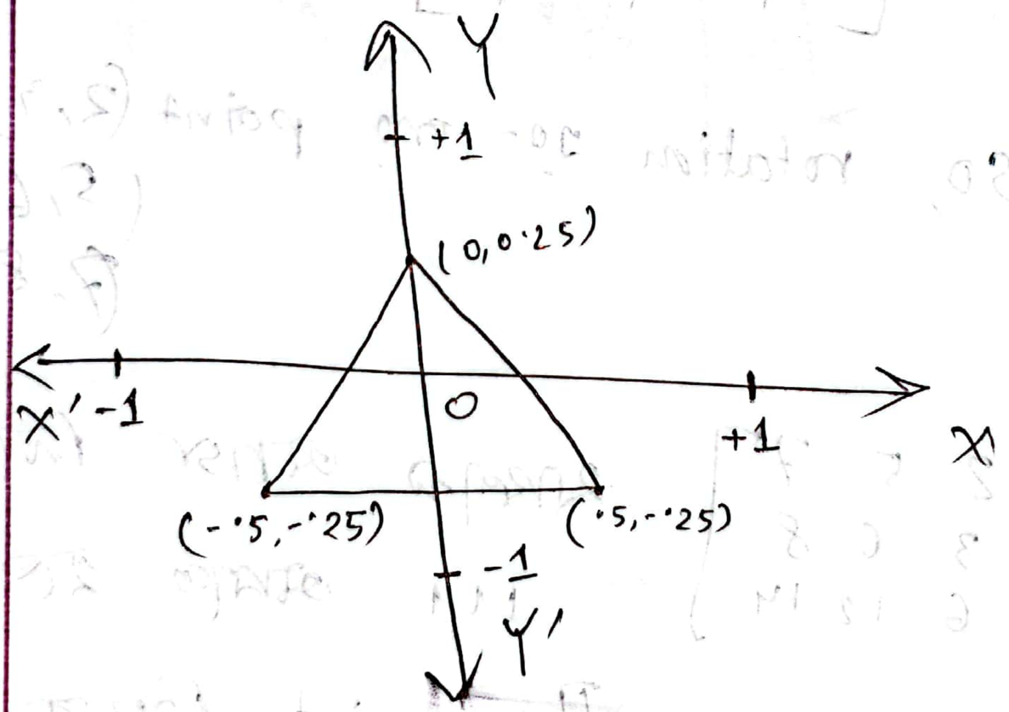
14/5/22

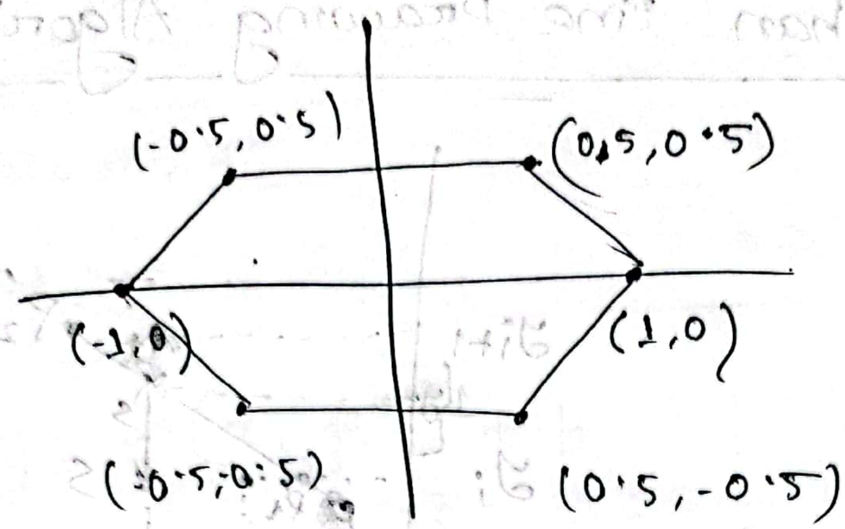
⇒ Initially the installation of Glut / OpenGL was done.

⇒ main.cpp was run to observe animated 3D drawings.

⇒ Basic.txt file was pasted in main.cpp to see how a basic program works.

## Coordinate System in OpenGL:



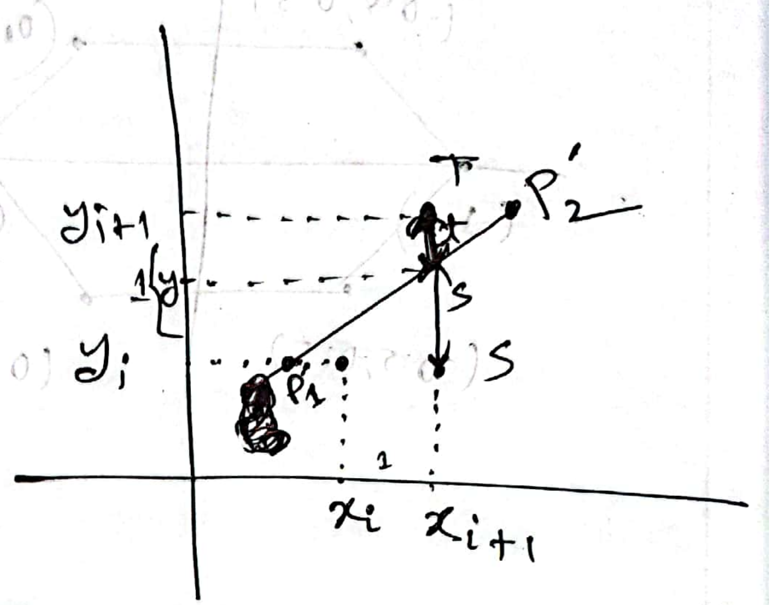

(-0.5, 0.5)
(0.5, 0.5)
(-1, 0)
(1, 0)
(-0.5, -0.5)
(0.5, -0.5)

MZI

## Class - 09+10

14/5/22

### Bresenham Line Drawing Algorithm:




দুইটি point $P_1$, $P_2$ দেয়া থাকবে। এর উপর
ভিত্তি করে Line Draw করব।

**S-point:**

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

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

**T-point:**

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

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

**Actual:**

$y = mx + b$

At point, $x = x_{i+1}$

$y = m(x_{i+1}) + b$

$\Rightarrow y = m(x_i + 1) + b$ —— (1)

$s = y - y_i$

$t = (y_i + 1) - y$

$s - t = y - y_i - [(y_i + 1) - y]$

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

$= 2y - 2y_i - 1$

$= 2[m(x_i + 1) + b] - 2y_i - 1$ —— (11)

$\boxed{= 2mx_i + 2m + 2b - 2y_i - 1}$

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

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

$$s - t = \frac{2\Delta y x_i + 2\Delta y + 2b\Delta x - 2y_i\Delta x - \Delta x}{\Delta x}$$

$$\Rightarrow (s - t)\,\Delta x = 2\Delta y x_i - 2y_i\Delta x + 2\Delta y + 2b\Delta x - \Delta x$$

$$\Rightarrow (s - t)\,\Delta x = 2\Delta y x_i - 2y_i\Delta x + \underbrace{2\Delta y + \Delta x(2b-1)}_{c}$$

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

Let,

$(s - t)\,\Delta x = d_i$ [$d_i$ is decision variable]

So,

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

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

Now,

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

~~$\Rightarrow d_{i+1} - d_i = 2\Delta y(x_i + 1$~~

If s-point is selected :

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

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

So, (iii),

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

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

$s - t \geq 0$ মানে $s \geq t$

So, $t$ less distance এ আছে line থেকে,

$(s-t)\Delta x \geq 0$ হলে

$\Rightarrow d_i \geq 0$ হলে (T point selected)

** $(s-t) \geq 0$ হলে — ও $(s-t)\Delta x \geq 0$ হবে

$\Rightarrow d_i < 0$ হলে (s.point selected)

$\rightarrow d_{i+1} = d_i + 2\Delta y$ (if spoint, $d_i < 0$) —— (IV)

T-point:

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

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

So, from (III),

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

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

($d_i \geq 0$ & T-point selected)

So, finally -

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

From eq$^n$ (ii),

$$s - t = 2[m(x_i + 1) + b] - 2y_i - 1$$

$$\Rightarrow (s-t) = 2mx_i + 2m + 2b - 2y_i - 1$$

$$= 2(mx_i + b) + 2m - 2y_i - 1$$

$$= 2y_i + 2m - 2y_i - 1$$

$$\Rightarrow (s-t) = 2m - 1$$

$$\Rightarrow (s-t)\Delta x = (2m - 1)\Delta x$$

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

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

So,

$d_1 = 2\Delta y - \Delta x$ ——— (v)

~~##~~ Another process to find $d_1$,

$s - t = 2[m(x_i + 1) + b] - 2y_i - 1$

$\Rightarrow (s-t)\Delta x = \Delta x(2mx_i + 2m + 2b - 2y_i - 1)$

$\cancel{\Rightarrow (s-t)\Delta x = 2m\Delta x x_i + 2m\Delta x + 2b\Delta x - 2y_i\Delta x - \Delta x}$

$\Rightarrow d_i = \Delta x(2mx_i + 2m + 2b - 2y_i - 1)$

$\therefore d_1 = \Delta x(2mx_1 + 2m + 2b - 2y_1 - 1)$

$= \Delta x\{2(mx_1 + b) + 2m - 2y_1 - 1\}$

$= \Delta x(\cancel{2y_1} + 2m - \cancel{2y_1} - 1)$

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

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

$$\therefore d_1 = 2\Delta y - \Delta x$$ ✓

So, till now

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

$$d_1 = 2\Delta y - \Delta x$$

**Algorithm:**

**input:**

2 points $P_1'(x_1', y_1')$

$P_2'(x_2', y_2')$

**Output:**

A line based on the input

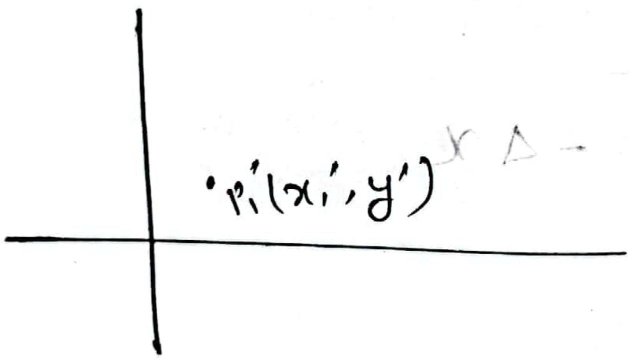
Method:

i) $x = x_1'$, $y = y_1'$

$d = 2dy - dx$ ; $dT = 2(dy - dx)$

$dS = 2dy$

ii) set pixel $(x, y)$




```
while (x ≤ x2'){
    x++;
    if di < 0 {
        d = d + dS;}
    else {
        d = d + 2dT;
        y++;}
}
```

set pixel (x,y);

} //end of while loop

Adv:

① এখানে Floating point এর operation removed হয়। So, আগের গুলোর চাইতে faster.

*** [Derivation আসবে exam এ]

RH

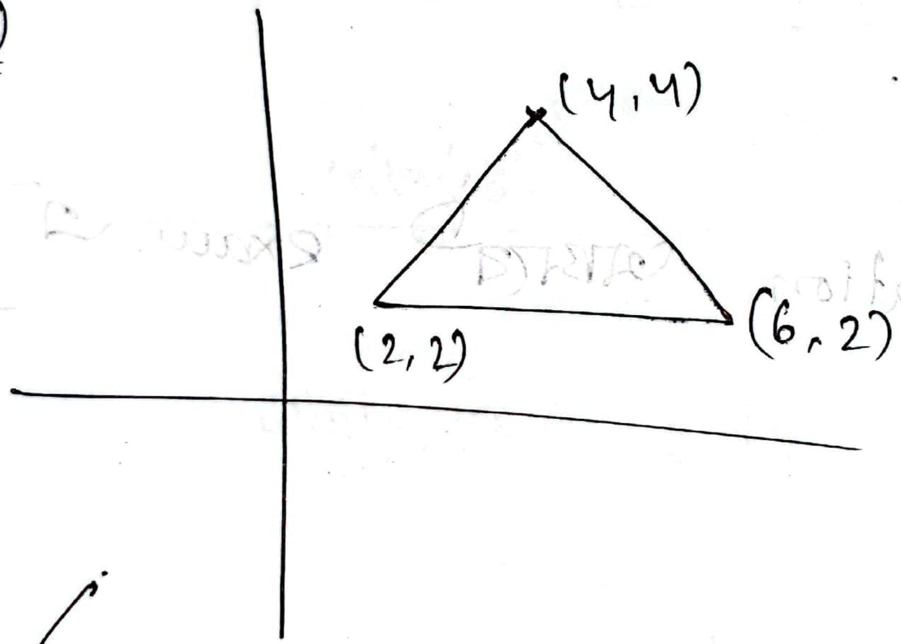
Class-11

17/5/22

Class Test -01 Syllabus:

Chapter 1, 2

Today a review class on Chapter 4 & 5 is taken.




এই Triangle কে Origin হতে সাপেক্ষে 30° rotate করাতে হবে.

Sol$^n$:

$R_{30°}$ কে Rotate করার matrix পেলাম।

$$R_{30} = \begin{bmatrix} \cos 30 & -\sin 30 & 0 \\ \sin 30 & \cos 30 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{30} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So, now
it has to be multiplied by.

$$P = \begin{bmatrix} 2 & 4 & 6 \\ 2 & 4 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$P' = R_{30} \cdot P = \begin{bmatrix} \sqrt{3}/2 & -1/2 & 0 \\ \frac{1}{2} & \sqrt{3}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 6 \\ 2 & 4 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{3} - 1 + 0 & 2\sqrt{3} - 2 + 0 & 3\sqrt{3} - 1 + 0 \\ 1 + \sqrt{3} + 0 & 2 + 2\sqrt{3} + 0 & 3 + \sqrt{3} + 0 \\ 0 + 0 + 1 & 0 + 0 + 1 & 0 + 0 + 1 \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{3} - 1 & 2\sqrt{3} - 2 & 3\sqrt{3} - 1 \\ 1 + \sqrt{3} & 2 + 2\sqrt{3} & 3 + \sqrt{3} \\ 1 & 1 & 1 \end{bmatrix}$$

(Ans)

* Perform 45° rotation of triangle A (0, 0), B (1, 1), C (5, 2)

Rotate this about P (-1, -1)

$$\underset{T_{P(x,y)}}{\begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}} \underset{R_\theta}{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}} \underset{T_{-P(x,y)}}{\begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix}}$$

Soln:

$$R_{45} \cdot T_{-P(-1,-1)} = \begin{bmatrix} \cos 45 & -\sin 45 & 0 \\ \sin 45 & \cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -(-1) \\ 0 & 1 & -(-1) \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}}+0+0 & 0-\frac{1}{\sqrt{2}}+0 & \frac{1}{\sqrt{2}}-\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}+0+0 & 0+\frac{1}{\sqrt{2}}+0 & \frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}} \\ 0+0+0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{2}{\sqrt{2}} \\ 0 & 0 & 1 \end{bmatrix}$$

Now;

$$T_{p(-1,-1)} \cdot R_{45} \cdot T_{-p(-1,-1)}$$

$$= \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \sqrt{2} \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt2}+0+0 & -\frac{1}{\sqrt2}+0+0 & \cancel{\frac{1}{\sqrt2}+0+0}\ 0+0-1 \\ 0+\frac{1}{\sqrt2}+0 & 0+\frac{1}{\sqrt2}+0 & 0+\sqrt2-1 \\ 0+0+0 & 0+0+0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt2} & -\frac{1}{\sqrt2} & -1 \\ \frac{1}{\sqrt2} & \frac{1}{\sqrt2} & \sqrt2-1 \\ 0 & 0 & 1 \end{bmatrix}$$

এটি হল final rotation Matrix.

Now, এটাকে দিয়ে,

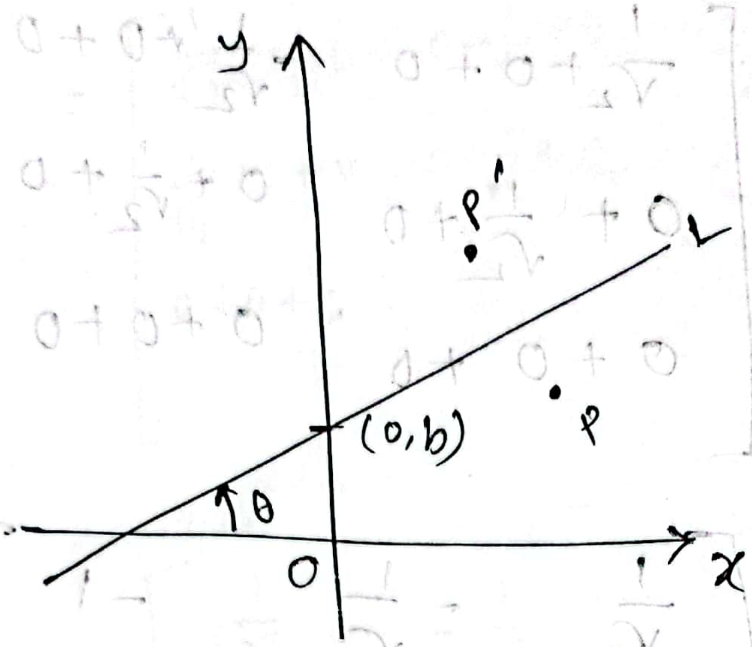
$$\begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

এর গুণ দিলে final coordinate matrix পাওয়া যাবে।

RH

# Class-12

22/5/22




* To find the reflection of P on the other side of line L.

Tips:

Line টাকে θ° rotate করে x অথবা y axis এ align করব. Then, ঐ axis অনুযায়ী reflection perform করব।

steps:

① Translate the intersection point B to the origin.

② Rotate by $-\theta^\circ$ so that line L align with $x$ axis.

③ Mirror reflect about the $x$ axis.

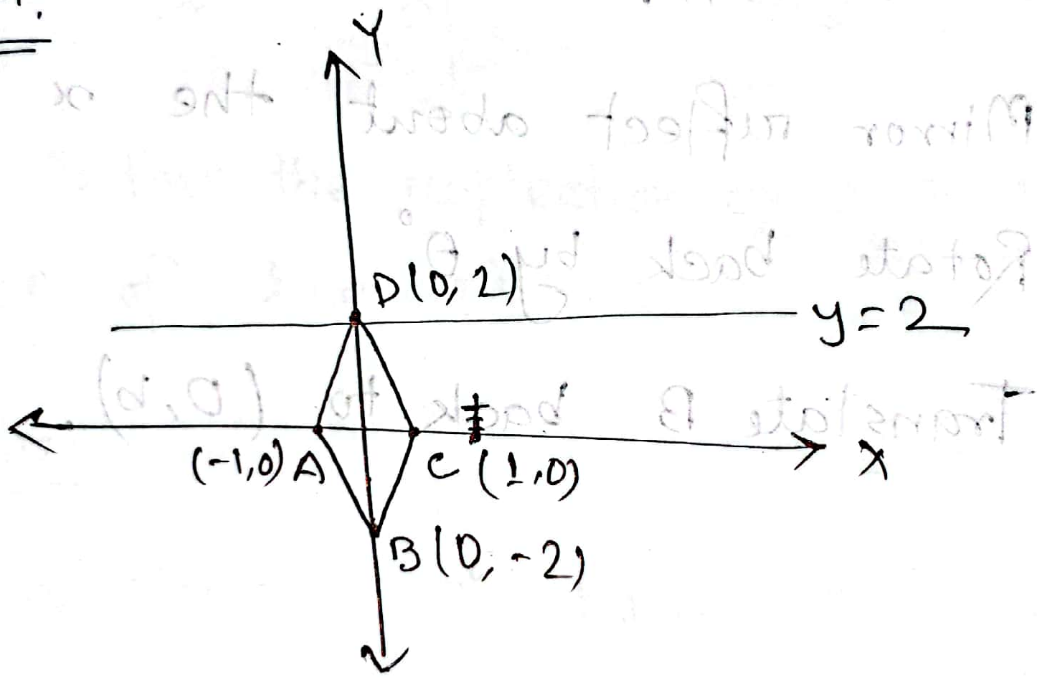
④ Rotate back by $\theta^\circ$.

⑤ Translate B back to $(0, b)$.

⇒ Reflect the diamond-shaped polygon whose vertices are $A(-1,0)$, $B(0,-2)$, $C(1,0)$ and $D(0,2)$ about

(a) the horizontal line $y = 2$

Sol$^n$:




Step 1:

Translate

$$T_{-b} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

; Now the line $y=2$ is on $x$ axis

**Step 2:**

Rotation করার লাগবে না as already X-axis এর সাথে aligned.

**Step 3:** (Reflection)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

**Step 4:** ~~Translate b~~ (Rotate back)

এটাও লাগবে না, as আগেও rotate করা লাগে।

**Step 5:** (Translate back)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 2:

Rotation করার লাগবে না as already X-axis এর সাথে aligned.

Step 3: (Reflection)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

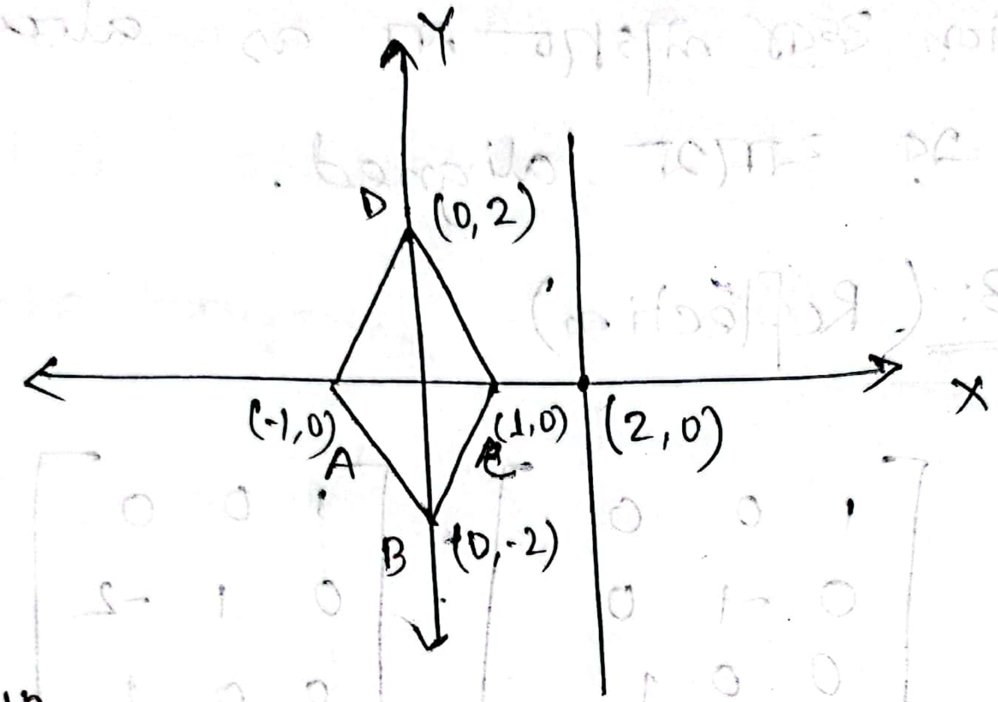
Step 4: ~~Translate b~~ (Rotate back)

কোন লাগবে না, as আগেও rotate করা লাগে-.

Step 5: (Translate back)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

(b) the vertical line $x = 2$




Soln:

Translate

$$T_{-b} = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

No Rotation

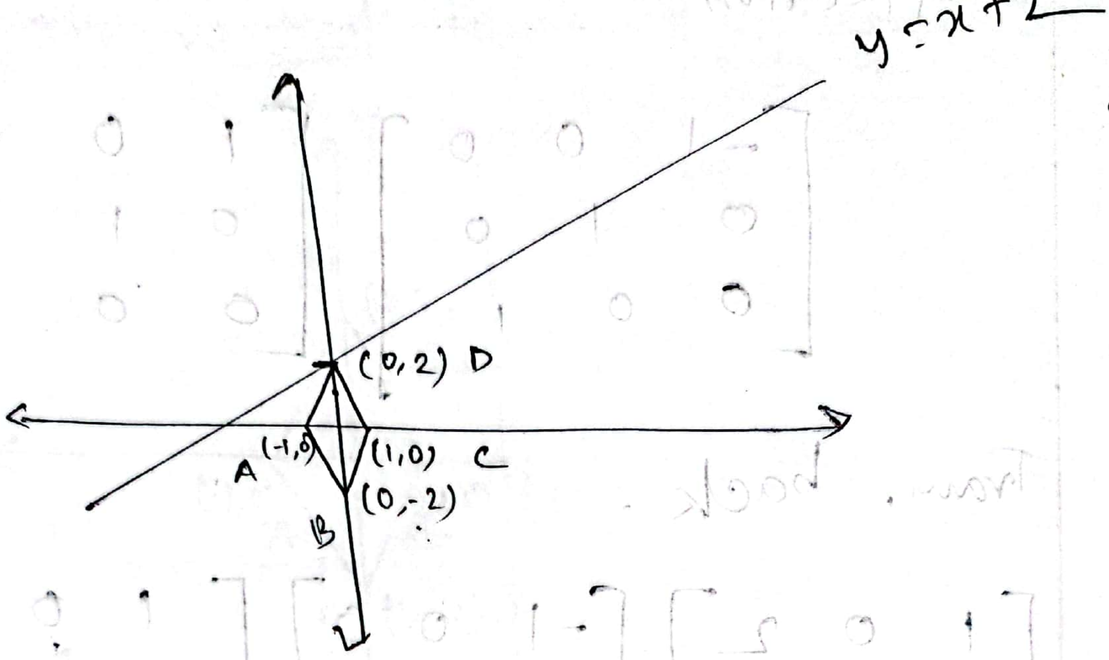
Reflection

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Trans. back.

$$\begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(c)




Here,

$m = 1$

$\therefore \theta = 45°$

Sol$^n$:

Translation

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

***। হয়তো Math next CT তে এরকম line এর eqn change করে দিবে, Negative slope দিয়ে Practice করে নিবো.

Rotate

$$\begin{bmatrix} \cos 45^\circ & \sin 45^\circ & 0 \\ -\sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

Reflection

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotate back & then again translation

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & -0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

(Ch-1)

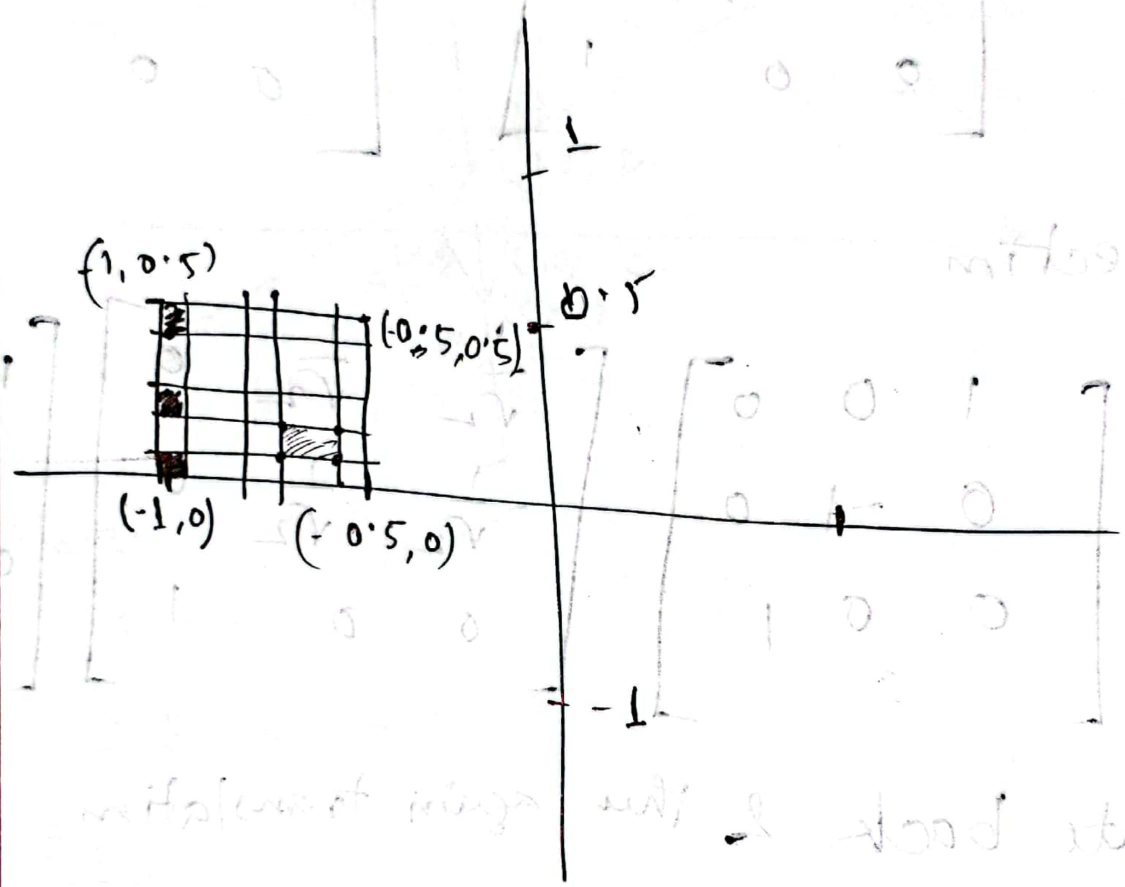
**CT-02 : "2D + 3D Transformation"**

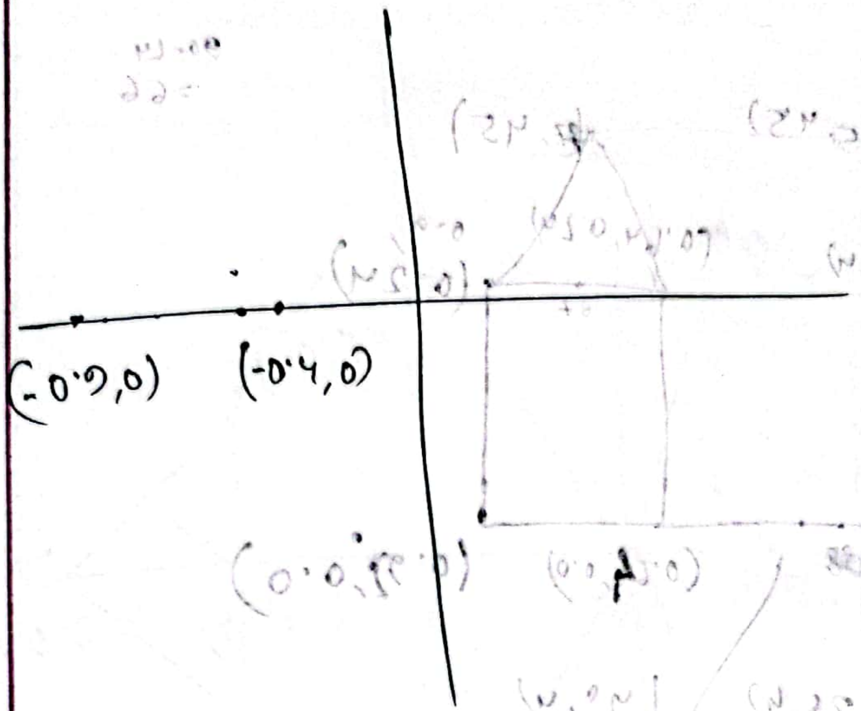
RH

28/5/22

# Lab-02

## Task 1:





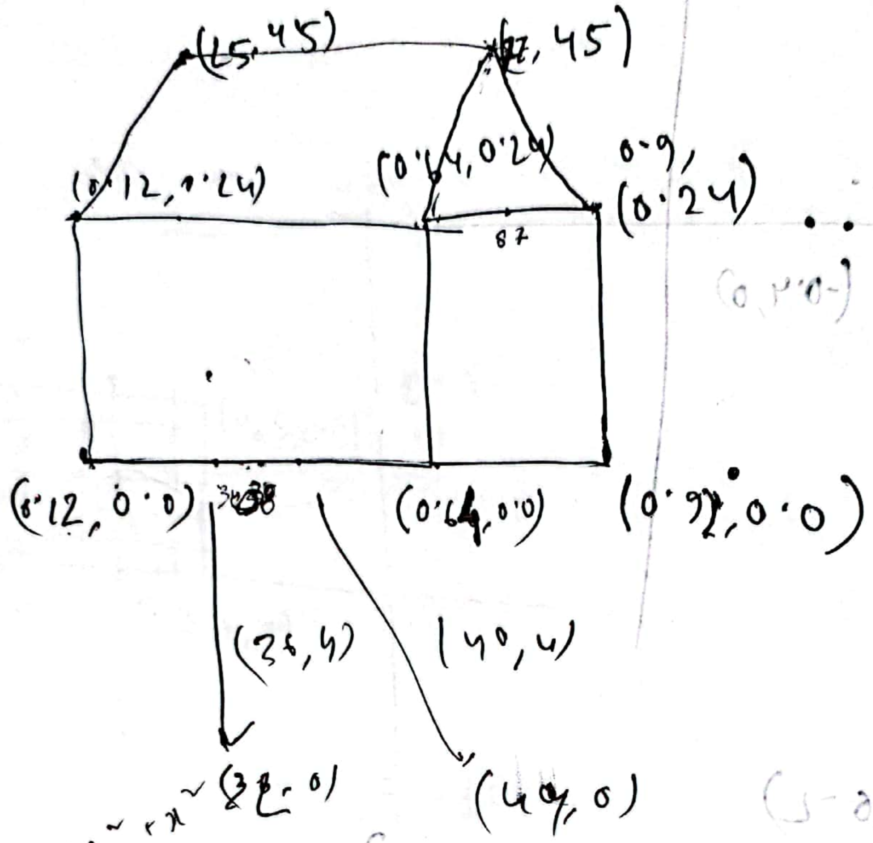

(-0.8, 0.1) (-0.6, 0.2)

(-0.7, 0.1) (-0.5, 0.2)

(-0.8, 0.1) (-0.7, 0.1) (0.6, 0.1) (-0.5, 0.1)

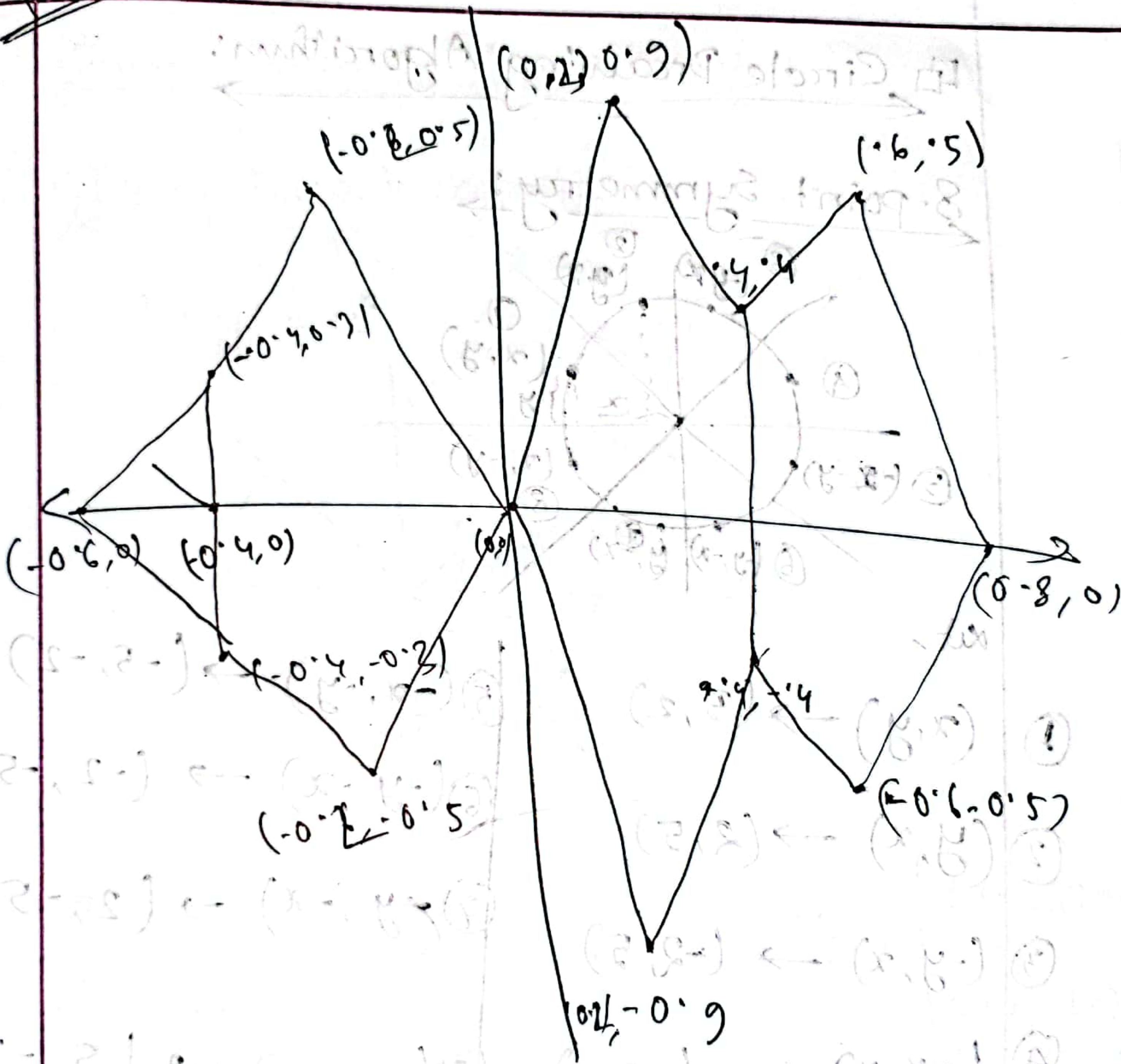
$90 - 72 = 18$

$90 - 24$
$= 66$




$26^2 = 13^2 + x^2$

Task: 2



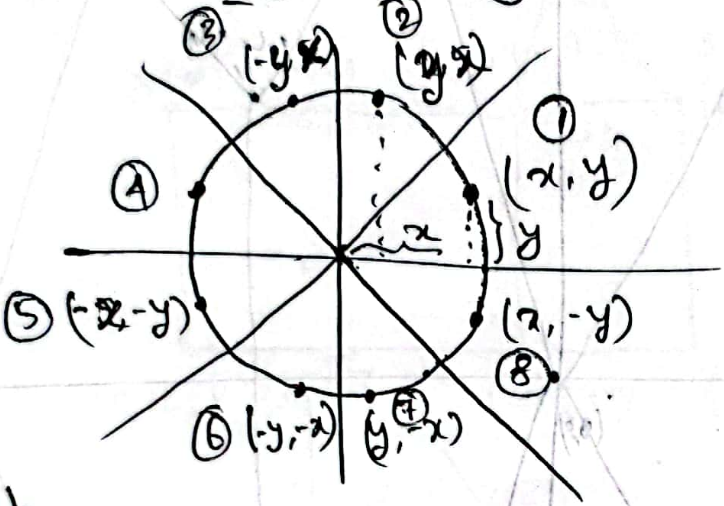
MZI

28/5/22

## Class - 13+14

### Circle Drawing Algorithm:

**8-point Symmetry:**




Let,

① $(x,y) \rightarrow (5,2)$

② $(y,x) \rightarrow (2,5)$

③ $(-y,x) \rightarrow (-2,5)$

④ $(-x,y) \rightarrow (-5,2)$

⑤ $(-x,-y) \rightarrow (-5,-2)$

⑥ $(-y,-x) \rightarrow (-2,-5)$

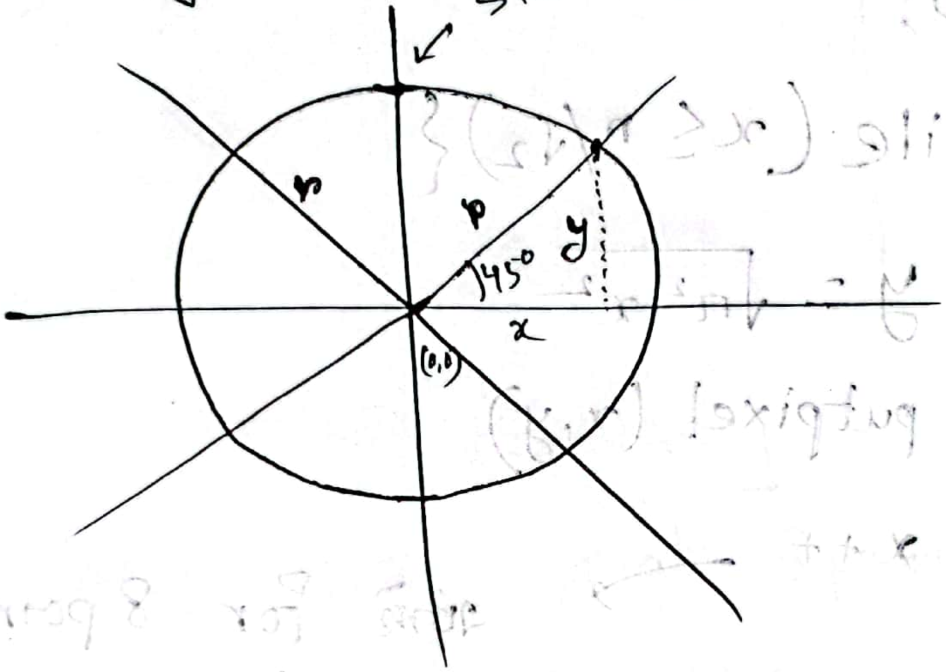
⑦ $(y,-x) \rightarrow (2,-5)$

⑧ $(x,-y) \rightarrow (5,-2)$

# Circle drawing Algorithm:

## i) Using Direct equation:

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




- $x \rightarrow$ increment (range 0 to $r/\sqrt{2}$)
- $y \rightarrow \sqrt{r^2 - x^2}$

**Calculation of range (x):**

$\cos 45° = x/r$

$\therefore x = r/\sqrt{2}$

## Algorithm:

i) input : $r$

ii) Output : a circle based on $r$.

iii) $x = 0;$

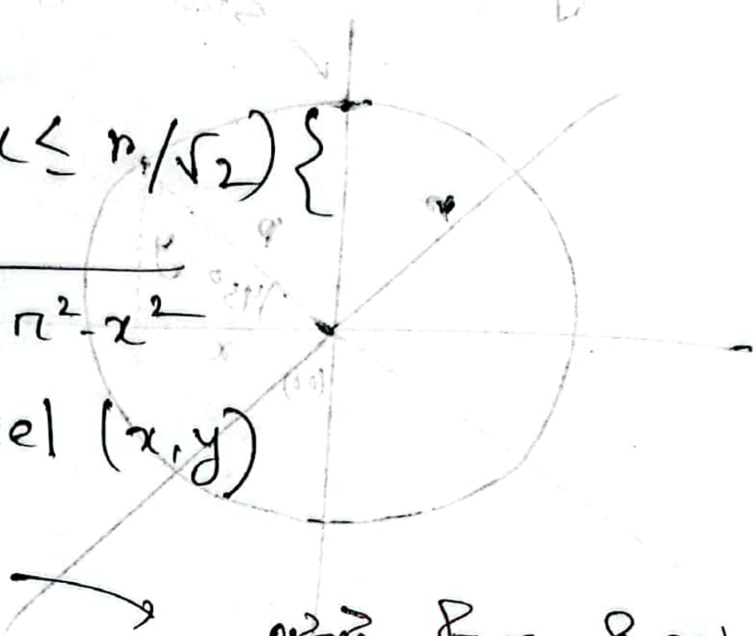
iv) while $(x \leq r/\sqrt{2})$ {

$$y = \sqrt{r^2 - x^2}$$

putpixel $(x, y)$

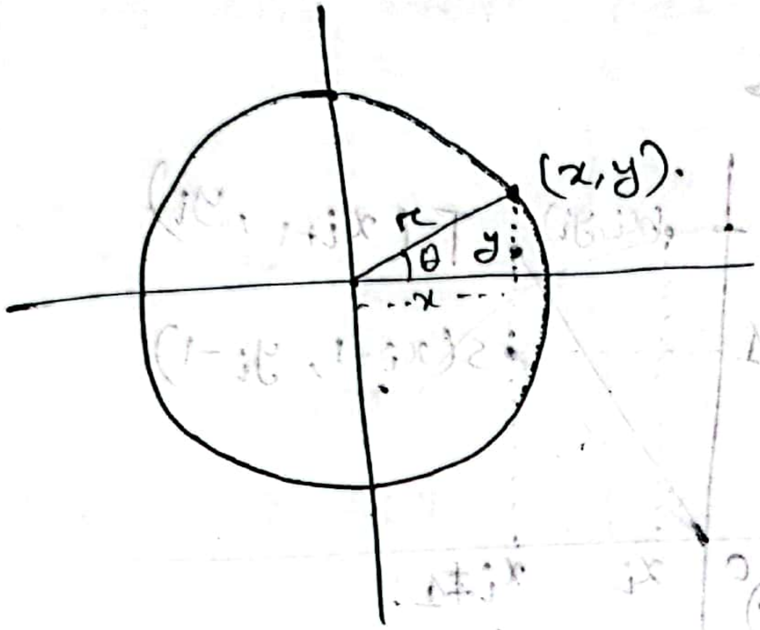
$x{+}{+}$

}



এটাই for 8 point symmetry এর জন্য, এখানে Just 8 point এর একটা পর্যন্ত আঁকা হইবে.

## ii) Using Trigonometric Function:




$x = r\cos\theta$

$y = r\sin\theta$

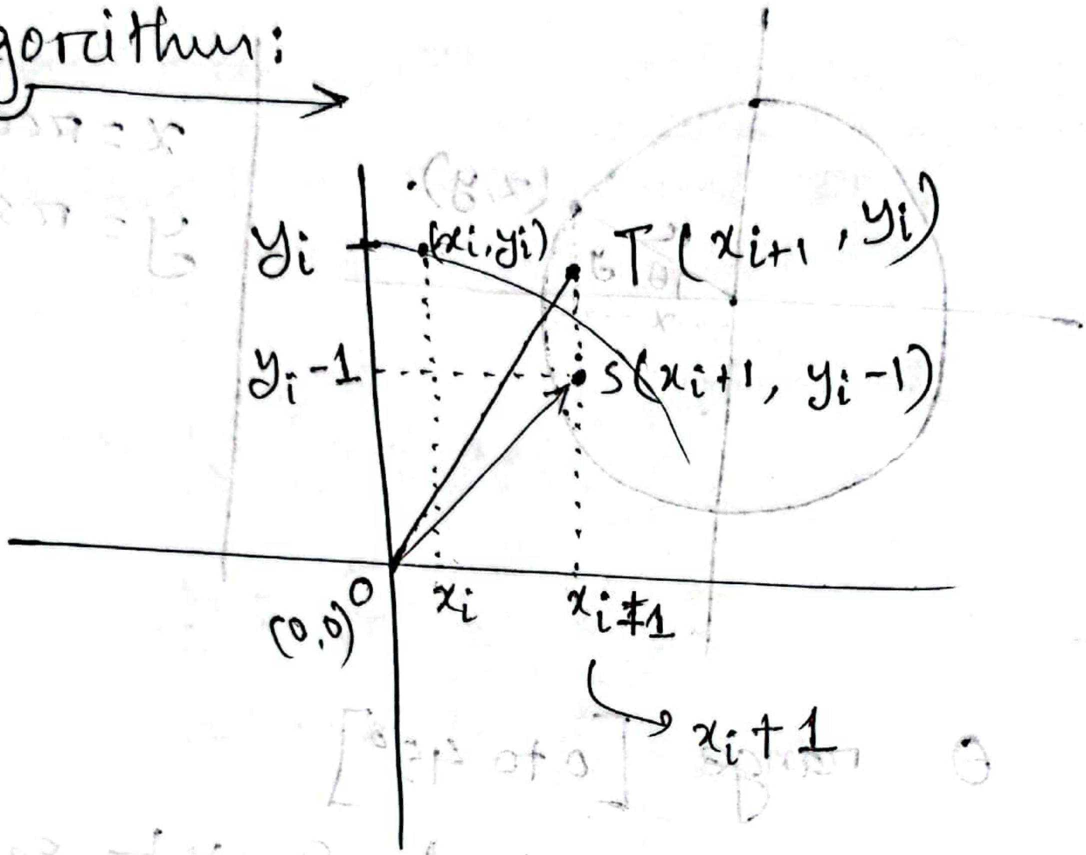
$\theta$ range [0 to 45°]

same code. just 8 point symmetry use করে বের করা যায়।

এটার complexity বেশি as এখানে x, y বের করতে multiplication + trigonometric func. দুইটাই used হচ্ছে।

## iii) Using Bresenham Circle Drawing

**Algorithm:**




→ $x_i, y_i$ আমরা Latest draw করেছি, পরের Point টা either S or T. Now, to decide which one of S, T?

(Circle এর Surface smooth হবে না as সব pixel regular circle এর ওপরে তো পড়বে না).

→ S or T এর যেই point টা circumference এর কাছাকাছি থাকবে সেটাই choose হবে.

**T এর জন্য:**

$y_{i+1} \rightarrow y_i$

$x_i+1 \rightarrow x_i+1$

**S এর জন্য:**

$y_{i+1} \rightarrow y_i - 1$

$x_i+1 \rightarrow x_i+1$

$$OT = \sqrt{y_i^2 + (x_i+1)^2}$$

$$OS = \sqrt{(y_i-1)^2 + (x_i+1)^2}$$

Now, let,

$d(T) = OT^2 - r^2$ [always +ve as $OT > r$]

$\therefore d(T) = (x_i+1)^2 + y_i^2 - r^2$

$d(S) = OS^2 - r^2$ [always -ve as $OS < r$]

$\therefore d(S) = (x_i+1)^2 + (y_i-1)^2 - r^2$

Decision variable

$d_i = d(T) + d(S)$

$\therefore d_i = 2(x_i+1)^2 + y_i^2 + (y_i-1)^2 - 2r^2$ ——— (I)

So,

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

Now,

$d_{i+1} - d_i = 2\{(x_{i+1}+1)^2 - (x_i+1)^2\}$
$+ (y_{i+1}^2 - y_i^2)$
$+ \{(y_{i+1}-1)^2 - (y_i-1)^2\}$
$+ 0$

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

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

$$1)^2 - 2r^2$$ (1)

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

$$-1)^2\}$$

$$y_i-1)^2\}$$

$$+1)^2\}$$

$$+\{(y_{i+1}-1)^2-(y_i-1)^2\}$$

$$\}+(y_{i+1}^2-y_i^2)$$

$$-1)^2\}$$

(N.V.I → Derivation of all algo.)

if, $d_i < 0$,

so, $|d(T)| < |d(S)|$

then, T point circle থেকে কাছাকাছি,

so, $(y_{i+1} = y_i)$

if, $d_i > 0$,

so, $|d(T)| \geq |d(S)|$

then, S point is selected.

so $(y_{i+1} = y_i - 1)$

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

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

Case-I:

For T point :

$(d_i < 0)$

$y_{i+1} \rightarrow y_i$

From eq$^n$ (iii),

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

$$\therefore d_{i+1} = d_i + 4x_i + 6 \text{ ——— (IV)}$$

## Case - II :

For S point :

$(d_i \geq 0)$

$y_{i+1} \longrightarrow y_i - 1$

From eq^n (iii),

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

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

$$= d_i + 4x_i + 6 + 1 - 2y_i - 2y_i + 3$$

$$= d_i + 4x_i - 4y_i + 10$$

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

(iv)

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

From eq$^n$ (1),

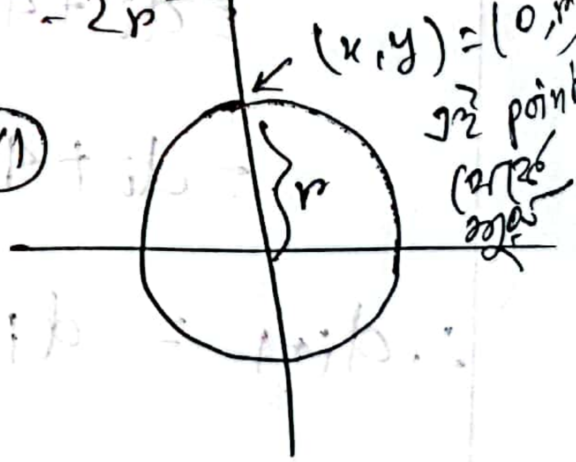
$$d_i = 2(x_i+1)^2 + (y_i - 1)^2 + y_i^2 - 2r^2$$

$$d_1 = 2(x_1+1)^2 + (y_1 - 1)^2 + y_1^2 - 2r^2$$

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

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

$$\therefore d_1 = 3 - 2r \text{ ——— (VI)}$$



এখন, $d_1$ থেকে $d_{i+1}$ এর মান দিয়ে 45° পর্যন্ত যাব each portion থেকে of the 8 portions of circle.

## Algorithm:

i) <u>Input</u>: '$r$'

<u>Output</u>: A circle

<u>Method</u>:

i) $x = 0,\ y = r,\ d = 3 - 2r$

ii) putpixel $(x, y)$

iii) while $(x \le r/\sqrt{2})$ {

$x{+}{+}$

if $(d < 0)$ {

$d = d + 4x + 6$

}

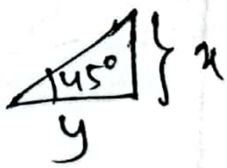
else {

$d = d + 4(x - y) + 10$

$y = y - 1;$

}

অথবা condition দিতে $x \le y$ ও যেখানে $45°$ তে $x = y$

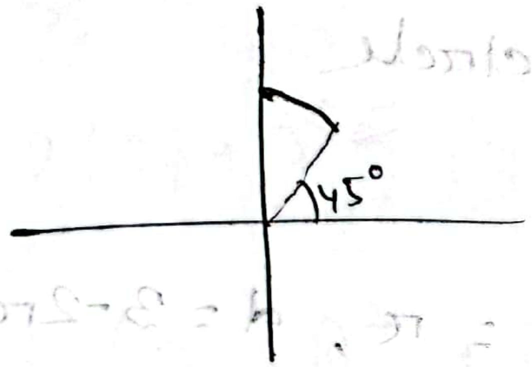


putpixel (x,y)

}

এই algorithm এ arc draw হবে 45° পর্য

যেমন :




Algo. তে

(ii আর iii) এ putpixel (y,x) /

দেয়া যাবে →

(-y,x) /

(-x,y) /

(-x,-y) /

(-y,-x) /

(y,-x) /

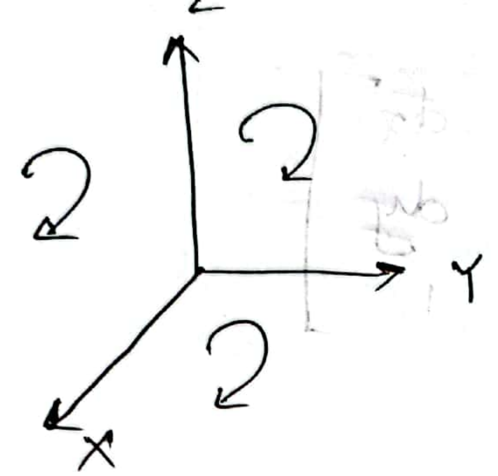
(x,-y)

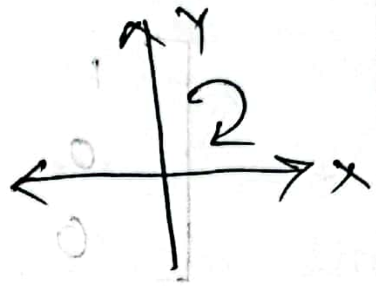
Class -15

29/5/22

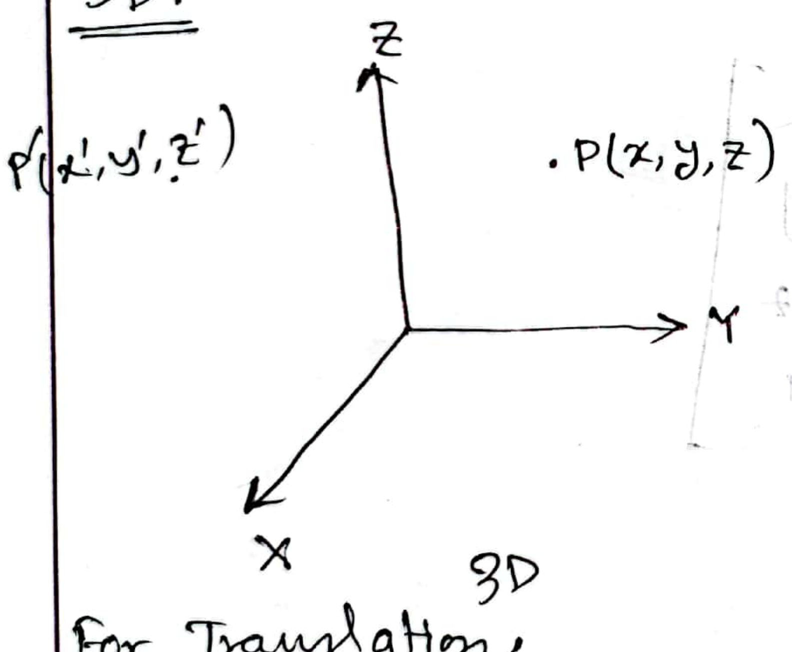
2D coordinate → 2 dimensions, 1 plane

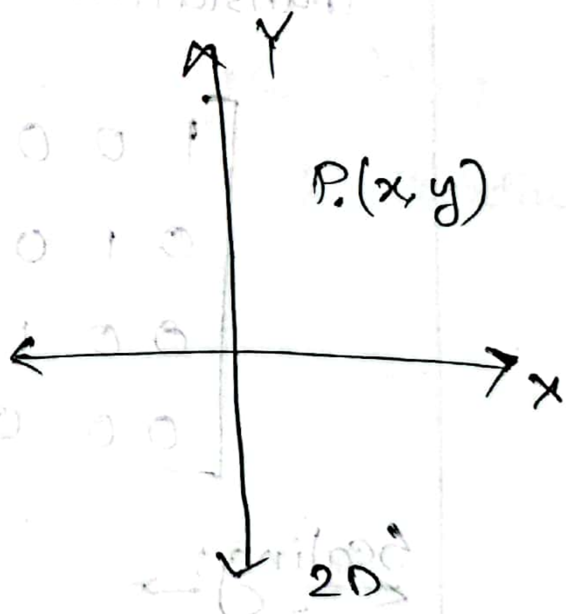
3D " " 3 " , 3 planes




## 3D:




For Translation,

$x' = x + dx$

$y' = y + dy$

$z' = z + dz$

For Translation:

$x' = x + dx$

$y' = y + dy$

At 2D, Trans. Mat. at hom. coordinate was –

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

For 3D,

Translation Matrix will be-

$$\begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling:

$$x' = S_x . x$$

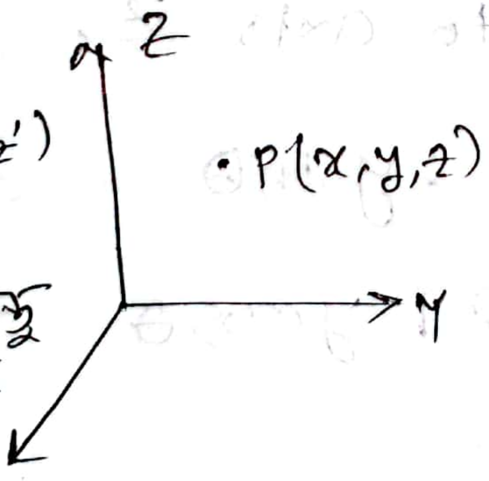
$$y' = S_y . y$$

$$z' = S_z . z$$

$$\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

→ 3D তে rotation angle + rotation direction জানা লাগবে।

→ x এর সাপেক্ষে rotation মানে rotation হবে YZ plane এ। তখন x এর coordinate change হবে না।




P'(x',y',z')

এইবিন্দু translate করে YZ plane এ আসে, XZ এ না

With respect to axis X,

$x' = x$

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

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

এগুলো মুখস্থ রাখতে হবে.

With respect to axis Y,

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

$y' = y$

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

With respect to axis Z,

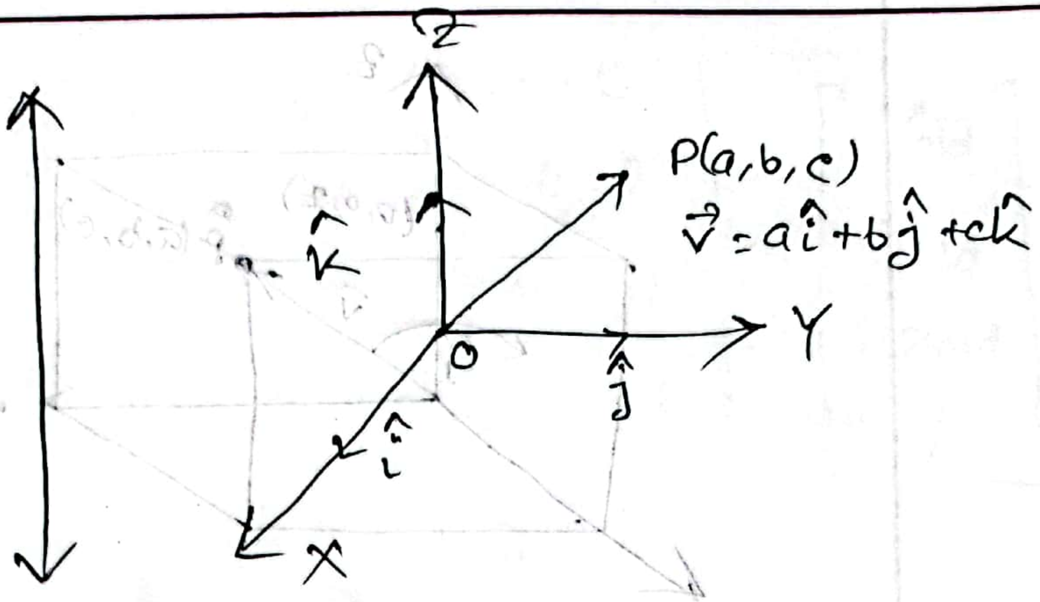
$x' = x\cos\theta - y\sin\theta$

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

$z' = z$

## Tilting :

(*)

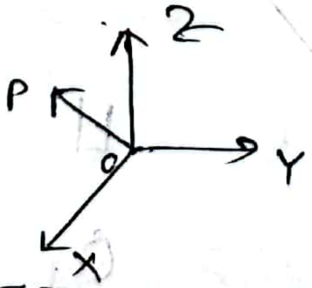



Math: এমনভাবে p কে translate করতে হবে যাতে $\vec{OP}$, Z axis এর সাথে aligned হয়।

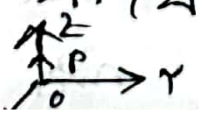
এখানে, দুইবার rotate করতে হবে,

[ultimately, Z এর সাপেক্ষে rotate করতেছি tilting এ ]

1) First এ X এর সাপেক্ষে rotate করবো, এরকম হবে।



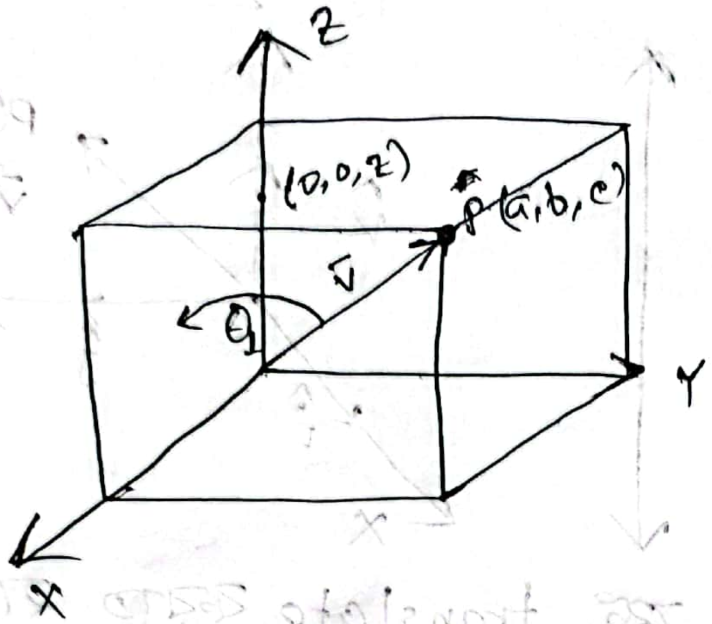

2) Then, Y এর সাপেক্ষে rotate করবো, তখন Z এর সাথে aligned হয়ে যাবে, like -



RH

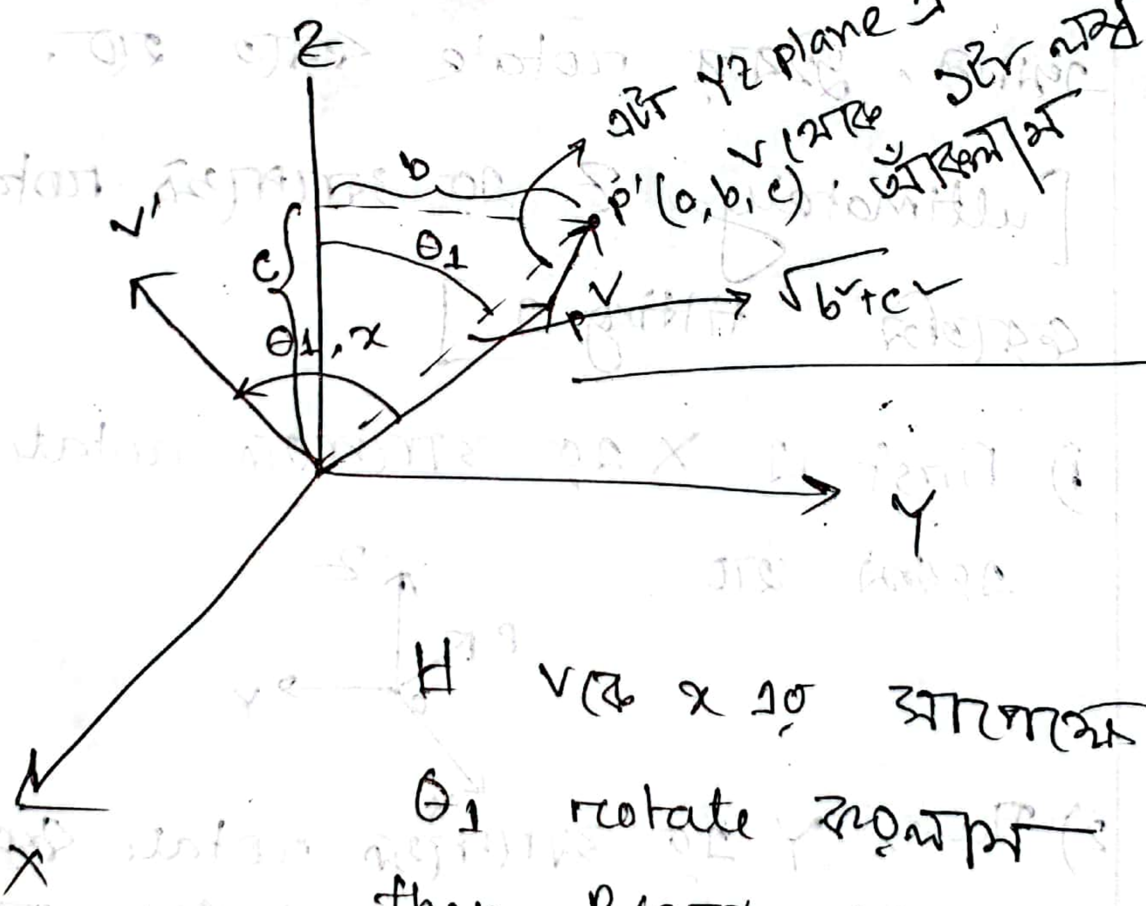
# Class · 15

31/5/22




$\vec{v} = a\hat{i} + b\hat{j} + c\hat{k}$

$\vec{v}$ কে z axis বরাবর align করতে হবে.




∴ V কে x এর সাপেক্ষে $\theta_1$ rotate করলাম

then, P থেকে YZ এর উপর লম্ব আঁকলাম

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

$$\text{so, } \cos\theta = \frac{c}{\sqrt{b^2+c^2}}$$

$$\sin\theta = \frac{b}{\sqrt{b^2+c^2}}$$

এইবার new coordinate calculate করি

তারপর এখানে বসাবো

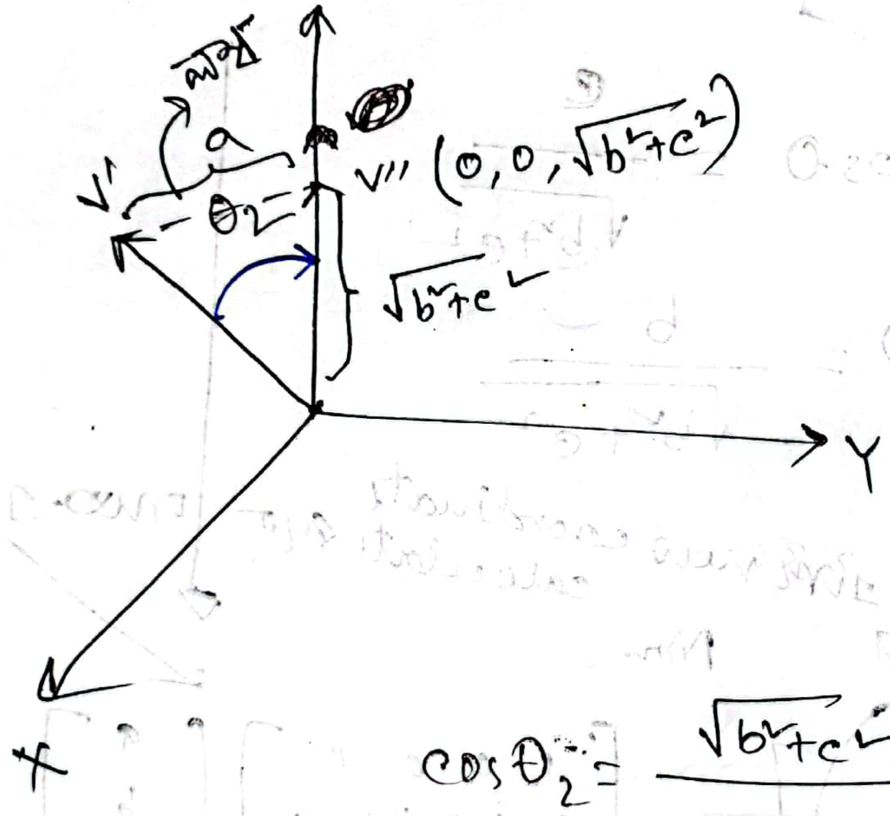
Now,

$$\begin{bmatrix} a \\ 0 \\ \sqrt{b^2+c^2} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 & 0 \\ 0 & \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix}$$

∴ V'-এ যেই point আছে তার coordinate হল $(a, 0, \sqrt{b^2+c^2})$

→ V' টা আছে XZ plane এ

এখন এটাকে y axis এর সাপেক্ষে $\theta_2$ rotate করালে, z axis এ চলে আসবে তারপর z axis এর দিকে V'V'' লম্ব আঁকলাম




$$\cos\theta_2 = \frac{\sqrt{b^2+c^2}}{\sqrt{a^2+b^2+c^2}}$$

$$\sin\theta_2 = \frac{a}{\sqrt{a^2+b^2+c^2}}$$

So,

$$A_{V,K} = R_{-\theta_2, y} \cdot R_{\theta_1, x}$$

Since,

$|V| = \sqrt{a^2+b^2+c^2}$ & $\lambda = \sqrt{b^2+c^2}$ we find

$$A_{V,K} = R_{-\theta_2, y} \cdot R_{\theta_1, x}$$

$$= \begin{bmatrix} \frac{\lambda}{|V|} & \frac{-ab}{\lambda|V|} & \frac{-ac}{\lambda|V|} & 0 \\ \frac{-ab}{\lambda|V|} & \frac{c}{\lambda} & \frac{b}{|V|} & 0 \\ \frac{-ac}{\lambda|V|} & -b/\lambda & \frac{c}{|V|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

এটা Math এর জন্য মুখস্থ রাখা লাগবে।

এটা ভুল আছে Slide or অন্য Class Note এ ঠিকটা আছে।

এখন,

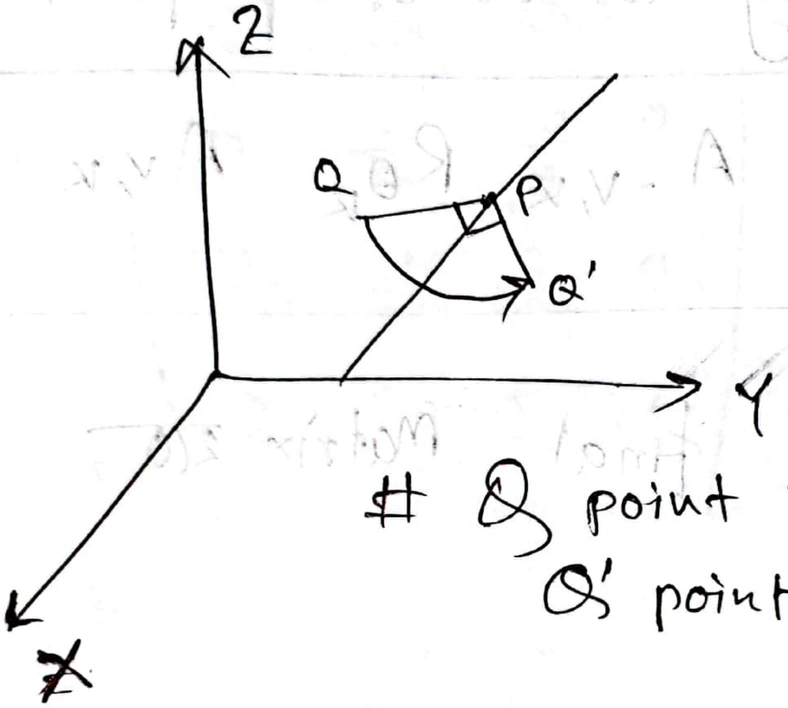
same জিনিস যদি z axis থেকে আস... p' point এ নিয়ে যেতে বলে?

তাহলে, order Reverse হবে
angle change হবে,

$$\cancel{\left(R_{-\theta_1, x}, R_{\theta_2, y}\right)}$$

$$\left(R_{-\theta_1, x} \cdot R_{\theta_2, y}\right)$$

Now, এতক্ষণ তো axis এর সাপেক্ষে rotation করলাম, এখন যেকোনো line এর সাপেক্ষে rotation করব।

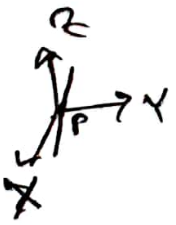



# Q point কে rotate করে Q' point এ নিব,

Firstly take P to origin,

$T_{-P}$

then,




z axis এ নিয়ে আসি

$A_{v,k} \, T_{-P}$

then, z এ নিয়ে rotate

$R_{\theta k} \, A_{v,k} \, T_{-P}$

then, আবার z axis থেকে সরাও

$$A_{-v,k} R_{\theta_k} A_{v,k} T_{-p}$$

then, finally rotated হবে p point এ আবার

$$\alpha \quad \boxed{T_p A_{-v,k} R_{\theta_k} A_{v,k} T_{-p}}$$

এটাই final Matrix হবে,

RH

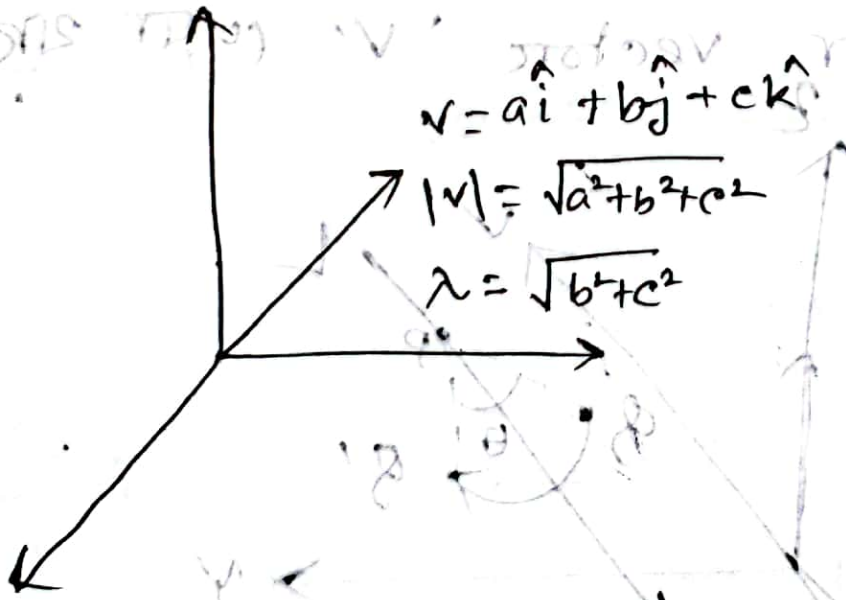
# Class-17

7/6/22

$$A_v = R_{-\theta_2 1} \cdot R_{\theta_2 1}$$

কোনটাকে কোনভাবে যেকোনো z axis এ নিতে use করা হয়.

$$= \begin{pmatrix} \frac{\lambda}{|v|} & -\frac{ab}{\lambda|v|} & -\frac{ac}{\lambda|v|} & 0 \\ 0 & \frac{c}{\lambda} & \frac{-b}{\lambda} & 0 \\ \frac{a}{|v|} & \frac{b}{|v|} & \frac{c}{|v|} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$$v = a\hat{i} + b\hat{j} + c\hat{k}$$

$$|v| = \sqrt{a^2+b^2+c^2}$$

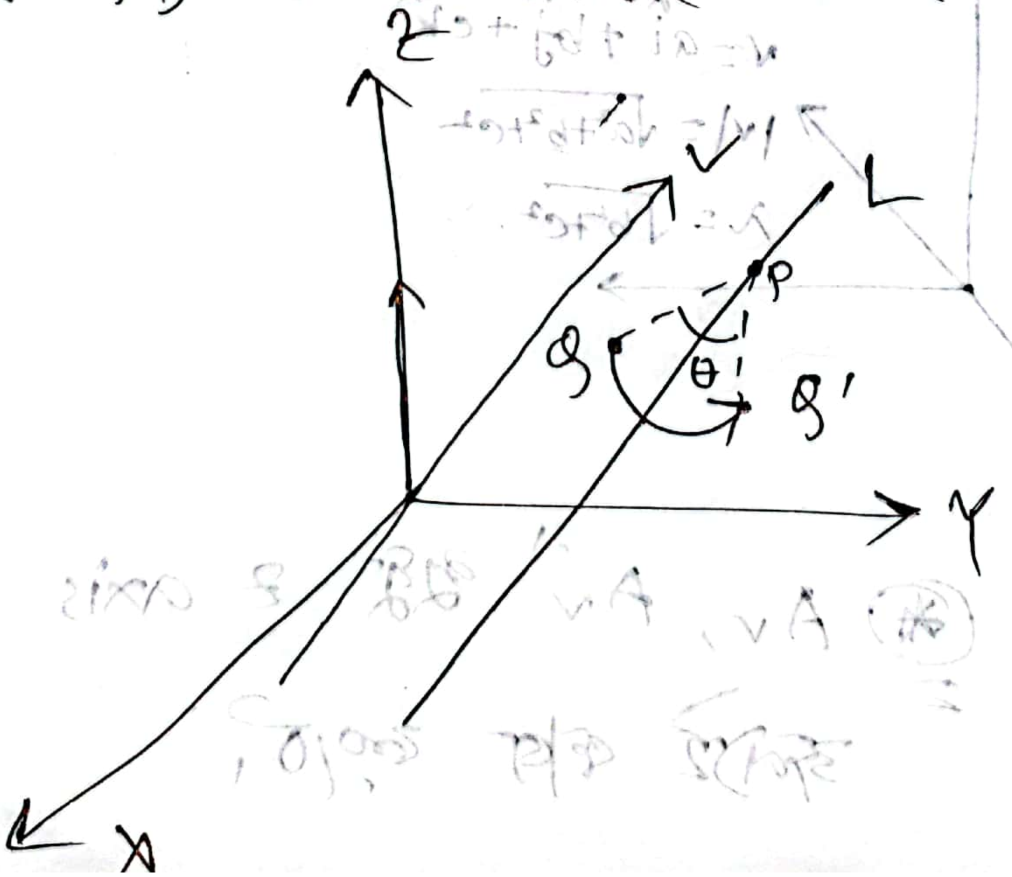
$$\lambda = \sqrt{b^2+c^2}$$

* $A_v$, $A_{v'}$ দুটি z axis এর অ্যালাইন করা হলো।

z axis থেকে আবার আগের অবস্থায় back করো.

$$A_V^{-1} = \begin{bmatrix} \frac{\lambda}{|V|} & 0 & \frac{a}{|V|} & 0 \\ \frac{-ab}{\lambda|V|} & \frac{c}{\lambda} & \frac{b}{|V|} & 0 \\ \frac{-ac}{\lambda|V|} & -\frac{b}{\lambda} & \frac{c}{|V|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#) Ques. এ Line ঘুরানোর কথা বলে দিলে Line এর একটি point বলে দিবে। এবং Line এর Direction দেয়া থাকবে। Direction এর জন্য একটি vector 'V' দেয়া থাকবে.



Total Rotation, $R_{\theta,L}$ → L line এর সাপেক্ষে $\theta^\circ$ তে ~~translation~~ Rotation

Sol$^n$ Step 1:

p কে origin এ নিয়ে যাই Translate করে.

$T_{-p}$

Step 2:

Then z axis এর উপর আনব,

So,

$$A_v \cdot T_{-p}$$

Step 3:

k axis সাপেক্ষে rotate করব.

$$R_{\theta,k} \, A_{v,k} \, T_{-p}$$

Step 4:

z axis এর ঐ অবস্থান থেকে সরাব

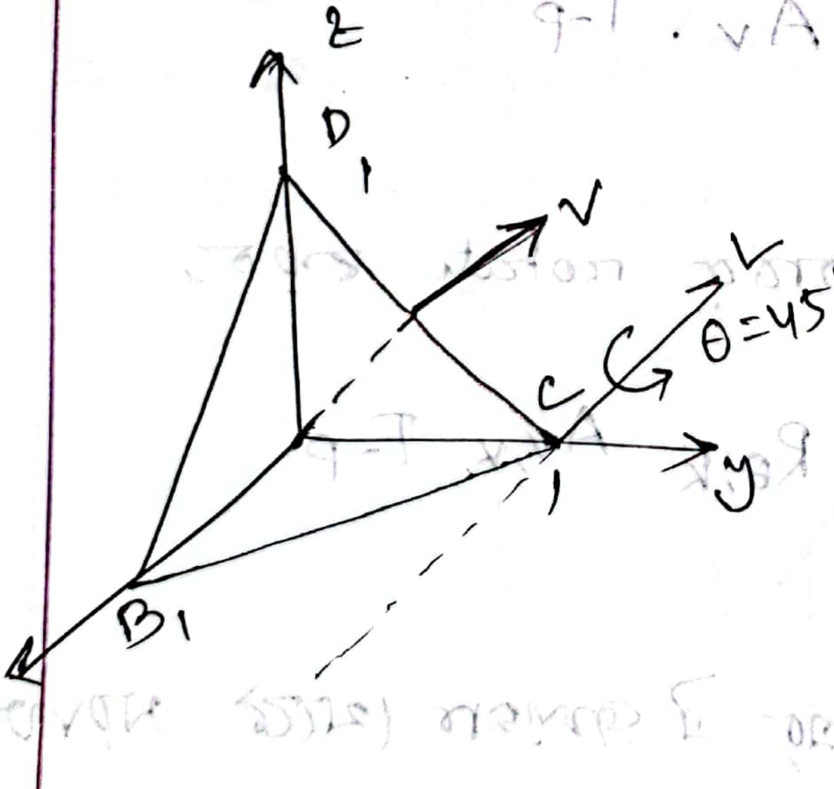
$$A_{v,k}^{-1} \, R_{\theta,k} \, A_{v,k} \, T_{-p}$$

## Step 5:

Original ~~po~~ line এর সাপেক্ষে Translation করলে যেখানে ছিলো point এ সেখানে দিবো,

$$T_p \, A_{v,k}^{-1} \, R_{\theta,k} \, A_{v,k} \, T_{-p}$$

## Math Problem:



Soln:

**Step 1:**

Origin এ নিয়ে Cpoint হবে

$$T_P = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Step 2:**

z axis এর উপর নিয়ে.

$$A_v \cdot T_P = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{2}} & -\frac{0.1}{\sqrt{2}\sqrt{2}} & -\frac{0.1}{\sqrt{2}\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{0}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$|v| = \sqrt{a^2+b^2+c^2}$
$= \sqrt{0^2+1^2+1^2} = \sqrt{2}$

$\lambda = \sqrt{b^2+c^2}$

$= \sqrt{1^2+1^2}$

$= \sqrt{2}$.

Given,

$\vec{V} = \hat{j} + \hat{k}$ or $V = J + K$

so, $\vec{v} = a\hat{i} + b\hat{j} + c\hat{k}$

where, $a = 0$, $b = 1$, $c = 1$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\therefore A_v \cdot T_p = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \sqrt{2} \\ 0 & 0 & 0 & \end{bmatrix}$$

step: 3 (Rotation k axis ঘোরাও)

$$R_{\theta,k}\, A_{v,k}\, T_{-p} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \sqrt{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

এখানে $\theta = 45^\circ$,

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \sqrt{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

step: 4

$$A_{v,k}^{-1}\, R_{\theta,k}\, A_{v,k}\, T_{-p} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \; \\ \; \\ \; \\ \; \end{bmatrix}$$

**Step 5:**

$$R_{\theta,L} = T_p \, A_{v,k} \, R_{\theta,k} \, A_v \, T_{-p}$$

$R_{\theta,L}$ হবে Rotation Matrix

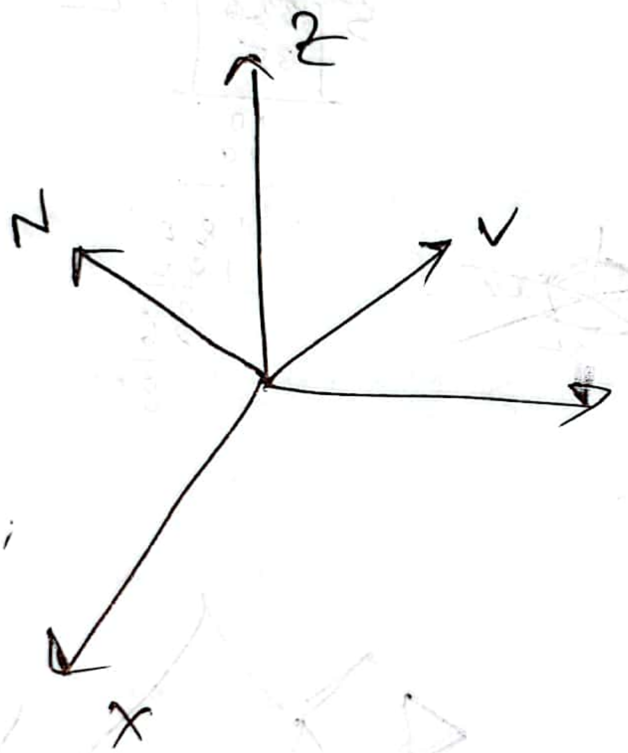
$\therefore$ Final step এ $R_{\theta,L}$ এর সাথে P point এর matrix গুণ করলেই P' পাবো,

$$P = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$\therefore R_{\theta,L} \cdot P$ is the final result.

এখানে প্রশ্ন নাই মুখস্থ, কাজ দেখতে হবে, এরকম একটা Math এ 8-10 marks থাকবে

~~[illegible]~~ Find a transformation $A_{V,N}$ which aligns a vector V with a vector N.




$$A_{V,N} = A_N^{-1} \cdot A_V$$

** CT#02 of RH sir 2D+3D এর উপর হবে.

RH

# LAB-03

18/6/22

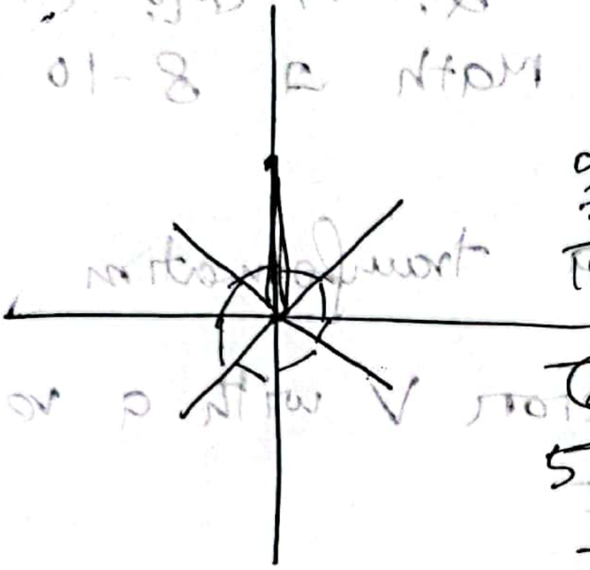
8 | 360 | 45
32
40

90
72
162
72
244

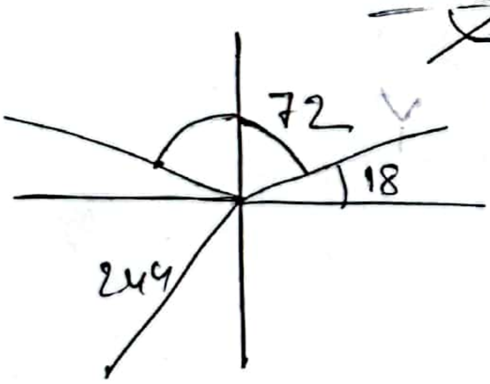
5 | 360 | 72
35
10
10

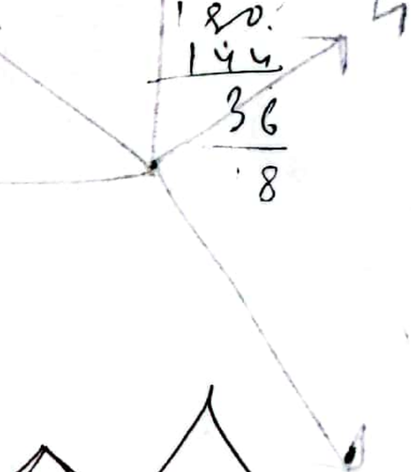
180
144
36
8

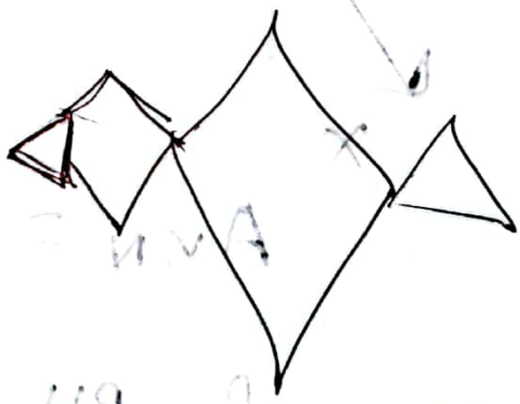
20
36
126
72
198
72
270





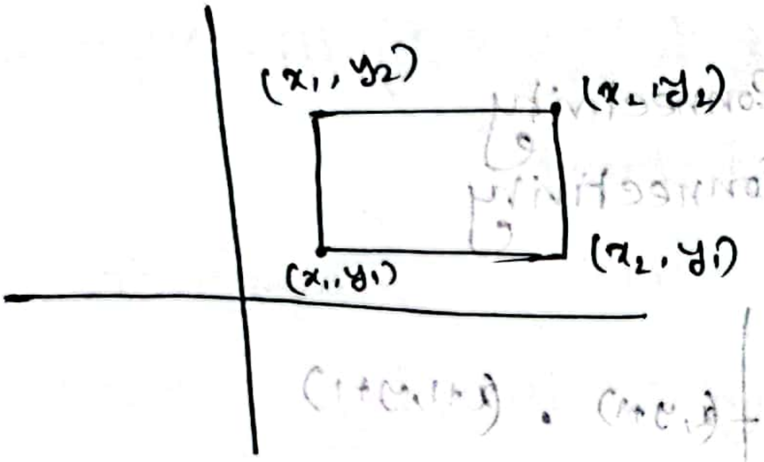


Class-18 21/6/22

# Rectangle Drawing:




Retangle $(x_1, y_1, x_2, y_2)$ {

line $(x_1, y_1, x_2, y_2)$

line ( . . . . )

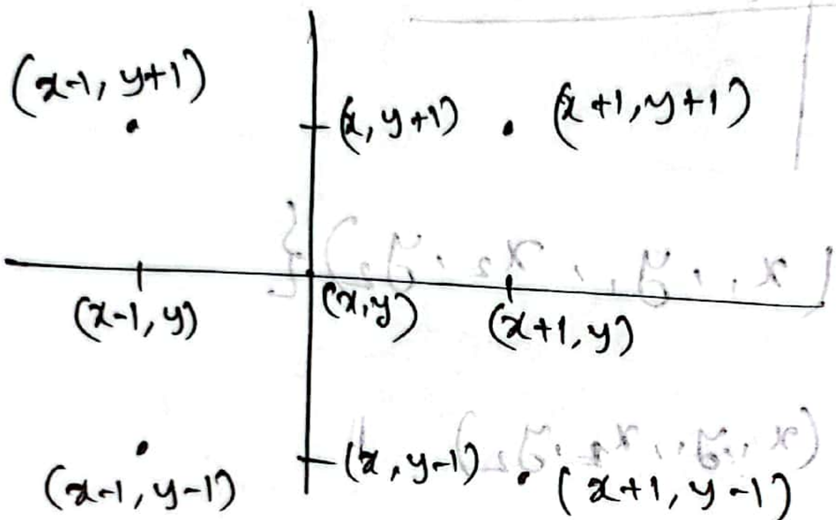
Line ( )

Line ( )

}

## Region Filling:

- 4-point Connectivity
- 8-point Connectivity




**Algo:**

1. Boundary Fill Algorithm
2. Flood Fill Algorithm

Region Filling Algos.

## Boundary fill Algorithm:

```
Boundary Fill (int x, int y, int Boundary_colour,
                              int Fill_colour) {
    int color;
    color = getpixel (x,y);
    if (color != Boundary_color && color != Fill_color)
    {
        setpixel (x, y, Fill_color);
        BoundaryFill (x+1, y, Boundary_Color,
                              Fill_Color);
            "     (x-1, y,   "   ,  " );
            "     (x, y+1,   "   ,  " );
            "     (x, y-1,   "   ,  " );
    }
}
```

Seed → Any Point inside the region/object

```
main () {

    setcolour (Blue);
    Circle (50, 50, 10);
    BoundaryFill (50, 50, Blue, Red);

}
```

Boundary color → Blue

$\underbrace{50, 50}_{\text{Seed}}$ → Seed (যেহেতু center জানা আছে, তাই দিলাম, circle এর ভিতর যেকোনো point দেয়া যায়)

## Flood Fill Algorithm:

```
Flood Fill (int x, int y, int original_color, int
                                        fill_color)
{
    int color;
    color = getpixel (x,y);
    if (color == original_color){
        set pixel (x,y, fill_color);
        Flood Fill (x+1, y, Original_color, fillColor);
           "       (x-1, y,      "      ,    "    );
           "       (x, y+1,      "      ,    "    );
           "       (x, y-1,      "      ,    "    );
    }
}
```

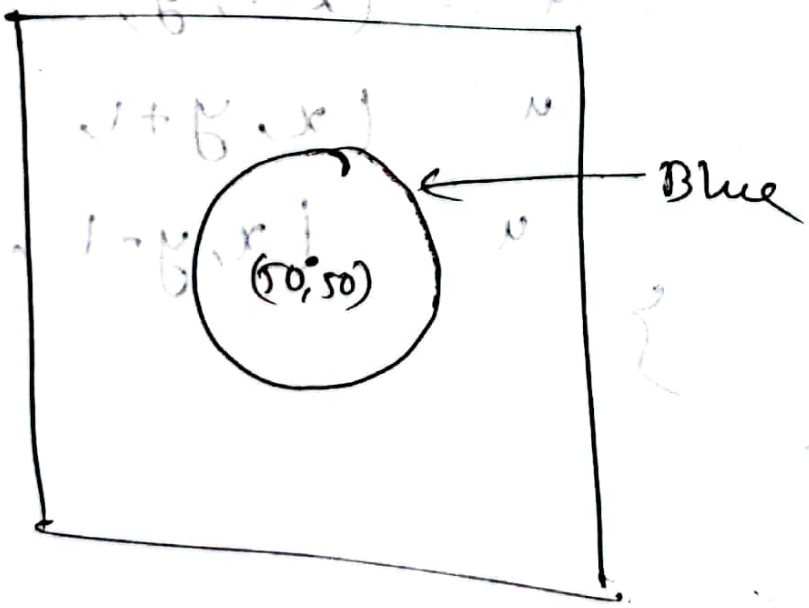
main () {

setcolour (Blue);

Circle (50,50,10); radius

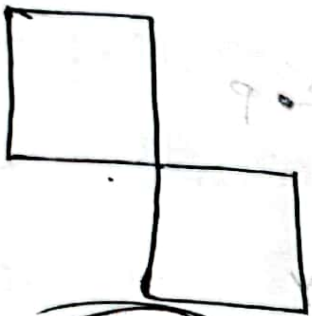
FloodFill (50,50, Black, Red);

}



## 8point vs 4 point Connectivity

→ কখনো কখনো 4-point কাজ করবে না.



← এরকম case এ 8-point Better.

H.W V.V.I

# Ques: Which region Filling algo is better?

When to use what?

CT

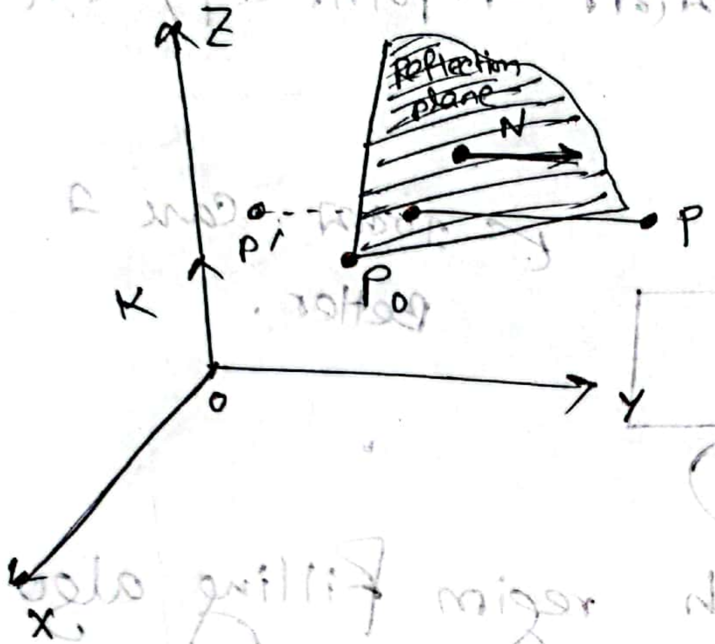
Next Saturday (Chapter-3 Full)

RH

Class-19

26/6/22

Ⓠ Find mirror reflection with respect to any arbitary plane.




1. Translate $P_0$ to origin
2. Align normal vector N along with z axis. তারপর, Reflection plane টা XY plane এর উপর বসে যাবে
3. Perform mirror reflection in xy plane.
4. Reverse step 1 & 2.

$$M_{N.P_0} = T_V^{-1} . A_N^{-1} . M . A_N . T_V$$

(*) Find the matrix for mirror reflection with respect to the plane passing through the origin & having a normal vector whose direction is $\vec{N} = \hat{i} + \hat{j} + \hat{k}$

Soln: এখানে, origin এ আছে already so,

$$M_{N.P_0} = A_N^{-1} . M . A_N$$

Here,

$$A_N = \begin{pmatrix} \frac{\lambda}{|v|} & \frac{-ab}{\lambda|v|} & -\frac{ac}{\lambda|v|} & 0 \\ 0 & \frac{c}{\lambda} & \frac{-b}{\lambda} & 0 \\ \frac{a}{|v|} & \frac{b}{|v|} & \frac{c}{|v|} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\vec{N} = \hat{i} + \hat{j} + \hat{k}$$

$$|v| = |N| = \sqrt{1^2+1^2+1^2} = \sqrt{3}$$

$$\lambda = \sqrt{1^2+1^2} = \sqrt{2}$$

$$A_N = \begin{pmatrix} \sqrt{\frac{2}{3}} & \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{6}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Reflection Matrix -

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

$$A_N^{-1} = \begin{pmatrix} \frac{\sqrt{2}}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} & 0 \\ -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & 0 \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$\therefore A_N^{-1} . M . A_N \longrightarrow$ এটিই Rotation Matrix.

exam এ object দেয়া থাকবে with coordinates.
দুই rotation Matrix দিয়ে ঐ coordinate গুলোর
গুণ দিলে Final reflected matrix পাবো।

(প্র) Find a transformation $A_{V,N}$ which aligns a vector $V = \hat{i}+\hat{j}+\hat{k}$ with a vector $N = 2\hat{i}-\hat{j}-\hat{k}$

Sol$^{n}$:

Firstly, V কে z axis এর উপর নিতে হবে then, z axis থেকে N এর উপর নিবো,

① $A_V$ দিয়ে z axis এ যাবো

② Then $A_V^{-1}$ দিয়ে V তে back না করে $A_N^{-1}$ দিয়ে N এ back করবো. as আমাদের N এ align করতে হবে।

So,

$$\boxed{A_N^{-1} \cdot A_V}$$

এটাই Rotation Matrix

2D+3D থেকেই প্রচুর math আসবে। Math গুলোতে just value, shape, object change করে দিয়ে আসবে;

$$N = 2\hat{i} - \hat{j} - \hat{k}$$

$$|N| = \sqrt{6}$$

$$\lambda_n = \sqrt{2}$$

$$V = 1\hat{i} + 1\hat{j} + 1\hat{k}$$

$$|V| = \sqrt{3}$$

$$\lambda_V = \sqrt{2}$$

$$A_V = \begin{pmatrix} \frac{\sqrt{2}}{\sqrt{3}} & \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{6}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_N^{-1} = \begin{pmatrix} \frac{\sqrt{2}}{\sqrt{6}} & 0 & \frac{2}{\sqrt{6}} & 0 \\ \frac{2}{\sqrt{12}} & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} & 0 \\ \frac{2}{\sqrt{12}} & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Class - 20 28/6/22

## OpenGL and Projection:

### OpenGL Rendering Pipeline:

→ Object create করা থেকে display তে show করা পর্যন্ত যা Process গুলোর combo. হলো Rendering Pipeline.

Modeling ⇒ View selection ⇒ Perspective Division ⇒ Displaying

Local Coordinate?
World Coordinate?

Local

Local

World Space

* World Space এ Shape Transformation, Scaling, Rotation etc. করা হয়।

Class: 21

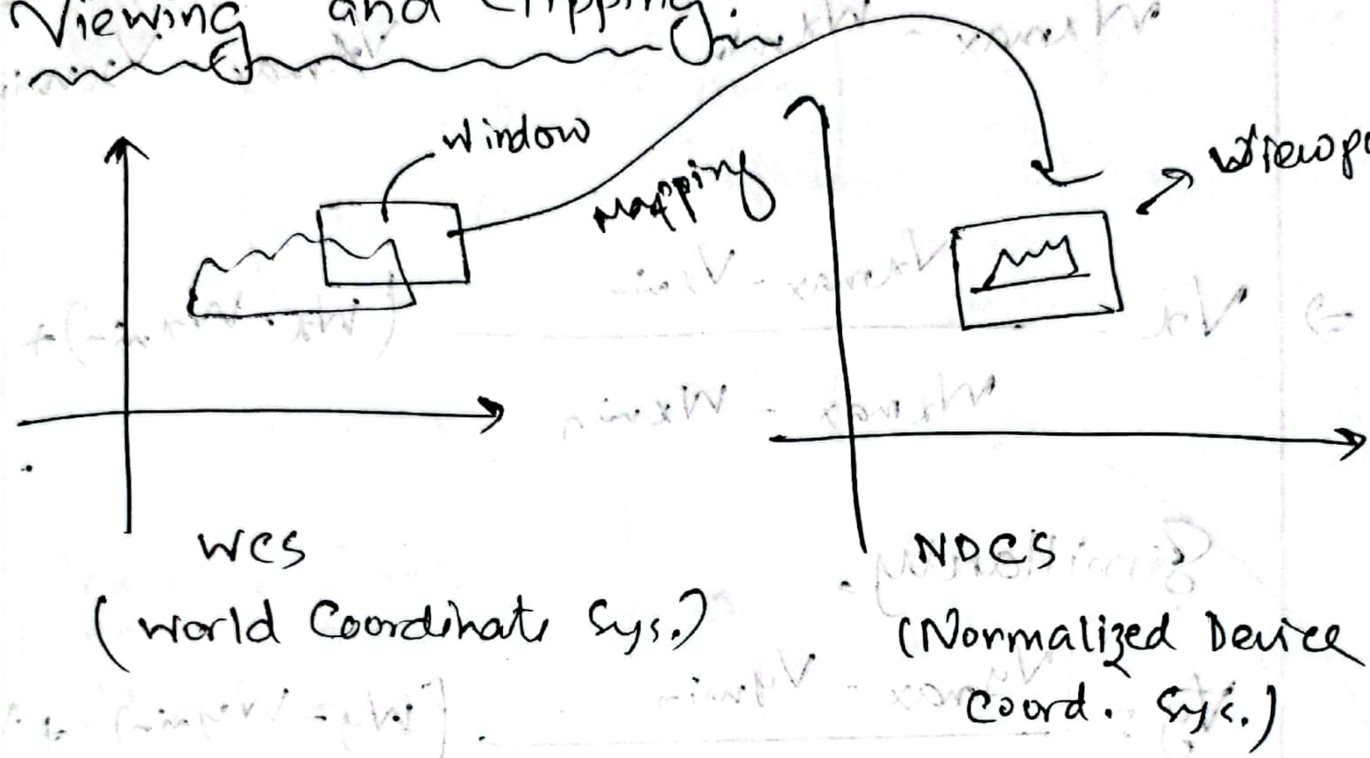
17/7/22

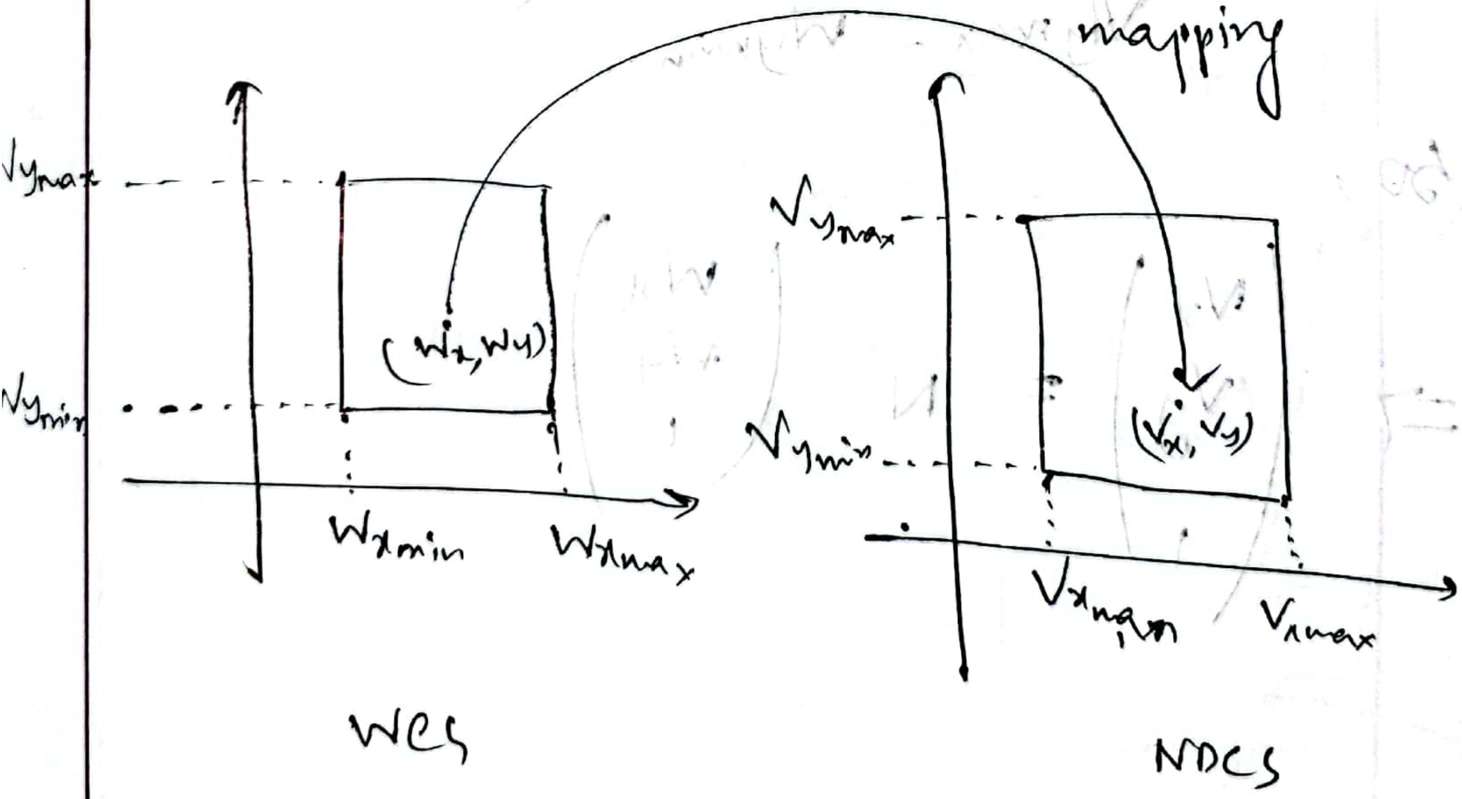
* OpenGL
  - Descriptive Ques. আসবে।
* Perspective Projection
  - Length, Width বুঝা যায় না।
* Parallel Projection
  - বুঝা যায়
* Types of Projection
  - চিত্র দেখে জানতে হবে
* Vanishing Point.

Class-22 19/7/22

## Chapter-5

### Viewing and Clipping:



$$\frac{W_x - W_{xmin}}{W_{xmax} - W_{xmin}} = \frac{V_x - V_{xmin}}{V_{xmax} - V_{xmin}}$$

$$\Rightarrow V_x = \frac{V_{xmax} - V_{xmin}}{W_{xmax} - W_{xmin}} (W_x - W_{xmin}) + V_{xmin}$$

Simillarily,

$$V_y = \frac{V_{ymax} - V_{ymin}}{W_{ymax} - W_{ymin}} (W_y - W_{ymin}) + V_{ymin}$$

So,

$$\Rightarrow \begin{pmatrix} V_x \\ V_y \\ 1 \end{pmatrix} = N \begin{pmatrix} W_x \\ W_y \\ 1 \end{pmatrix}$$

When,

$$\text{mat. } 3 = \begin{pmatrix} 1 & 0 & -W_{xmin} \\ 0 & 1 & -W_{ymin} \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{mat } 2 = \begin{pmatrix} \dfrac{V_{xmax} - V_{xmin}}{W_{xmax} - W_{xmin}} & 0 & 0 \\ 0 & \dfrac{V_{ymax} - V_{ymin}}{W_{ymax} - W_{ymin}} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

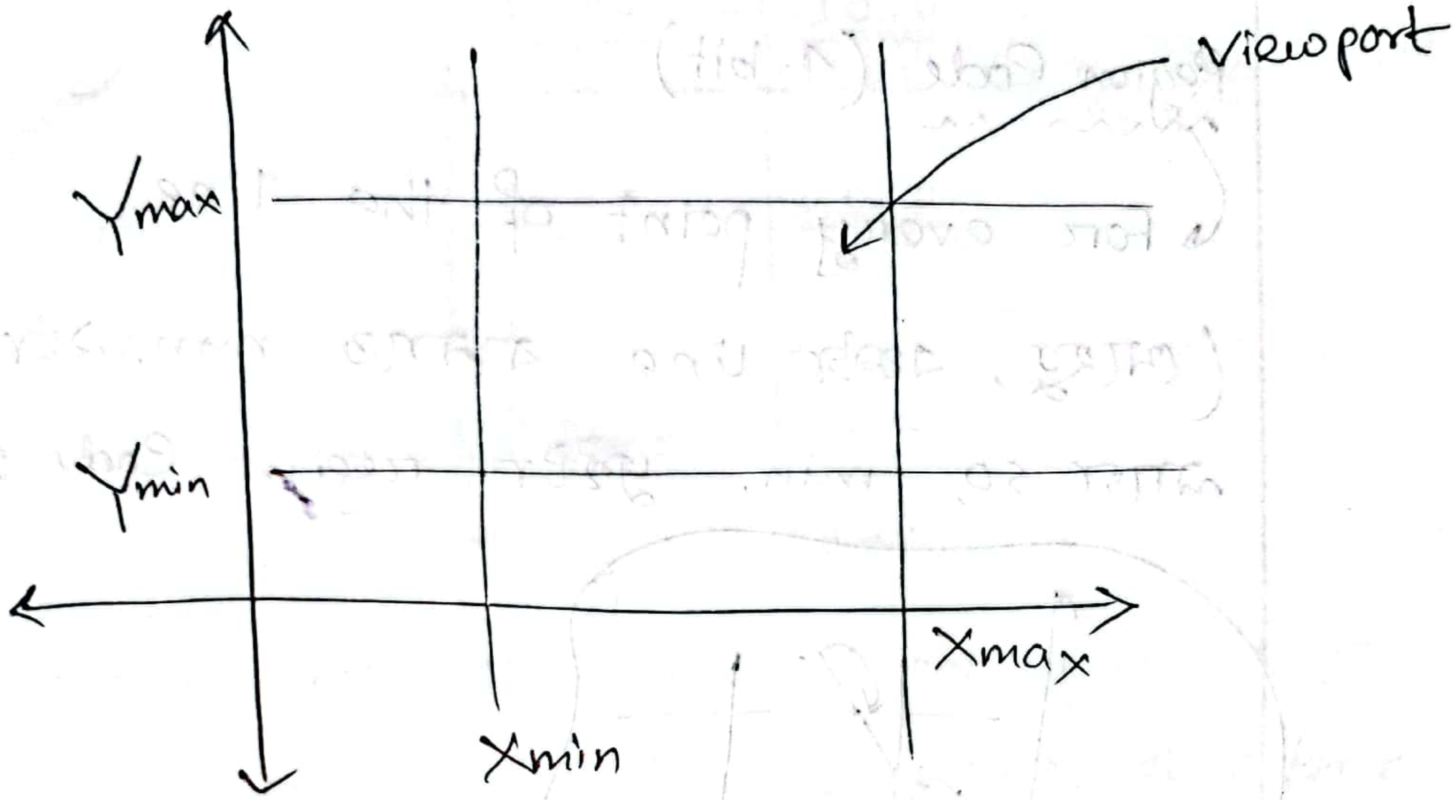
$$\text{Mat1}: \begin{pmatrix} 1 & 0 & V_{xmin} \\ 0 & 1 & V_{ymin} \\ 0 & 0 & 1 \end{pmatrix}$$

$$N = \text{Mat1} \cdot \text{Mat2} \cdot \text{Mat3}$$

# Class - 23

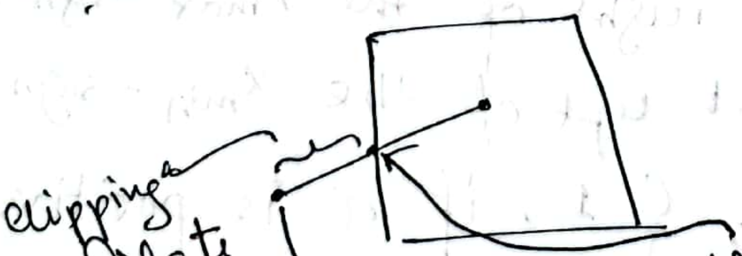
26/7/22

## Viewing & Clipping:




* Viewport এর বাইরে যা কিছু থাকবে সব consideration এর বাইরে / Not visible.

* কিছু বাইরে কিছু ভিতরে থাকলে clipping হবে।



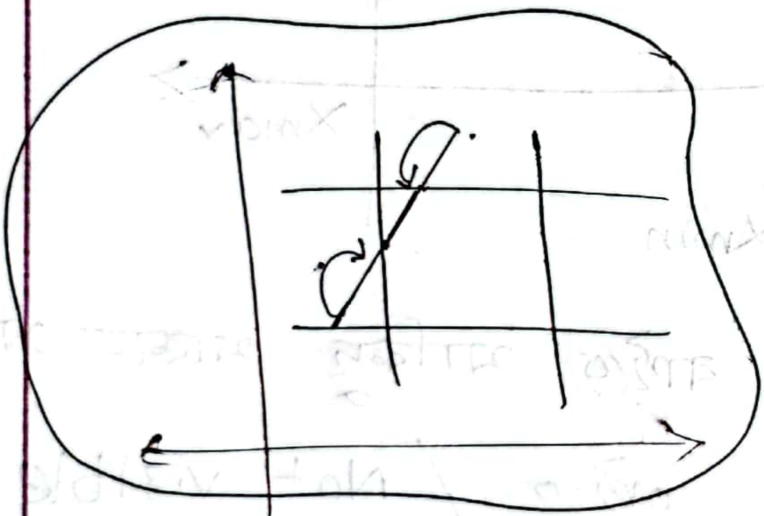

এই point কে এখান নিয়ে গেলে clipping হয়ে যাবে।

## Sutherland Algorithm:

Region Code (4-bit)

→ For every point of the line

(যেহেতু, একটার line থাকতে min. ২টা point লাগে so, min. দুইটার region code হবে)



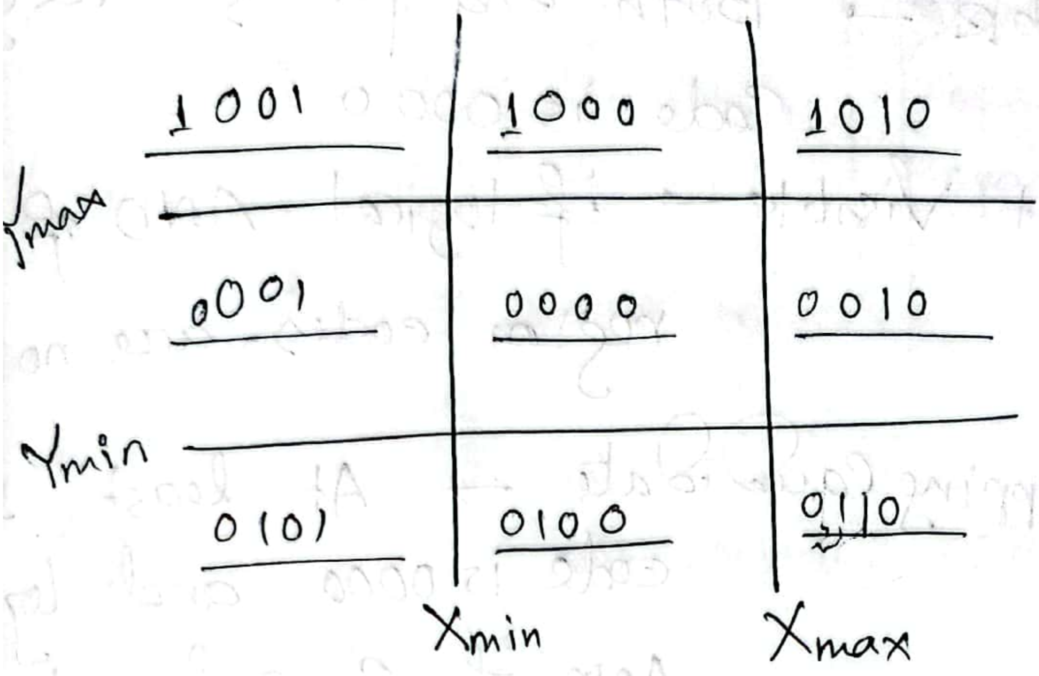
i) bit 1: the point above of the $y_{max} = sign(y - y_{m}$

ii) bit 2: the point below the $y_{min} = sign(y_{min} -$

iii) bit 3: the point right of the $x_{max} = sign(x - x_{max}$

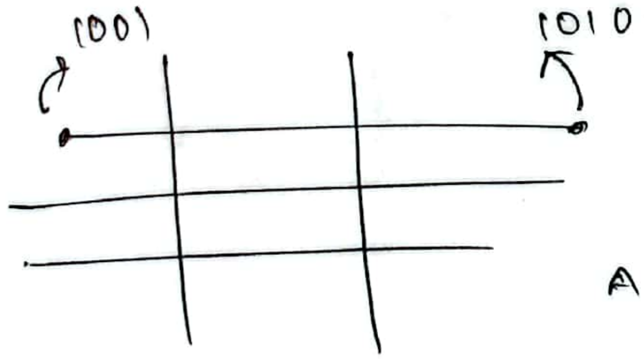
iv) bit 4: the point left of the $x_{min} = sign(x_{min} - x$

$$sign(a) = \begin{cases} 1, & \text{if } a \text{ is positive} \\ 0, & \text{otherwise} \end{cases}$$




কোনো line এর দুই প্রান্তের point নিয়ে আগে দেখব যে Bit Code 0000 আছে কিনা যদি না থাকে, তাহলে দুইগুলোকে AND করব, তবু যদি না থাকে তাহলে Line টা Viewport এর বাইরে।

Like -




In this case

| | |
|---|---|
| | 1001 |
| | 1010 |
| AND → | 1000 |

still not 0000

So, line is বাইরে

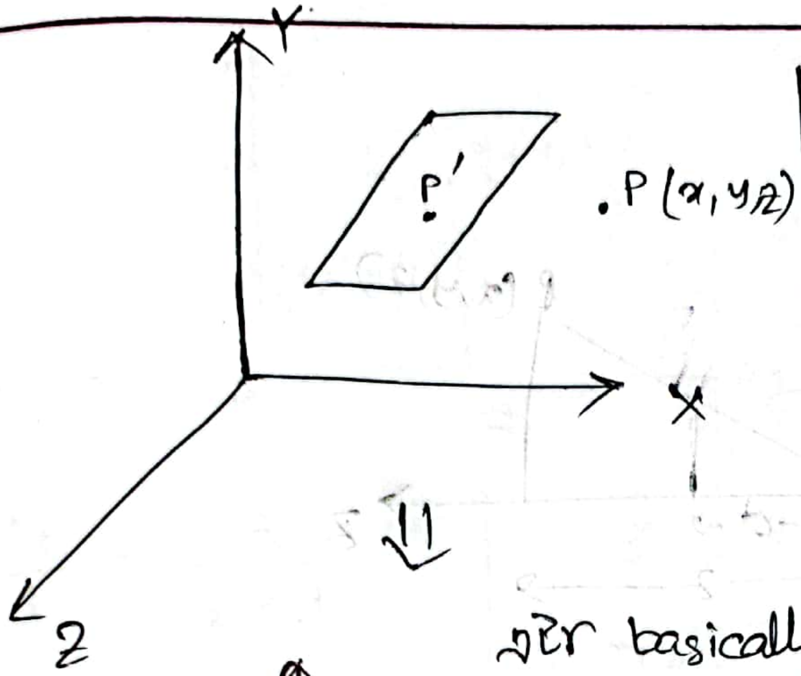
i) Visible → Both end point's region Code is 0000

ii) Not Visible → if logical AND of the region codes are not 0000

iii) Clipping Candidate → At least 1 point's code is 0000 and logical AND of 2 endpoints must be 0000.

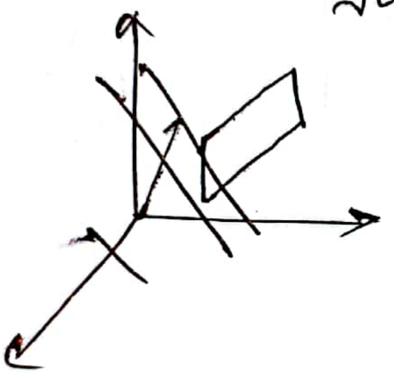
Class - 24

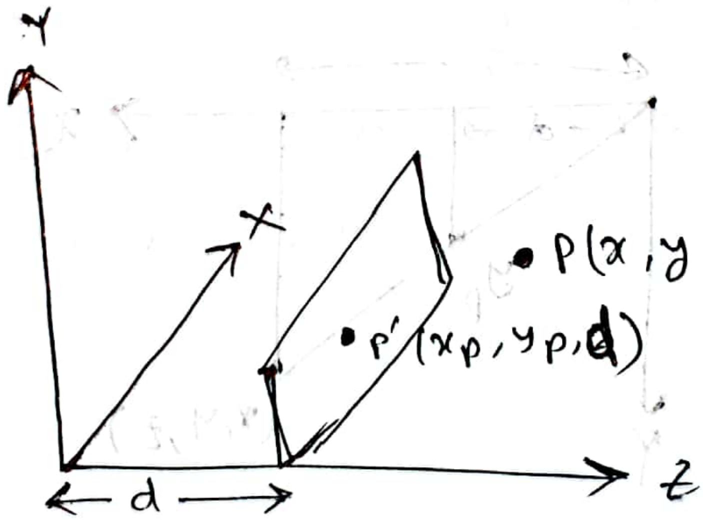
31/7/22

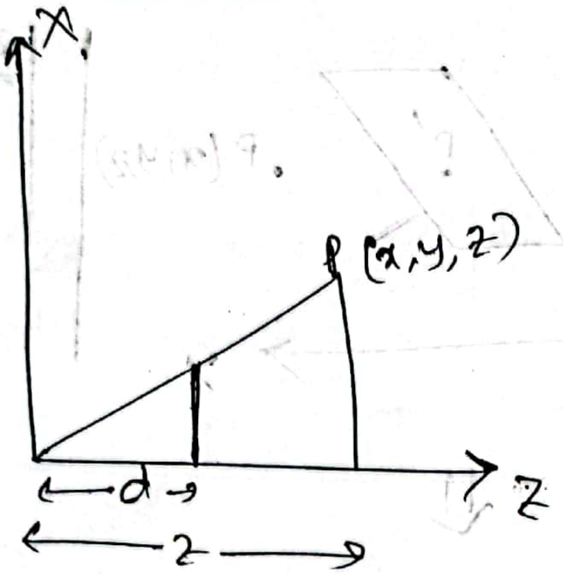



(A) Projection 3D object কে 2D space এ map করে

এটা basically এরকম

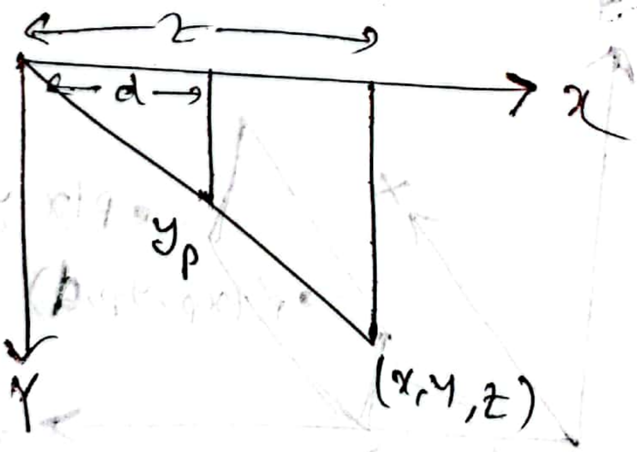






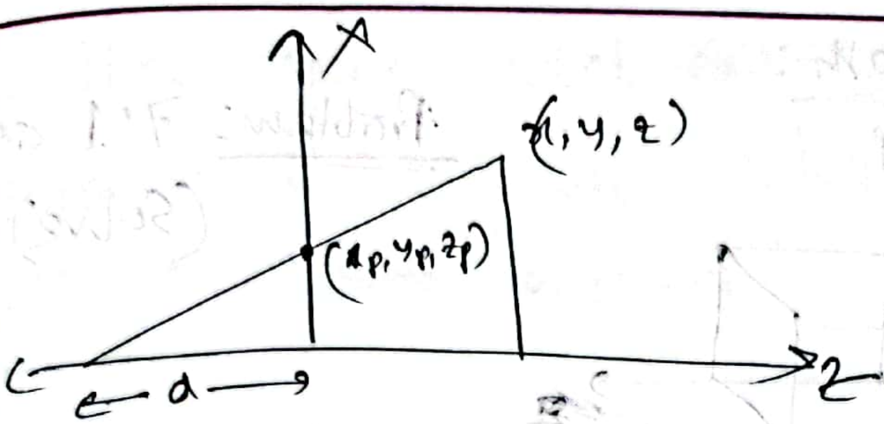

$$\frac{x_p}{d} = \frac{x}{z}$$

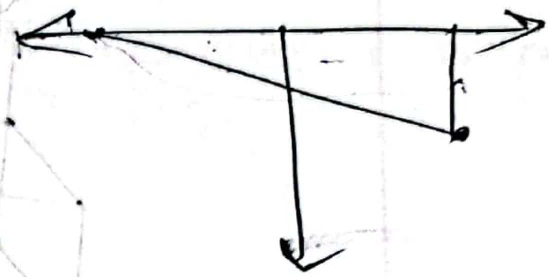
$$\Rightarrow x_p = dx/z = x/\frac{z}{d}$$




$$\Rightarrow Y_p = \frac{y}{z/d}$$

প্রমাণ হলো .



$$\frac{x_p}{d} = \frac{x}{z+d}$$

$$\therefore x_p = \frac{x}{z/d+1}$$

Similarly

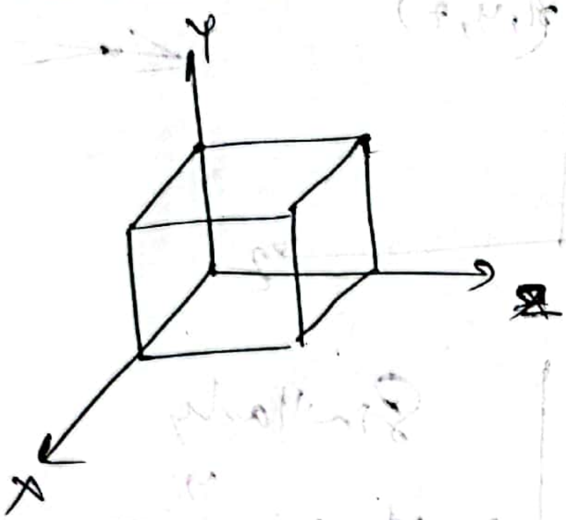
$$\Rightarrow y_p = \frac{y}{z/d+1}$$

[V.N.I] ***

Generalized Projection Matrix এবং Derivation exam এ আসার দিক থেকে অনেক কমন, তবে আসার chance অনেক।

Projection Math:

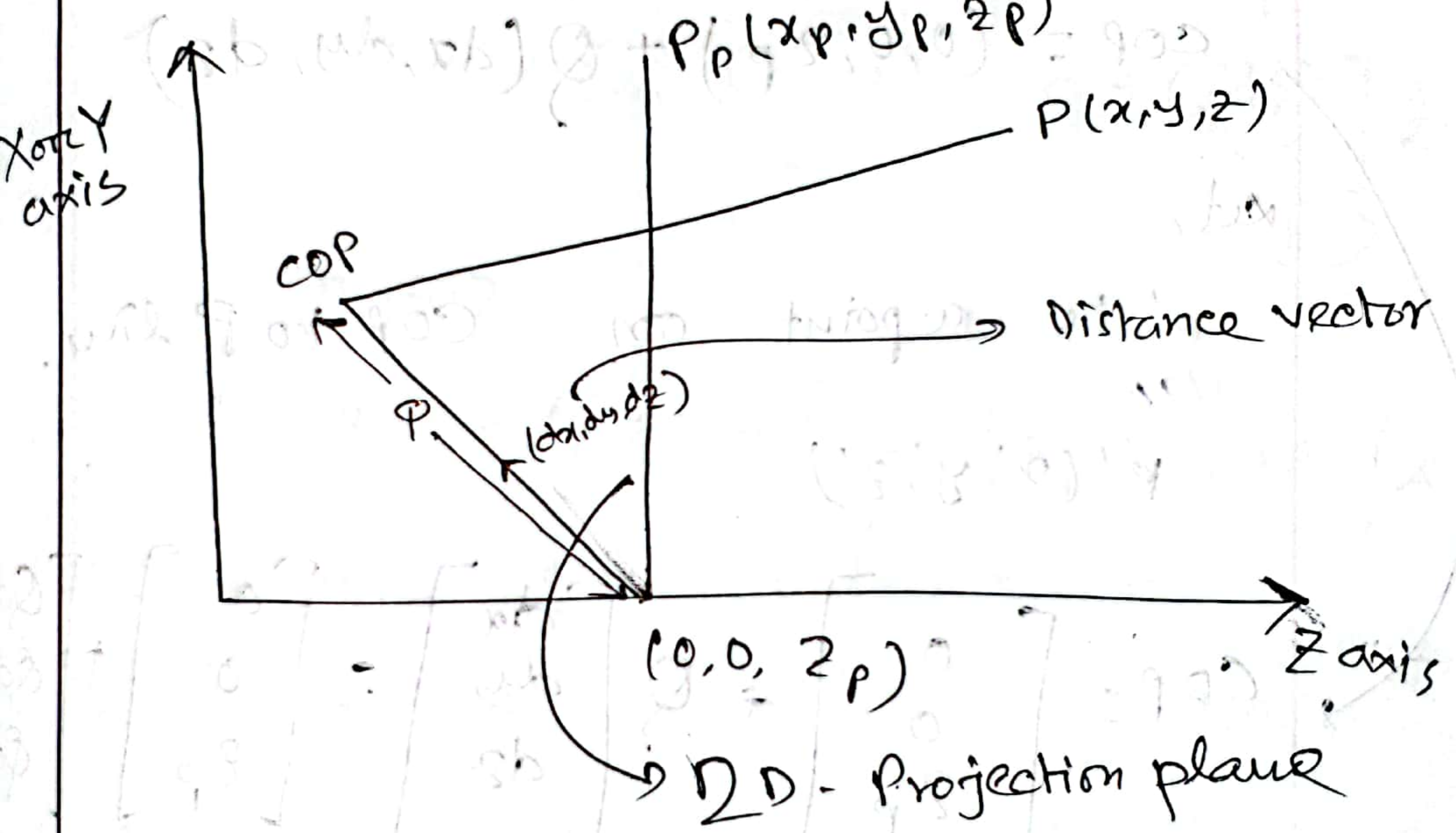
Problem: 7.1 করবে (Solve)



## Class - 25      2/8/22

# বিভিন্ন Color Model দেখানো হলো, CMY, RGB, HSV, HSL etc. কীভাবে convert করতে হয় একটা থেকে আরেকটায়

## General Projection Matrix Derivation:




<u>Main goal</u>: $x_p, y_p, z_p$ বের করা,

COP থেকে P পর্যন্ত line/vector হল।

$= COP + t(P - COP)$ ← এটা হচ্ছে parametric eqn.

এখন, COP কে ভাঙ্গলে হবে,

$COP = (0, 0, z_p) + Q(dx, dy, dz)$

Let,

$p'$ is a point on COP to P line.

$p'(x', y', z')$

$$COP = \begin{bmatrix} 0 \\ 0 \\ z_p \end{bmatrix} + Q \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ z_p \end{bmatrix} + \begin{bmatrix} Qdx \\ Qdy \\ Qdz \end{bmatrix}$$

$$= \begin{bmatrix} Qdx \\ Qdy \\ z_p + Qdz \end{bmatrix}$$

p' (x', y', z') বিন্দুটির ঐ Line কে Satisfy করবে.

So, ঐ eqn এ বসিয়ে লিখতে পারি.

$$x' = Q\,dx + (x - Q\,dx)\,t \quad \text{—— (i)}$$

$$y' = Q\,dy + (y - Q\,dy)\,t \quad \text{—— (ii)}$$

$$z' = z_p + Q\,dz + [z - (z_p + Q\,dz)]\,t \quad \text{—— (iii)}$$

at eqn (iii),

$$z_p = z_p + Q\,dz + [z - (z_p + Q\,dz)]\,t$$

এখান থেকে $t$ এর value পাবো,

এই $t$ এর value (i), (ii) তে বসালে,

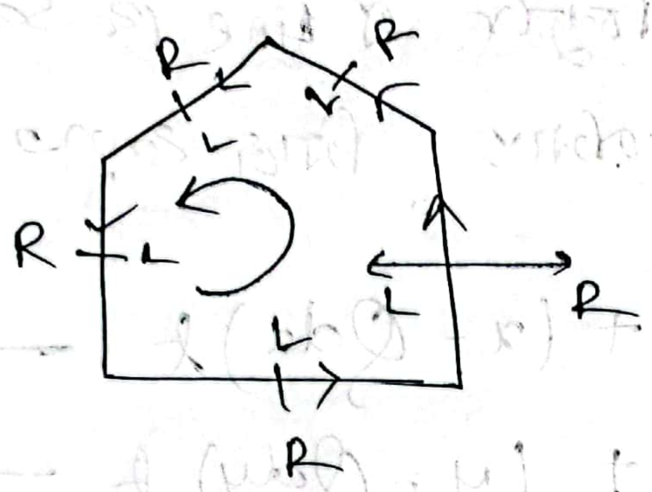
So, $x' = x_p$, $y' = y_p$ পেলাম $x_p$, $y_p$ বের হবে,

Slide + খাতা এ আছে

MZI

# THE LAST CLASS

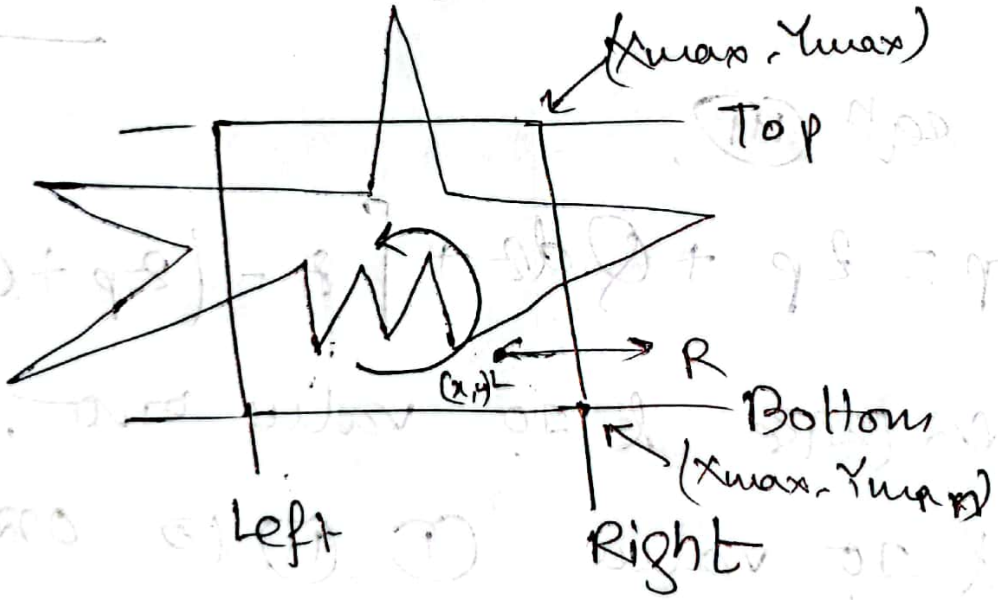
"17/8/22"




* All lefts are inside for AntiClockwise

## Sutherland-Hodgman Algorithm:

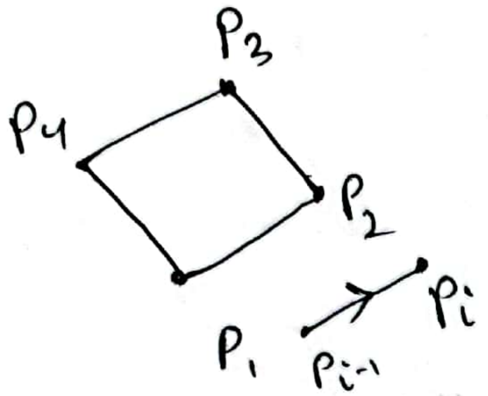
→ Perfect Algo. for Polygon Clipping.

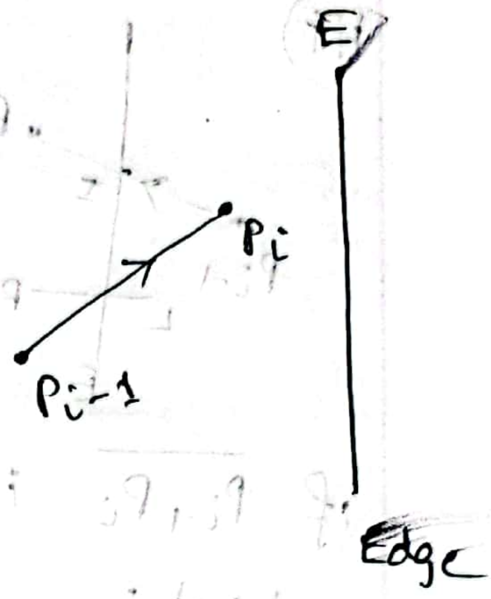



$|\vec{C}| = (x_2 - x_1)(y - y_1) - (y_2 - y_1)(x_2 - x_1)$

= +ve হলে as left এ

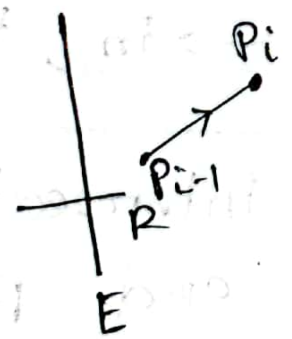
| | |
|---|---|
| $x_1 \to X_{max}$ | $y_1 \to Y_{min}$ |
| $x_2 \to X_{max}$ | $y_2 \to Y_{max}$ |



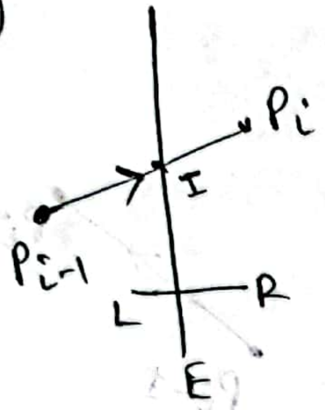

i) if $\overline{P_{i-1}P_i}$ is completely ~~outside~~ inside of the edge E, then output is $\boxed{P_i \text{ vertex.}}$

ii) if $\overline{P_{i-1}P_i}$ is completely outside of edge E, then output is $\boxed{\text{nothing}}$




↳ কারণ, viewport এর বাইরে চলে যাবে, দেখাবেই যাবে না.

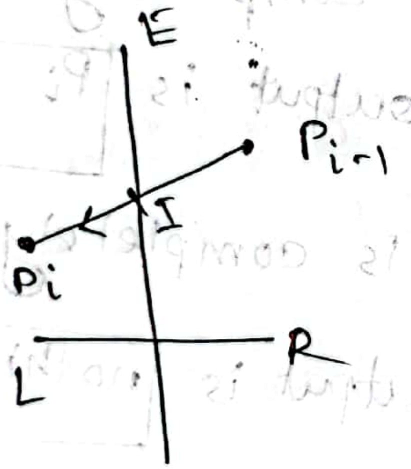
iii) (inside → outside)




if $\overline{P_{i-1}P_i}$ is in → out then,

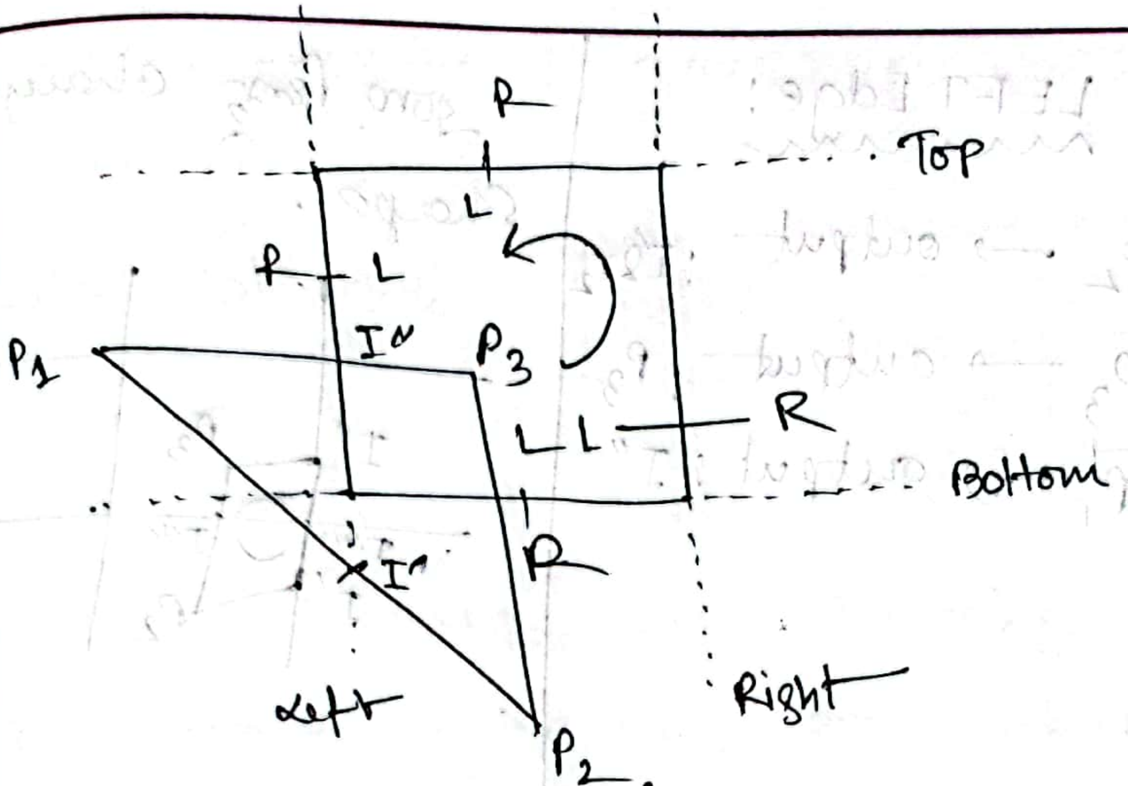
output is ~~vertex~~ I $\boxed{\text{intersection point I}}$

iv) (out → in)




if $\overline{P_{i-1}P_i}$ is out → in, then,

output is $\boxed{\text{both intersection point I and } P_i}$




i) RIGHT EDGE:

$\overline{P_1P_2} \longrightarrow$ output : $P_2$

$\overline{P_2P_3} \longrightarrow$ output : $P_3$

$\overline{P_3P_1} \longrightarrow$ output : $P_1$

ii) TOP Edge:

$\overline{P_1P_2} \longrightarrow$ output : $P_2$

$\overline{P_2P_3} \longrightarrow$ output : $P_3$

$\overline{P_3P_1} \longrightarrow$ output : $P_1$

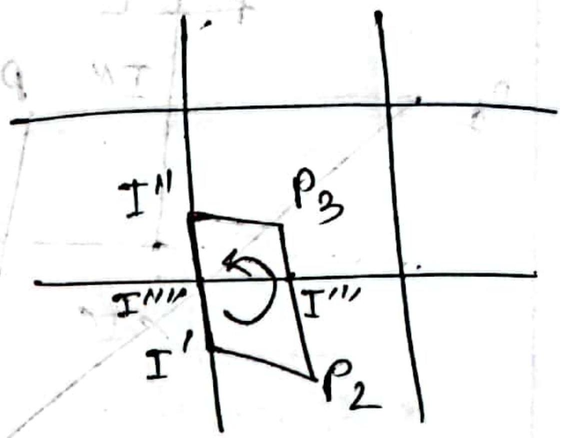
iii) LEFT Edge:

$\overline{P_1P_2} \longrightarrow$ output : $I'$ & $P_2$

$\overline{P_2P_3} \longrightarrow$ output : $P_3$

$\overline{P_3P_1} \longrightarrow$ output : $I''$

এবার কিন্তু change হয়ে shape.




iv) BOTTOM EDGE:

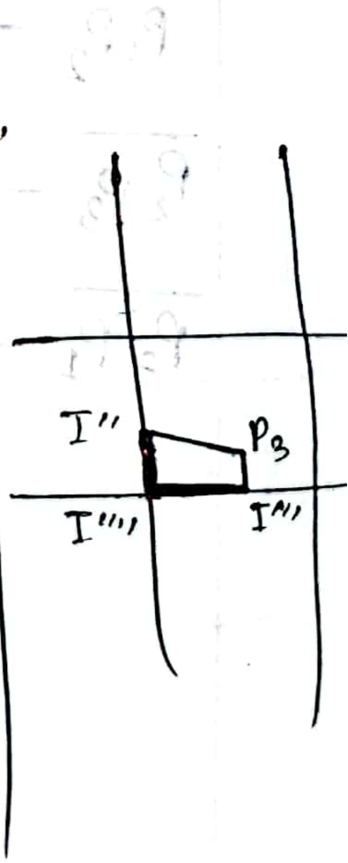
$\overline{I'P_2} \longrightarrow$ output : –

$\overline{P_2P_3} \longrightarrow$ output : $I'''$ & $P_3$

$\overline{P_3I''} \longrightarrow$ output : $I''$

$\overline{I''I'} \longrightarrow$ output : $I''''$

$\therefore$ Object : $I''' \; P_3 \; I'' \; I''''$



এখানে, $I''I''''$ & $I''''I'''$ তে কোনো Double edge হতে পারে।

এটার $Sol^n$ হবে Weiler Atherton Algorithm

***

এটা দেখে নিতে হবে।

Cohen-Sutherland, Liang Barsky এর example এর math গুলো A sec. এর lecture এ sir করিয়ে দিয়েছেন। এগুলো দেখে নিতে হবে।

*** V.V.I → All Derivations