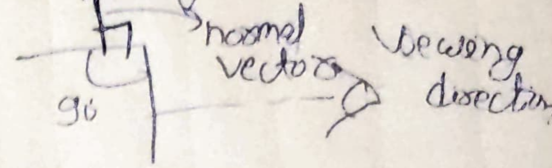
Date - 17/4/23

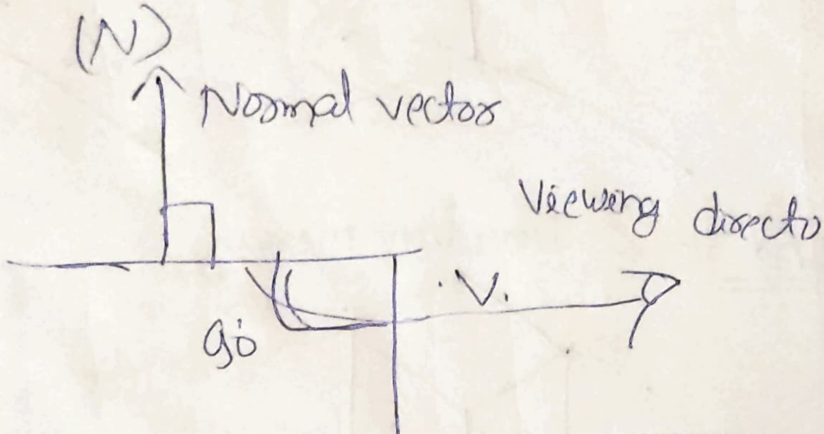
## Back face removal / Hidden surface removal algorithm :-

⇒ Removing the surfaces which are not visible.

Mathematics behind this - based on angle




Normal vector → 90° से above.




$\boxed{V.N. = |V||N| \cos\theta}$

if $\theta = 0$ to $90° \Rightarrow +ve$.
if $\theta = 90° > 180 \Rightarrow -ve$

If V.N. is +ve = back face.
if V.N. is -ve = visible

## Algorithm

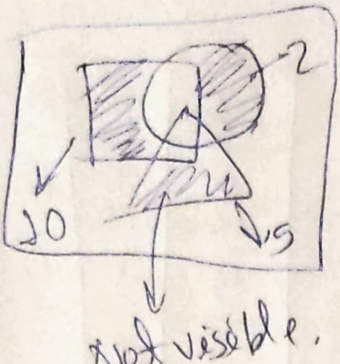
① **Z- buffer algo. :-** means Z co-ordinate or Z(depth)

& calculate $Z_{min}$

⇒ take places pixel by pixel

जिसका depth कम को select करना है

One pixel → No compare.




here select circle 1st → depth 2

Triangle - rectangle.

depth কম select হবে

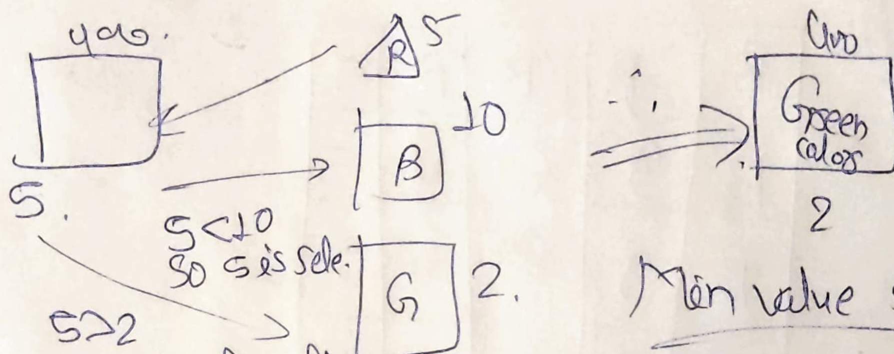
depth = distance from user

## Algorithm:

(i) Set all depth value. $d(i,j) = max$

(ii) Set all color value $C(i,j)$ = background.

eg.




here 2 is less than 5 ∴ 2 select.

Min value select

(iii) For each polygon for each pixel

find depth of $P(i,j) = Z$

if $Z < d(i,j)$

than $d(i,j) = Z$

$C(i,j)$ = color

eg (1)

Blank ⇒ Background color (B)

Q.

| | | | | |
|---|---|---|---|---|
| B | B | B | B | S |
| B | B | B | 4 | 5 |
| B | B | 3 | 4 | 5 |
| B | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |

1 is sliding

| | | | | |
|---|---|---|---|---|
| B | B | B | B | B |
| B | 9 | 6 | | |
| B | 9 | 6 | 3 | 0 |
| | 9 | | | |
| | | | | |

Compare each pixel

Final Soln

| | | | | |
|---|---|---|---|---|
| B | B | B | B | S |
| B. | 9 | 6 | 4 | 5 |
| B | 9 | 3. | 3. | 0 |
| B | 2 | 3 | 4 | 5. |
| 1 | 2 | 3 | 4 | 5. |

5 is compare with B ⇒ 5.

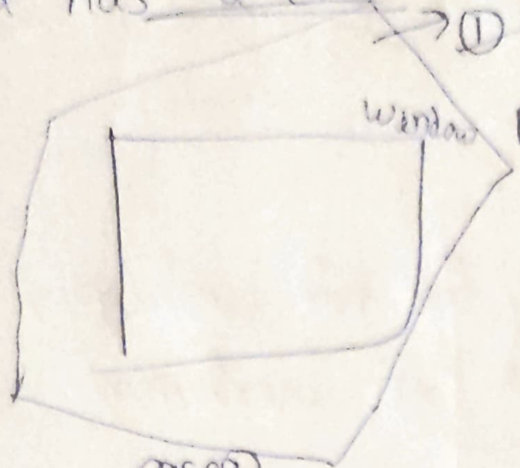
here $6 > 3 \Rightarrow 3$ (min depth select.)

$4 > 3 \Rightarrow 3$ (min)

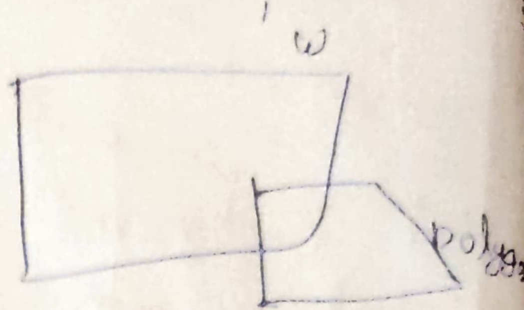
$5 > 0 \Rightarrow 0.$

## ② Area subdivesion :-

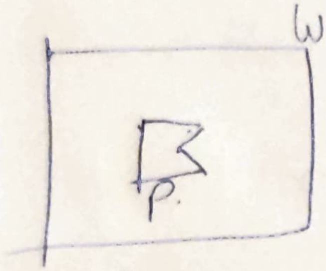
It has 4 case → (I) overlapping/surrounding polygon



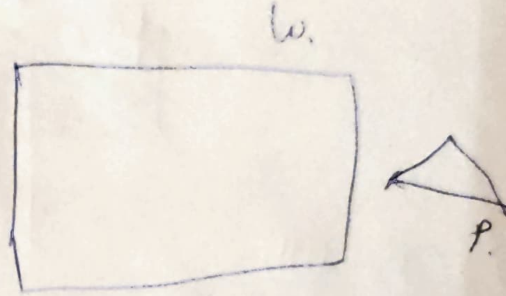

case(I)




case(II)




case (III)




case (IV)

**case (I) Surrounding Polygon :-**

**case (II) Overlapping** :- ~~tor~~ partial. (some portion of polygon cutting the

**case (III) Inside polygon.**

**case (IV) Outside polygon,**

If 4 cases ~~nor~~ are not than we need to divide.

If case-1:-(we have only one polygon)

(i) Set all pixel to the color of surrounding polygon.

f.

Case-(ii)

(i) Set background color → For not overlapping

(ii) Set polygon color. → जहां overlapping

(Intersection color)

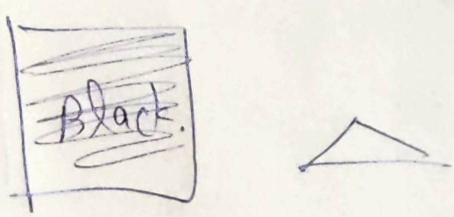

If case-(iii)

(i) Set background color

(ii) Set polygon color.

If case-(iv)

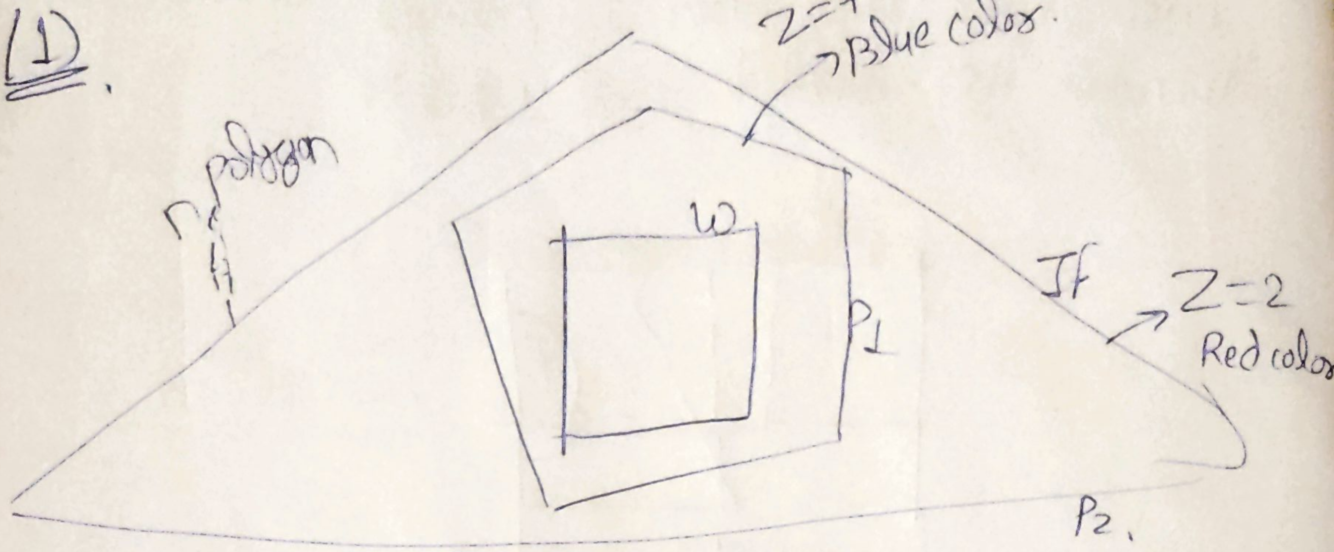
(1) Set background




If this 4 cases or not than the situation

⇒ If not above 4 cases

(1).

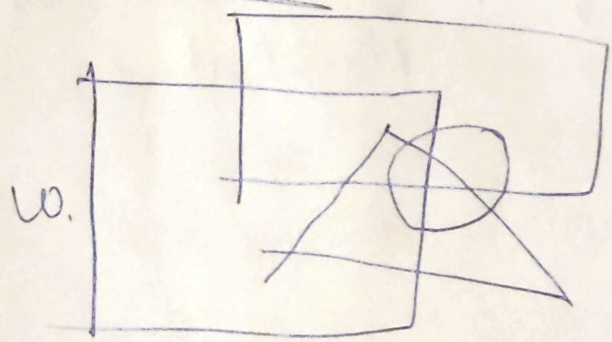



⇒ ∴ Color = Red color (because Z(depth) is min)

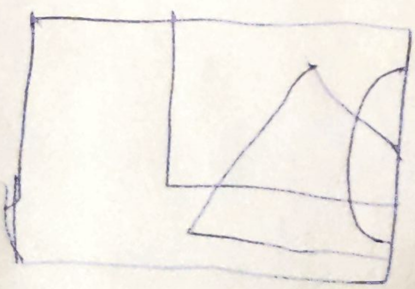
(i) If more than 1 polygon & all are Surrounding polygon than compare Z(depth) value.

(ii) Set color of min. Z value.

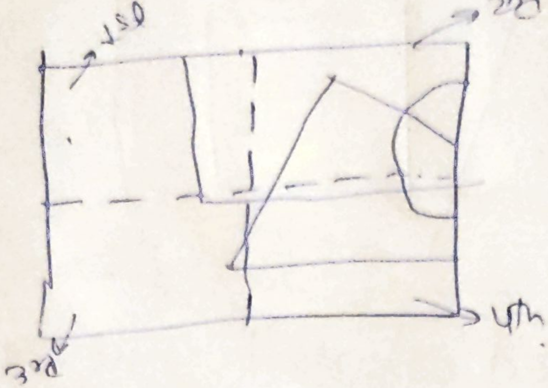
(2) Areas of division :-



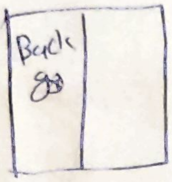

Now remove window outside portion.



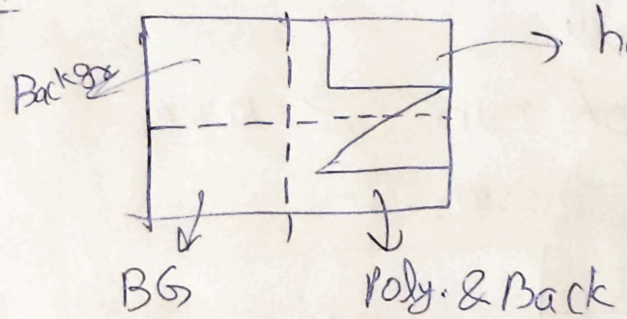
*Next* we have to divide the window until we get min 1 polygon.
Window is divided in 4 parts




For 1st part

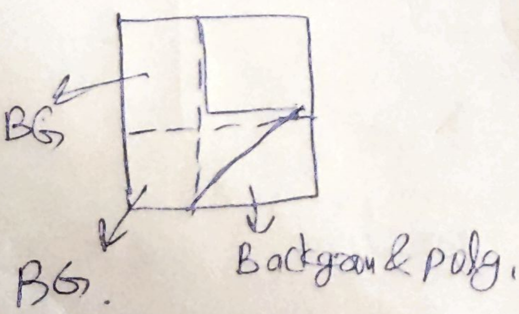



For 2nd part




→ here ucase is not apply.
∴ divide into 4.

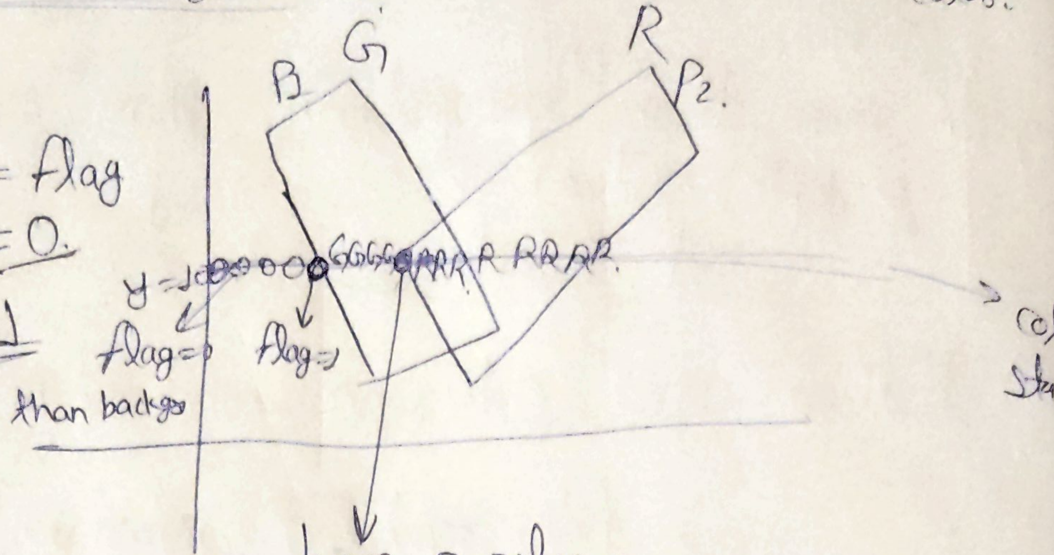
(a)



③ Scanline Algorithm :- Draw a line & start the color.

Background = flag
= 0.

Polygon = 1




here 2 polygon

⇒ more than 2 polygon → compare Z(depth) value
select Zmin

⇒ Area coherer

⇒ Algorithm :-

Step(i) Initialization.

(a) keep list of all polygon

(b) keep edge of all polygon

(c) keep active edge list info. (currently active edge)

(d) Flag information (on/off)

If flag = 0 ⇒ Background

flag = 1 ⇒ Polygon

Step(ii) (a) Write y scanline

(b) write all edges of y scan line.

(c) Scan ~~unul~~ untill flag ≠ 1.

(d) If one polygon, than set the color of polygon.

(e) If more than ~~two~~ one polygon., than compare Z(depth) value
- Set lowest Z value color.

(f) Follow area coherence mechanism.

(means not repeted comparision
बार बार compare)

10-mark Derive W to V. [] matrix

Date- 18/4/23

## Window to viewport transformation :-

window
⇒ **World co-ordinate.**
↳ portion of image to be displays.

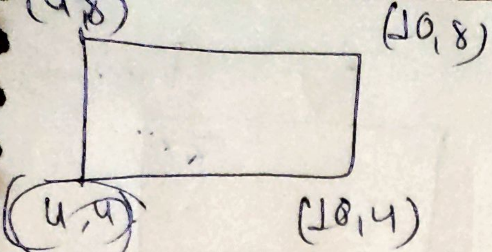
**Viewport**: portion of display device where image / ~~pixel~~ pic appears.

**Steps:-**

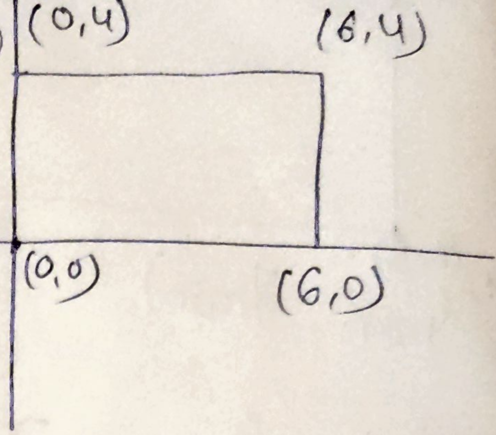
**Step 1** Translate window towards origin.
(-ve translation ~~rate~~)

add $(-x, -y)$ lower corner.

$\therefore x' = 0$
$y' = 0.$




eg.




means $4+(-4)=0$
$4+(-4)=0$

(−4 in all)

lowest corner के value के हिसाब से minus

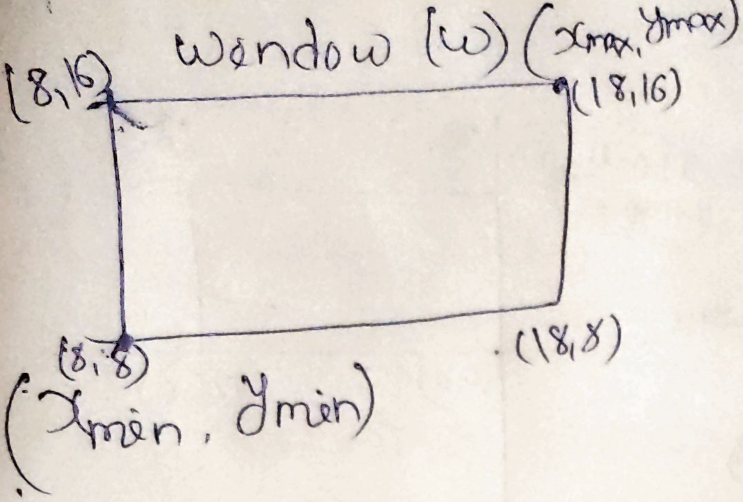
if $(9,6) \rightarrow (-9,-6)$

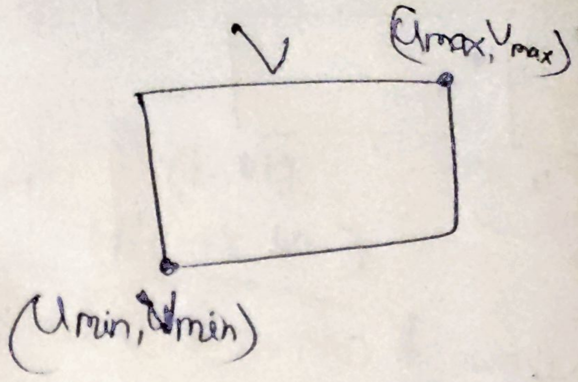
सबको से minus.

<u>Step (II)</u>. Resize window to the size of view port. (Scaling transformation).

<u>Step (III)</u>. Translate the resized window. (+ve translation).

−ve translation
+ve Scaling
+ve translation



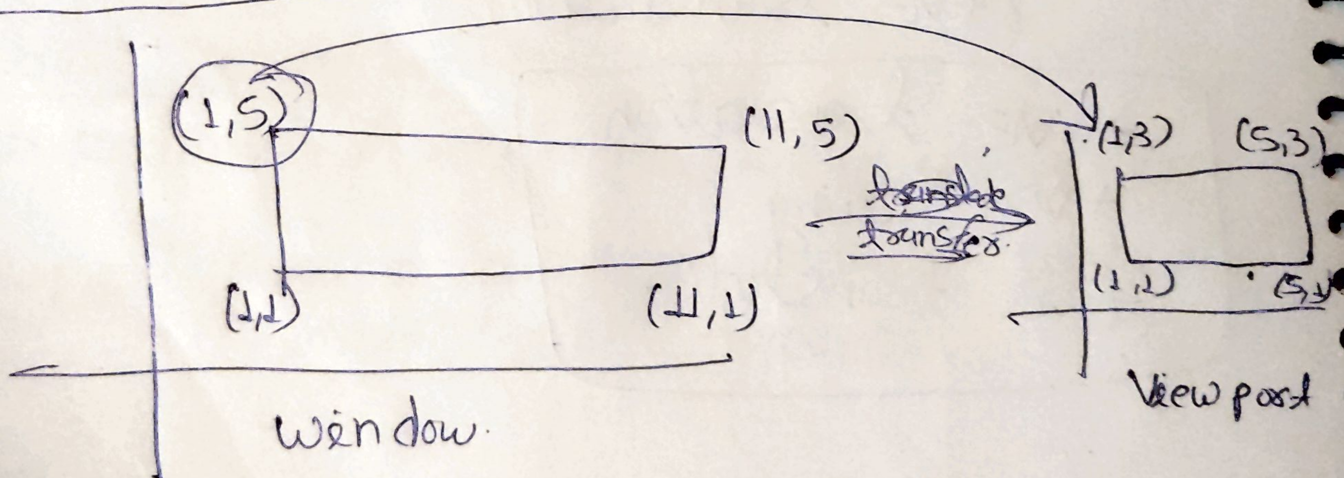

Ratio rule :- For any point $(x, y) \longrightarrow (u, v)$
or Proportional rule :-

$$\boxed{\frac{x - x_{min}}{x_{max} - x_{min}} = \frac{u - u_{min}}{u_{max} - u_{min}}} \quad (I)$$

&

$$\frac{y - y_{min}}{y_{max} - y_{min}} = \frac{v - v_{min}}{v_{max} - v_{min}} \quad (II)$$

eg:




$x_{min} = 1$ $\qquad$ $u_{min} = 1$

$x_{max} = 11$ $\qquad$ $u_{max} = 5$.

$y_{min} = 1$ $\qquad$ $v_{min} = 1$

$y_{max} = 5$ $\qquad$ $v_{max}$ 3.

Let Point $x = 1$ $\qquad$ & point $u = 1$

$y = 5$ $\qquad$ $v = 3$

$\therefore$ Formula For $x$

$$\frac{1-1}{11-1} = \frac{1-1}{5-1} \Rightarrow \boxed{0 = 0}.$$

For $y$

$$\frac{5-1}{5-1} = \frac{3-1}{3-1} \Rightarrow \boxed{1 = 1}.$$

From eqⁿ (I) $(x, y) \longrightarrow (u, v)$

$$\frac{x - x_{min}}{x_{max} - x_{min}} = \frac{u - u_{min}}{u_{max} - u_{min}}$$

+ve translate

$$\therefore \; u = \frac{\overset{\text{+ve Trans}}{(x - x_{min})}(u_{max} - u_{min})}{x_{max} - x_{min}} + u_{min}$$

$\rightarrow$ Scaling.

From eqⁿ (II)

$$\frac{y - y_{min}}{y_{max} - y_{min}} = \frac{v - v_{min}}{v_{max} - v_{min}}$$

$$v = \frac{(y - y_{min})(v_{max} - v_{min})}{y_{max} - y_{min}} + v_{min}.$$

$x_{min} = A$. $\quad$ $u_{min} = C$

$y_{min} = B$ $\quad$ $v_{min} = D$.

& $\dfrac{u_{max} - u_{min}}{x_{max} - x_{min}} = E$.

$\dfrac{\cancel{y_{max} - y_m}\; v_{max} - v_{min}}{y_{max} - y_{min}} = F$.

$\therefore$

$$\boxed{\begin{aligned} u &= (x - A)E + C. \\ v &= (y - B)F + D. \end{aligned}}$$

Find this in matrix form:- $\quad -T, S, +T.$

calculation में stack में first means +ve translation.

| |
|---|
| +T |
| S |
| -T |

① +ve translation (means C & D)

$t_x = C, \; t_y = D.$

$$\begin{bmatrix} 1 & 0 & C \\ 0 & 1 & D \\ 0 & 0 & 1 \end{bmatrix}.$$

② ~~Scaling~~ ② Scaling $(E, F)$

$$\begin{bmatrix} E & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(3) -ve translation (A & B)

$$\begin{bmatrix} 1 & 0 & -A \\ 0 & 1 & -B \\ 0 & 0 & 1 \end{bmatrix}$$

∴ Concatenation

$$\begin{bmatrix} 1 & 0 & C \\ 0 & 1 & D \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} E & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -A \\ 0 & 1 & -B \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} E & 0 & C \\ 0 & F & D \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -A \\ 0 & 1 & -B \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} E & 0 & -EA+C \\ 0 & F & -FB+D \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

3×3 3×1 x y → (UV)

$$= \begin{bmatrix} xE + (-EA+C) \\ yF + (-FB+D) \\ \cancel{0}\ 1 \end{bmatrix} \Rightarrow \boxed{\begin{matrix} E(x-A)+C \\ F(y-B)+D \end{matrix}}$$

Ans

$u =$

$v =$

Que. $x_{min} = 20$, $y_{min} = 40$, $u_{min} = 30$

$x_{max} = 80$, $y_{max} = 80$, $u_{max} = 60$

$v_{min} = 40$

$v_{max} = 60$

$(x, y) \longrightarrow (30, 80)$

than. $\downarrow$

$(u, v) \longrightarrow ?$

$\because$ $$u = \frac{(x - x_{min})(u_{max} - u_{min})}{x_{max} - x_{min}} + u_{min}$$

$$= \frac{(30-20)(60-30)}{80-20} + 30.$$

$$= \frac{10 \times 30}{60} + 30 \Rightarrow \frac{\cancel{300}^{15}}{\cancel{60}_{2}} + 30 \quad \frac{\cancel{300}^{5}}{\cancel{60}} + 30$$

$$\frac{15+60}{2} \Rightarrow \frac{75}{2} \quad 35.$$

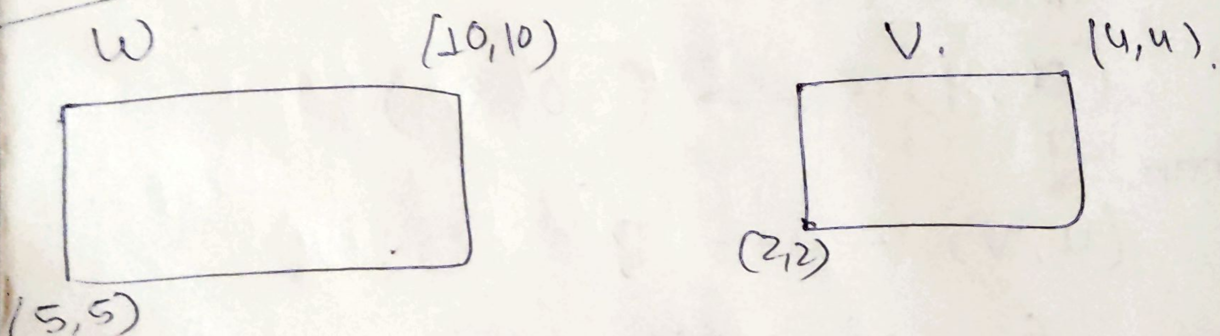
$$\therefore V = \frac{(y - y_{min})(V_{max} - V_{min})}{y_{max} - y_{min}} + V_{min}$$

$$= \frac{(80-40)(60-40)}{80-40} + 40.$$

$$= \frac{40 \times 20}{40} + 40$$

$$\Rightarrow 20 + 40 = 60 \quad A$$

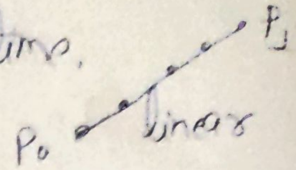
In diagram form Ae,



Date- 24/4/23

# ☆ Curve Generation :-

⇒ Parametric form :- Operate with time.

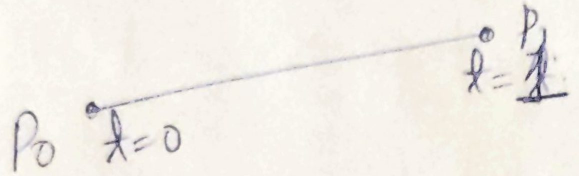



eqn of parametric form:

linear curve. $\boxed{L_0(t) = (1-t)P_0 + tP_1.}$

where $P_0, P_1$ = Points  
$t$ = time.

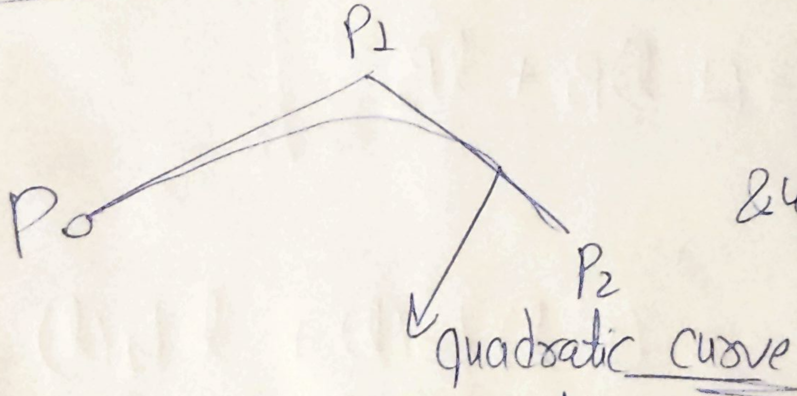
$P_0$ = Previous point  
$P_1$ = last point




⇒ At $t=0$, ∴ $L_0(0) = (1-0)P_0 + 0 \times P_1 \Rightarrow \underline{P_0}$

At $\underline{t = 1}_{max}$, ∴ $L_0(t) = (1-1)P_0 + 1 \times P_1 \Rightarrow \underline{P_1}$

For curve we use min 3 point

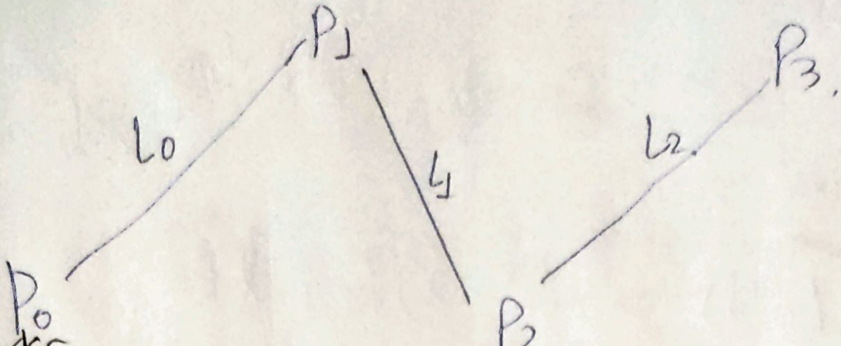



$P_0P_1$ = linear  
$P_1P_2$ = linear  
& we get curve.

2 linear = one quadratic curve.

2 quadratic = one cubic.

For cubic → 4 points.

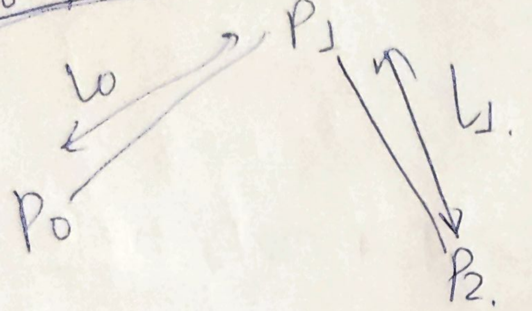



$L_0, L_1, L_2$ = linear curve

10 marks

* Derive quadratic & cubic eqn for curve generation

For quadratic




For $P_0P_1$

$$L_0(t) = (1-t)P_0 + t P_1$$

For $P_1P_2$

$$L_1(t) = (1-t)P_1 + tP_2$$

Quadratic

$$Q_0(t) = (1-t)\, L_0(t) + t\, L_1(t)$$

$$= (1-t)[(1-t)P_0 + tP_1] + t[(1-t)P_1 + tP_2]$$

$$= (1-t)^2 P_0 + (1-t)tP_1 + (1-t)tP_1 + t^2P_2$$

$$= (1+t^2-2t)P_0 + 2[(1-t)tP_1] + t^2P_2 \quad \text{— (1)}$$

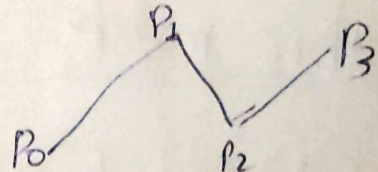
$$= (t^2 - 2t + 1) P_0 + (-2t^2 + 2t) P_1 + t^2 P_2$$

$$\begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix} \begin{bmatrix} P_0 & P_1 & P_2 \end{bmatrix} \begin{bmatrix} -1 & 2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Same

$$= \begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 & P_1 & P_2 \end{bmatrix}$$

Quadratic matrix form




$\Rightarrow$ For cubic :- 2 Quadratic = 1 cubic

Quadratic

From eq(1)

$$Q_0(t) = (1-t)^2 P_0 + 2(1-t)t P_1 + t^2 P_2.$$

$$\therefore Q_1(t) = (1-t)^2 P_1 + 2(1-t)t P_2 + t^2 P_3$$

$\rightarrow$ only point change

$\therefore$ Cubic.

$$C_0(t) = (1-t) Q_0(t) + t\, Q_1(t)$$

$$= (1-t)\left[(1-t)^2 P_0 + 2(1-t)t P_1 + t^2 P_2\right]$$

$$+ t\left[(1-t)^2 P_1 + 2(1-t)t P_2 + t^2 P_3\right].$$

$$G_0(t) = (1-t)^3 P_0 + 2(1-t)^2 t P_1 + (1-t) t^2 P_2$$
$$+ (1-t)^2 t P_1 + 2(1-t) t^2 P_2 + t^3 P_3.$$

$$= (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2.$$
$$+ t^3 P_3.$$
$$= (1 - t^3 + 3t^2 - 3t) P_0 + 3(1 + t^2 - 2t) t P_1.$$
$$+ (3t^2 - 3t^3) P_2 + t^3 P_3.$$
$$= (1 - t^3 + 3t^2 - 3t) P_0 + (3t + 3t^3 - 6t^2) P_1$$
$$+ (3t^2 - 3t^3) P_2 + t^3 P_3.$$

<u>Cubic matrix</u>

$$\begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 & P_1 & P_2 & P_3 \end{bmatrix}$$

convert in matrix

arrange in higher order.

$$= (-t^3 + 3t^2 - 3t + 1) P_0 + (3t^3 - 6t^2 + 3t) P_1$$
$$+ (-3t^3 + 3t^2) P_2 + t^3 P_3$$

Imp

$\boxed{\text{Derive cubic eqn}}$

# Color Models :-

Date-25/4/23

## (1) RGB color Model :-

Base color — Red, Green, Blue.

R – (0-255) = $2^8$ bit

G – 256 = $2^8$ bit

B – 256 = $2^8$ bit

1 single color = 8+8+8 = 24 bit

start 24 0's (0 0 0 0 - - - -

end 24 1's (1 1 1 - - - -

which is more than 16 million color.

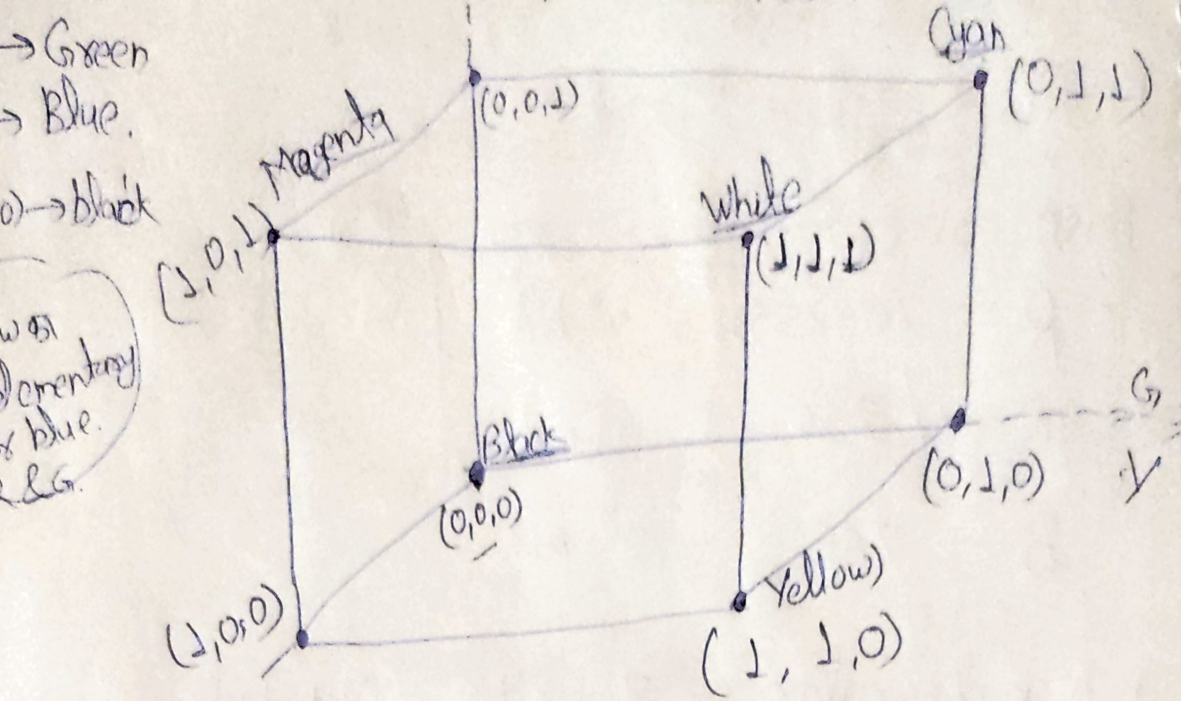
⇒ 16777216 ⟶ Total color represent using RGB

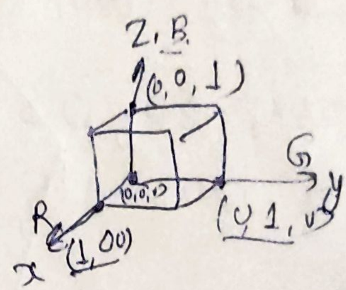
It is represent by using cube

X → Red
Y → Green
Z → Blue.
(0,0,0) → black

RBG color model

Yellow as complementary color blue.
∴ R & G.




Also known as
⇒ Additive color model

$R + B = (1,0,0) + (0,0,1)$
$= (1,0,1)$ ⇒ Magenta.

$B + G = (0,0,1) + (0,1,0)$
$= (0,1,1)$ ⇒ Cyan.

$\underline{R + G} = (1,0,0) + (0,1,0)$
$= (1,1,0)$ ⇒ Yellow

$WHITE = R + G + B$
$= (1,0,0) + (0,1,0) + (0,0,1)$
$= \underline{(1,1,1)}$

Black = (0,0,0)

[In co-ordinate 255]

⇒ **Tabular Representation :-**

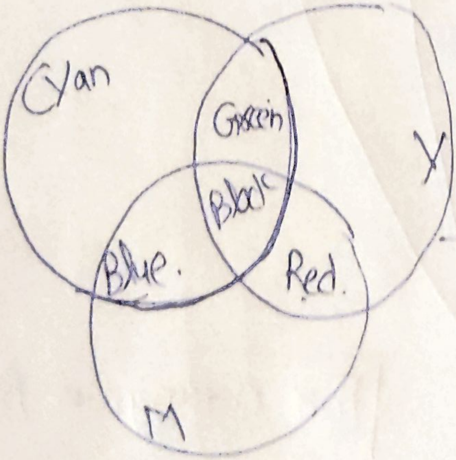
| | Actual range | White | Yellow | Red | Blue | Green | Magenta/Cyan |
|---|---|---|---|---|---|---|---|
| R | 0-255 | 255 | 255 | 250 | | | |
| G | 0-255 | 255 | 255 | 0 | | | |
| B | 0-255 | 255 | 0 | 0 | | | |

## ⚠ CMYK Color Model :- Opposite of RGB color model.

K = Black
C = Cyan
M = Magenta
Y = Yellow.

It is subtractive color model Means we have to substract one color to get another color.

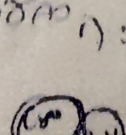
or complimentary color




→ Red get by combine Magenta & yellow.

Cyan color removes = Red.
Yellow — // = Blue.
Magenta — // — = Green

Green = Cyan + Yellow



(3) HSV Color Model:-

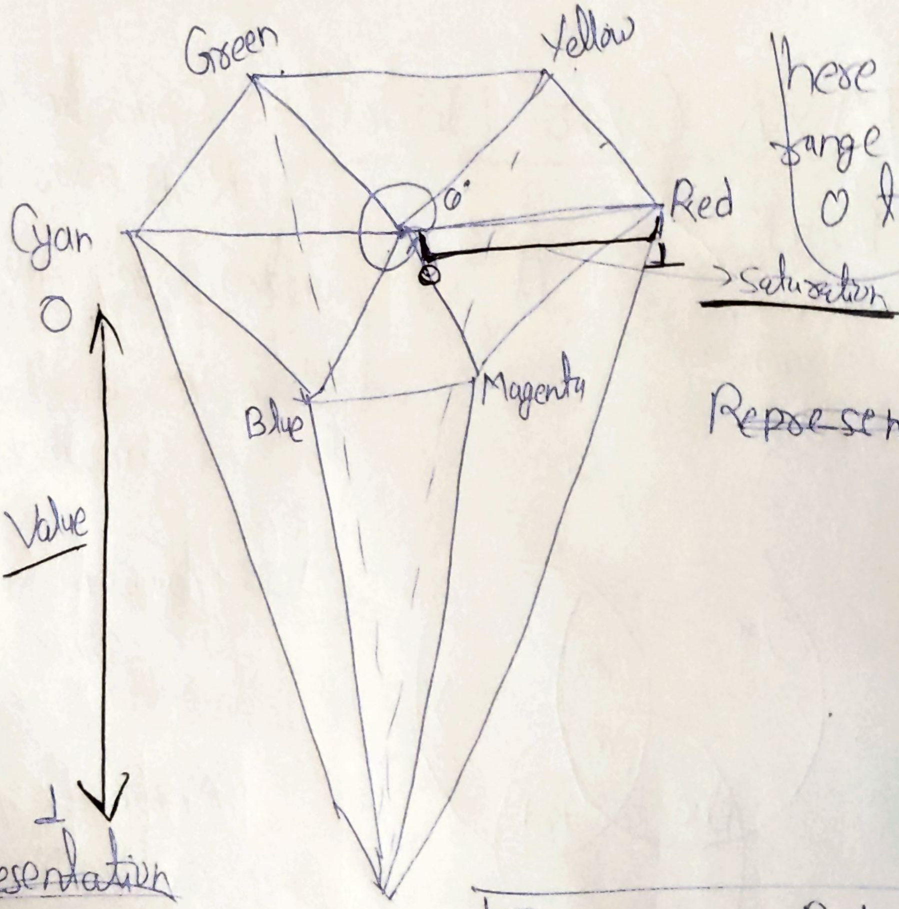
H = Hue. (color) (~~Range 0 to 1~~)

S = Saturation (How pure it is light/dark) (0) (1)

V = value (Intensity/ brightness) (0) (1)

Representation using hexagon:-

Orange = Red + Yellow




here θ is range from 0 to 360

~~Represent~~

Representation

(θ, 0–1, 0–1)

(H, S, V)

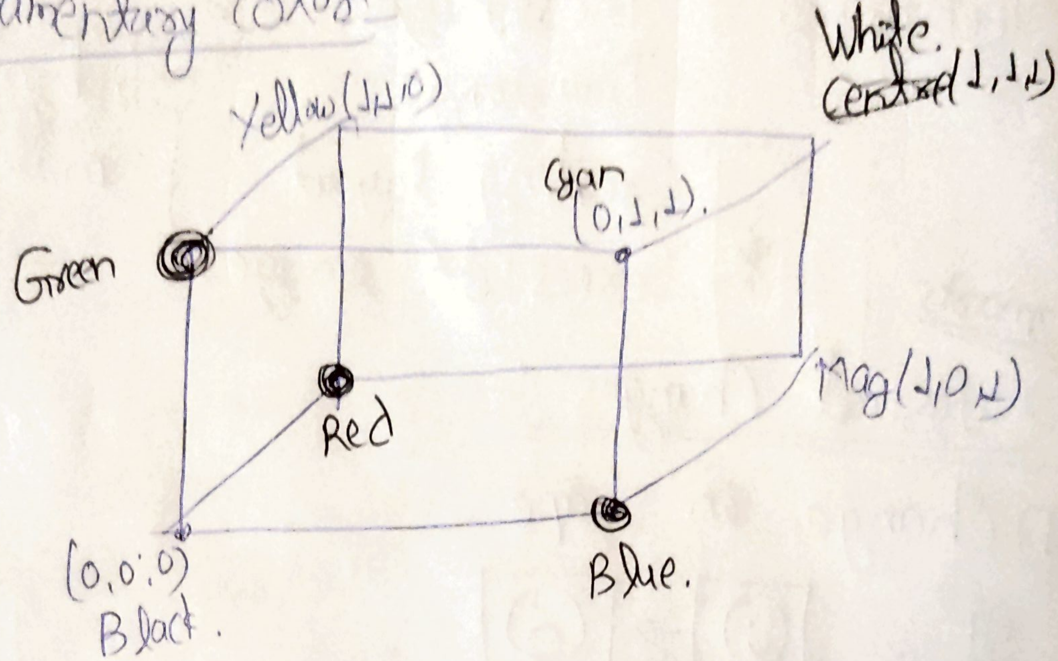
For orange = Red + Yellow

(30°

Que. (orange, pure, bright)

$(30^\circ, 0.5, 0.5)$ ~~or~~ $(30^\circ, 1, 1)$

⇒ Complimentary color



Date- 26/4/23

2 marks

**Animation** :- Movement on the screen by displaying series of still images

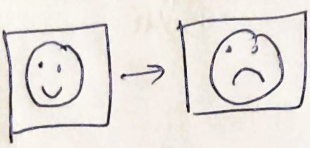
Technique - Designing
Drawing
Creating layout
& Series of images

} Integrated with multi-media

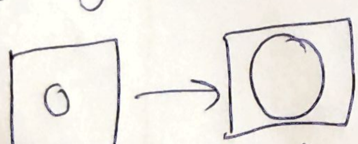
2 marks

⇒ **Visual Changes** :-

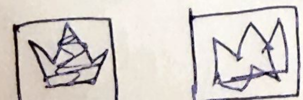
① Change in shape

eg. 

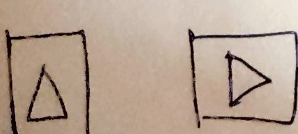
② Change in size



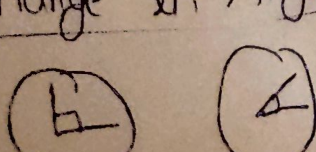
③ Change in color



④ Change in structure



⑤ Change in Angle :



⇒ Functions of Animation (features)

1. Morphing.
2. Wrapping
3. Twinning / In betweening
4. Zooming
5. Panning
6. Masking
7. Onion skinning
8. Motion Capture.
9. Motion editing
10. ~~Fractural~~ Fractals.

(1) Morphing :- Object transformation

Eg Human face converted into Animal face.

child face ⟶ old age face.

their are 3 steps:-

(i) Initial image generate & final image generate
(ii) ↳ Design key frame. ↳ Key Frame.
(iii) Transformation.

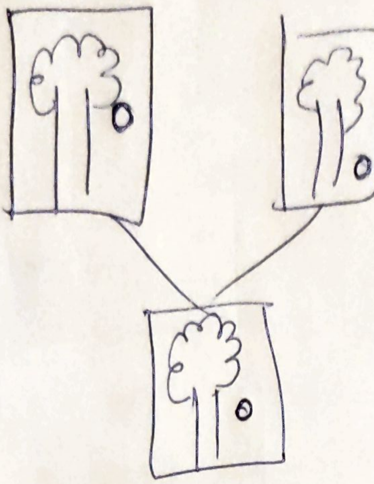
⇒ b/w key frame generate

② **Wrapping** :- (object trans).

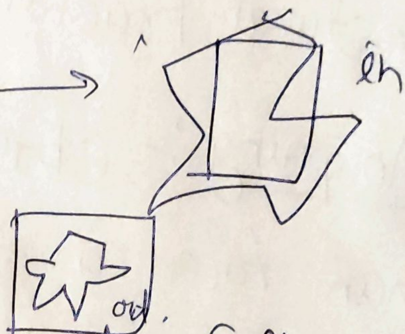
Initial image $\xrightarrow{\text{convert}}$ Final image.

here continuous key frame generate.

③ **Tuning / Inbetweening** :-

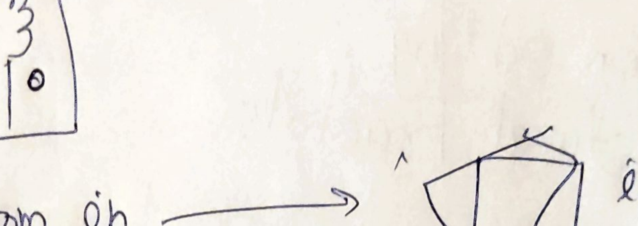


④ **Zooming** :- Zoom in
Zoom out




⑤ **Panning** :- Horizontal movement. of the window. either forward or backward.

⑥ **Masking** :-
protecting part of window is masking.




⑦

& allowing rest portion for changes.

⑦ Onion Skinning :- Effect is just like skin of onion

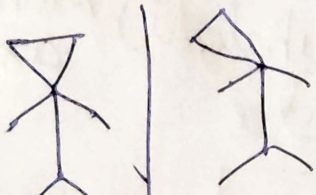
used to see all the sequence at a time.


⑧ Motion Capture :- Capture original human effect

Steps:-

(i) Capture human motion
(ii) Create key frame
(iii) Apply to object.

⑨ Motion Editing :- Capture motion & edit the motion.



⑩ Fractals :- Means Repetation of movements. like walking, climbing, swimming, crying

Forming a circle → the circle is repeated with small changes.

# Illumination & Shading Model
Date - 27/4/23

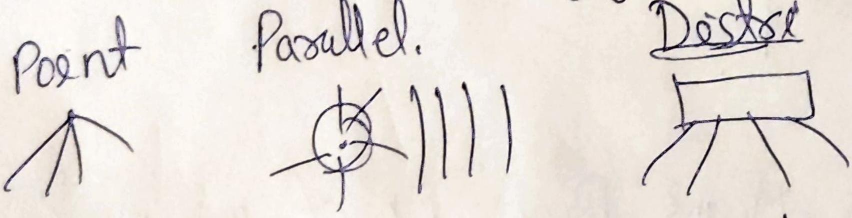
## Or lightening model :-

**uses:-** (i) To calculate intensity of light reflected from a point of the surface.

**Factors of illumination model :-**

① Light Source.
② Surface
③ Observer

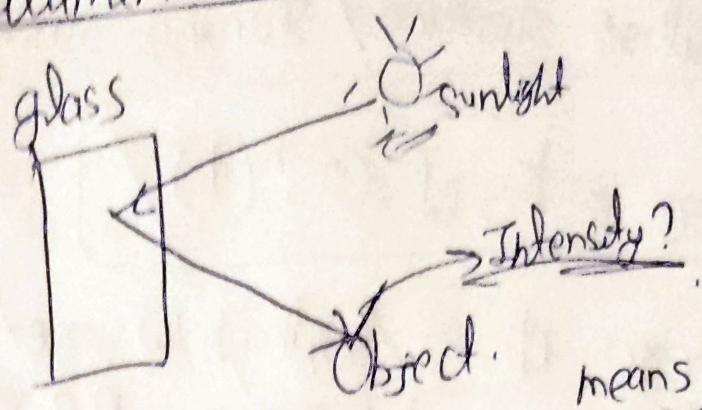
I) Light Source
- (a) Point Source (eg. bulb.) → from one point.
- (b) Parallel source (For infinite distance eg. sun)
- (c) Distributed source. (eg. tubelight) means finite area.




② **Surface** :- Any surface on which light fall.

For intensity

① Ambient Illumination / Indirect light :-




means calculate reflect.

Formula

$$\boxed{I_{amb} = K_a I_a}$$

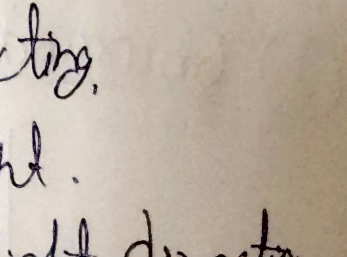
where $I_{amb}$ = Reflection light.

$K_a$ = Surface reflectivity.

$I_a$ = Intensity of ambient light

② Diffuse Reflection :- For Rough surfaces

⇒ Intensity (I) depends on θ.




$$\boxed{I_{diff} = K_d I_p \cos\theta.}$$

where $K_d$ = Surface reflecting.

$I_p$ = Intensity of light.

where $\cos\theta \Rightarrow N.L.$ = Normal vector, light direction

2 marks

## (3) Specular Reflection :-

reflect. water

↳ Glass surface / Shining surface.

$$\boxed{I_{spec} = K_s I \cos^n(\phi)}$$

where $\phi$ = Angle b/w reflected ray & viewer R & V

$K_s$ = Reflectivity index.

## ✭ Shading Model :-

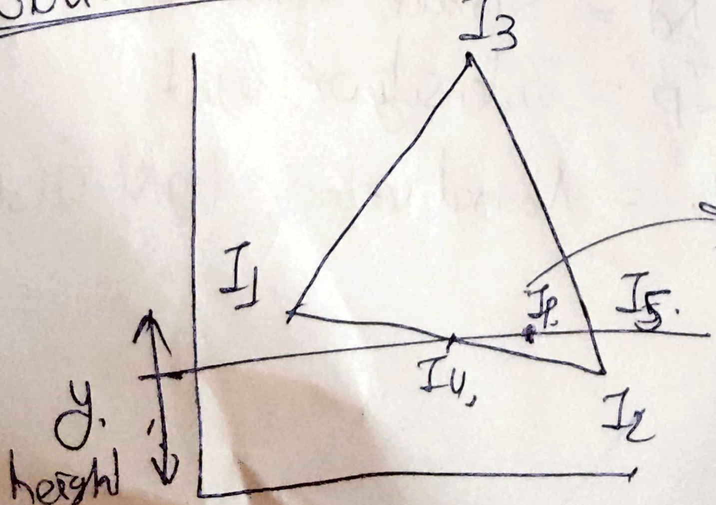
1. Constant / Flat / Faceted shading.
2. Gouraud shading
3. Phong Shading.

### (1) Constant / Flat / Faceted shading :-

Object is converted into polyhedra (collection of polygon).

### (2) Gouraud Shading :-

process is




here y is constant so take x,

Step (i) Intensity of $I_1, I_2, I_3$ using linear interpolation.

(ii) Take a scanline.

(iii) Calculate $I_4$ using $I_1$ & $I_2$.

(iv) Calculate $I_5$ using $I_2$ & $I_3$.

(v) Calculate $I_p$ using $I_4$ & $I_5$.

$$\boxed{I_4 = I_1 * \frac{Y_4 - Y_2}{Y_1 - Y_2} + I_2 * \frac{Y_1 - Y_4}{Y_1 - Y_2}}$$

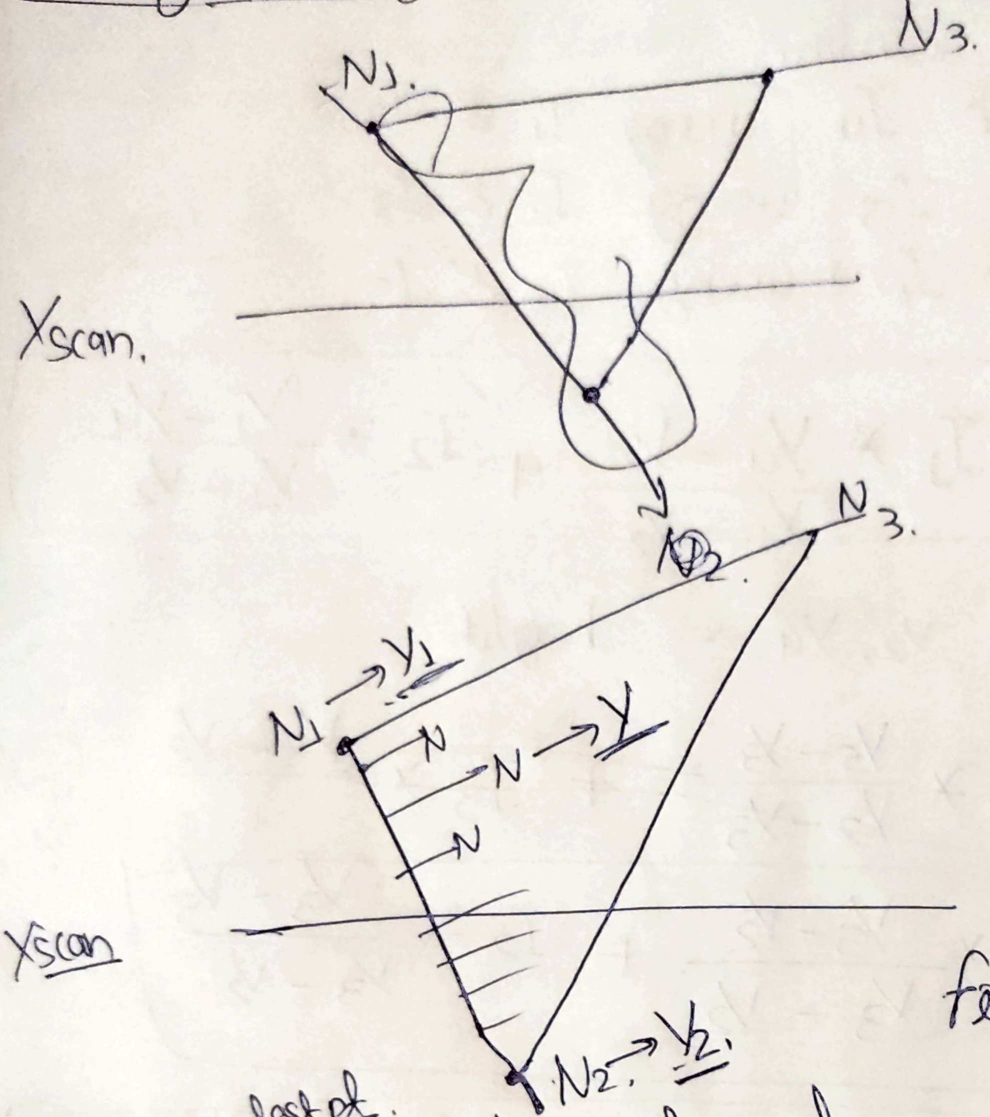
where $Y_1, Y_2, Y_4$ = height.

$$\cancel{I_5 = I_2 * \frac{Y_5 - Y_3}{Y_2 - Y_3} + I_3 * \frac{Y_5 - Y_2}{}}$$

$$\boxed{I_5 = I_3 * \frac{Y_5 - Y_2}{Y_3 - Y_2} + I_2 * \frac{Y_3 - Y_5}{Y_3 - Y_2}}$$

$$\boxed{I_p = I_4 * \frac{X_p - X_5}{X_4 - X_5} + I_5 * \frac{X_4 - X_p}{X_4 - X_5}}$$

③ **Phong shading** :- Based on. Normal vector.




y = height

first calculate.
Normal vector

finally last pt. calculate intensity.

**Calculation for normal:-**

$$\boxed{N = \frac{Y - Y_2}{X_1 - Y_2} * N_1 + \frac{Y_1 - Y}{Y_1 - Y_2} * N_2}$$

## Input Devices:-

Keyboard, Mouse, Light pen, touch-screen, touch-pad, joystick, CTR, Cathod, ray tube.