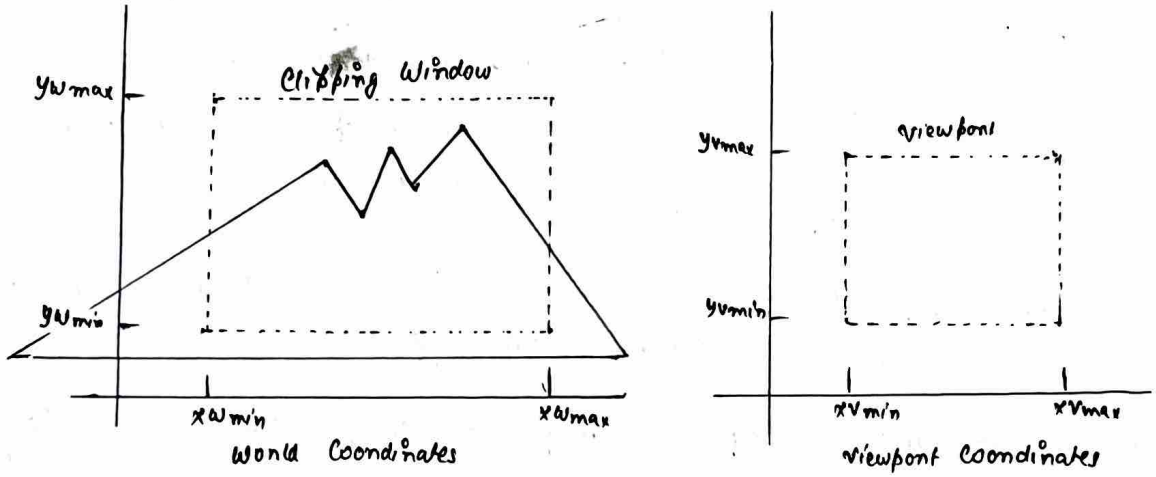
Name - Humaira Imran Haque
USN - 1BY20CS118
Branch - CSE B section

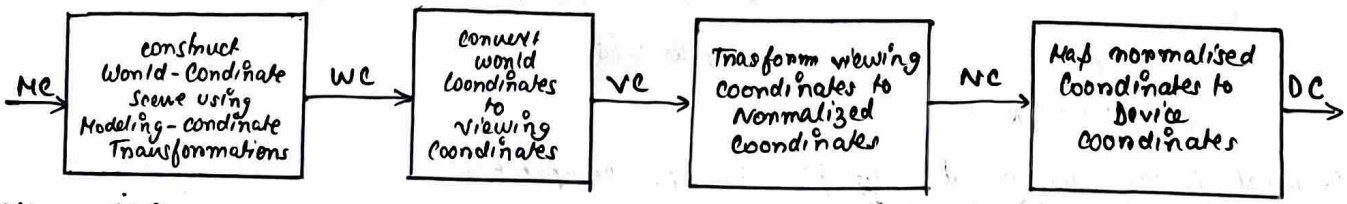
(1) Build a 2D viewing transformation pipeline and also explain OpenGL 2D viewing functions

⇒ Clipping window: a section of 2D scence that is selected for display is called clipping window.

- clipping window is also alluded to as world window on viewing window
- Graphic packages allow us to control the placement within the display window using another window called as viewpoint.



- The viewpoint indicates where the selected object is to be viewed on the output device.
- By changing the position of the viewpoint we can view the objects at different positions on the display area of an output device
- Usually clipping window and viewpoints are rectangles in standard position where the rectangle edges are parallel to the coordinate axes.



The mapping of 2D, world coordinate scence description to device coordinates is called a 2D viewing transformation. This transformation is also called as window to viewport transformation.

Here are some key functions used in OpenGL for 2D viewing:

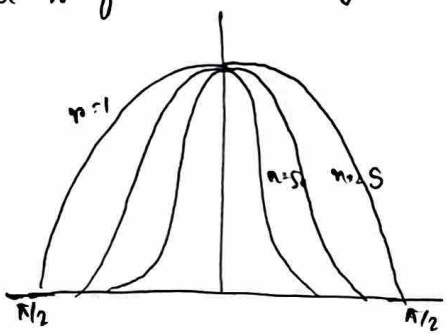
1. 'glMatrixMode()': Sets the current matrix mode to define whether subsequent matrix operations affect the model view of the projection matrix
2. 'glLoadIdentity()': replaces the current matrix with an identity matrix.

3. gl Ortho(): specifies the orthographic projection matrix.

4. glViewpoint(): sets the viewpoint transformation

(2) Build Phong Lighting Model with equation

Phong reflection is an empirical model of local illumination it describes Phong the way a surface reflect light as combination of diffuse reflection of rough surface with specular of shining surface. It is based on Phong's informal observation that shiny surface have small intense specular highlights while dull surface have larger highlights that off gradually



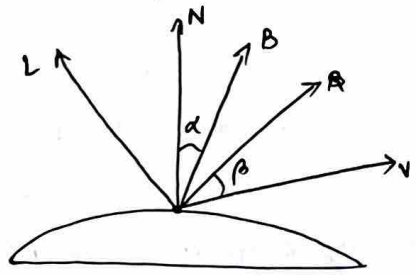
- Phong model sets the intensity of specular reflection of $\cos^{ns}\phi$

$I_{specular} = w(\theta)\, I_l \cos^{ns}\phi$

$0 \le w(\theta) \le 1$ is specular reflection coefficient

If light direction L. and viewing direction V are some side of normal on If h is behind the surface do not have any effect.

For most opaque material opacular reflection coefficient is nearly ks



$$I_{specular} = \begin{cases} k_s I_l (VR)^{ns} & VR > 0 \text{ and } N \cdot L > 0 \\ 0 & \text{otherwise} \end{cases}$$

$R = (2NL)N - L$

Normal N way vary at each pt. to pts computation.

coefficient computations $H = \frac{L+V}{|L+V|}$

If is the distance between source and viewer are relatively for then α is constant. It is direction yielding maximum specular reflection in viewing direction V if surface normal N would coincide with H. If V is coplanar with R and L the $\alpha = \phi/2$

(3) Apply homogeneous coordinate for transition, rotation and scaling w.r.a
=> matrix representation Translation. It moves all points in an object along some straight line path to new position.

path is represented by vector

$P_x' = P_x + t_x$

$P'_y = P_y + t_y$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

In homogeneous coordinates:

$(x_h\ y_h\ h)$

$x = \frac{x_h}{h}$

$x_h = x * h \qquad y_h = y * h$

consider $h = 1$

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

Rotation:

It repositions all points in an obj. along a circular path in plane centered at pivot

center

$\cos\phi = x/r \qquad \sin\phi = y/r$

$x = r\cos\phi \qquad y = r\sin\phi$

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

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

$P' = R \cdot P$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

In homogeneous coordinate

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

Scaling: It is used to change the size of an object and involve two scaling factor s_x & s_y

$P'_x = s_x \cdot P_x$

$P'_y = s_y \cdot P_y$

P

$$P' = S \cdot P$$

$$S = \begin{bmatrix} Sx & 0 \\ 0 & sy \end{bmatrix}$$

Homogeneous coordinate $= \begin{bmatrix} Sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$

(4) Outline difference between raster scan display and random scan displays.

Raster scan display:	Random scan display
• Electron beam is swept across one row data from top to bottom.	
• As it moves across each row, beam intensity is turned on & off to create a pattern of illuminated spots.	
• Scaning process called refreshing. Complete scaning of screen is normally called frame.	

Random Scan Display

- When operated as random scan display unit CRT has election beam distributed only to those pairs of screen where picture to displayed.
- Pictures are generated as line drawing with electron beam tracing out the component lines one after the other.
- Also known as vector displays. Comment line of picture can be drawn & refreshed by a random scan system in any specified order
- Refresh rate depends on number of lines to be displayed on that system.

(5) Display window management using GLUT

=> Step1: Initialization of glut

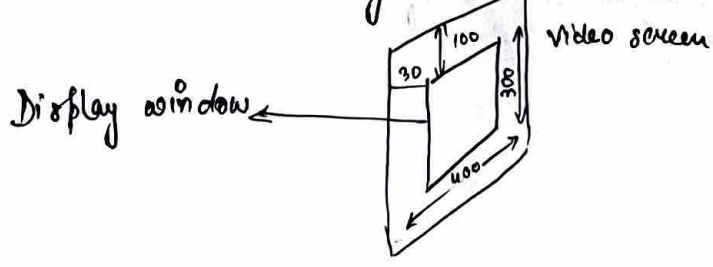
- We use GLUT
- We perform initialization —

glutInit(&argc, argv)

Step2: Title

glutCreateWindow("An example");

There is single argument for this function can be any character string that we want character string that we want to display for window



glutDisplayFunc (lineSegment):
It is used to call the display function repeatedly. Name of the function to be called line segment.

glutMainLoop():
It is last line. It displays the initial graphic and put program to infinite loop.

glutInitWindowPosition:
It is used to specify upper left corner of the display window.

glutInitWindowPosition:
It is used to set initial fixed with height & width of display window.

glutInitDisplayMode (GLUT SINGLE | GLUT_RGB):
It is used for and color mode like red, green & blue component to select colors value.

(6) Explain OpenGL visibility detection function

=> a. OpenGL polygon culling function:

Back face removal with functions
glEnable (GL_CULL_FACE)
glCullFace (mode)
mode can be GL_Back, GL_FRONT, GL_FRONT_AND_BACK

Disable with
glEnable (GL_CULL_FACE),

b. OpenGL Depth Buffer functions:
To use OpenGL depth buffer visibility detection function, we need to modify GLUT initialization function.

glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
glClear (GL_DEPTH_BUFFER_BIT);

Display the depth test
glDisable (GL_DEPTH_TEST);

we can set states of depth buffer so that it is only in read state or read write state
glDepthMask (write status);

C. OpenGL wireframe surface visibility method:

A wire frame display can be obtained in OpenGL by requesting that only its edge are generated
glPolyModel (GL_FRONT_AND_BACK, GL_LINE);

d. OPENGL-DEPTH_CURING-function.

It is used to vary the brightness of an object as a function of its distance from viewing position with.

```
glEnable(GL_FOG);
glFogi(GL_FOG_MODE, GL_LINEAR);
```

It applies to depth in domin = 0.0 and damax = 1.0 and set different values for dmin & dmax

```
glFogf(GL_FOG_START, minDepth);
glFogf(GL_FOG_END, maxDepth);
```

(7) Write special cases that we discussed with respect to perspective projection transformation coordinates.

$$x_p = x\left[\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right] + x_{prp}\left[\frac{z_{prp} - z}{z_{prp} - z}\right]$$

$$y_p = y\left[\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right] + y_{prp}\left[\frac{z_{prp} - z}{z_{prp} - z}\right]$$

cases

1) Projection reference point is limited along z view axis $- x_{prp} = y_{prp} = 0$.

$$x_p = x\left[\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right] \qquad y_p = y\left[\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right]$$

2) When projection reference point is at coordinate origin.

$(x_{prp}, y_{prp}, z_{prp}) = (0, 0, 0)$

$$x_p = x\left[\frac{x_{prp}}{z}\right) \quad y_p = y\left(\frac{y_{vp}}{z}\right)$$

3) If view plane is uv plane and no restrictions and placement of projection reference point.

$z_{vp} = 0$

$$x_p = x\left[\frac{z_{prp}}{z_{prp} - z}\right] - x_{prp}\left[\frac{z}{z_{prp} - z}\right]$$

$$y_p = y\left[\frac{z_{prp}}{z_{prp} - z}\right] - y_{prp}\left[\frac{z}{z_{prp} - z}\right]$$

4. If uv plane is a projection reference point on z view axis

$x_{prp} = y_{prp} = z_{vp} = 0$

$$x_p = x\left[\frac{z_{prp}}{z_{prp} - z}\right] \qquad y_p = y\left[\frac{z_{prp}}{z_{prp} - z}\right]$$

(8) Explain Bezier curve equation along with equation along with properties

* Developed by French Engineer Pierre Bezier for use in design of Renault automobile bodies.
* It has number of properties that make them highly useful for curve and surface design.

It can be fitted to any number of control points.

Equation:

$P_k (x_k, y_k, z_k)$ P_k = general (n+1) control point positions.

P_u = position vector that describes path

$$P(u) = \sum_{k=0}^{n} P_k \, BEZ_{k,n}(u)$$

$BEZ_{k,n}(u)$ $C(n,k)\, u^k (1-u)^{n-k}$ P.S Bernstein polynomial

$$C(n,k) = \frac{n!}{k!(n-k)!}$$

Properties:

- Basic function are real
- Degree of polynomial is one less than number of control point
- Curve generally follows shape of defining polygon.
- It connects first & last control points

$P(0) = P_0$

$P(1) = P_n$

- curve lies within convex hull of control points.

(9) Explain normalisation transformation for Orthogonal projection.

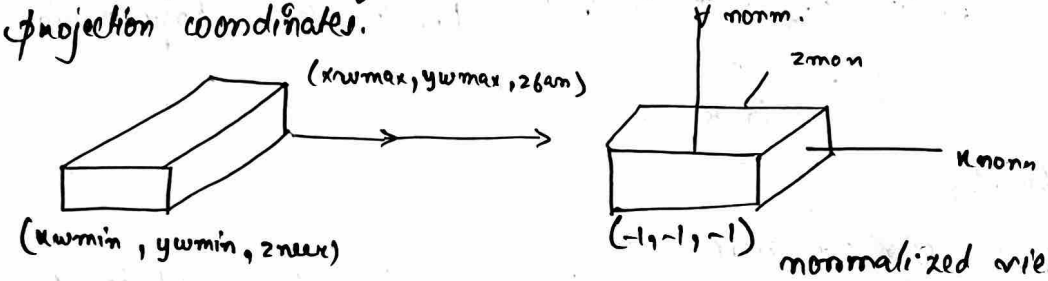
=> We assume the orthogonal projection view volum to mapped into symmetric normalization within left-handed reference frame. Also z-condinate positions for handed reference frame. Also x-condinate for near & far position is denoted as z_{near} & z_{far} respectively this position is mapped to normalized position (-1, -1, -1) and position $(x_{max}, y_{max}, z_{max})$ is mapped to (1, 1, 1)

Transforming rectangular parallel piped view volume to normalized cube is similar to method for converting the clipping method into normalized symmetric square

Normalization transformation for view volume

$$M_{ortho,norm} = \begin{bmatrix} \frac{2}{x_{wmax}-x_{wmin}} & 0 & 0 & \frac{x_{wmax}+x_{wmin}}{x_{wmax}-x_{wmin}} \\ 0 & \frac{2}{y_{max}-y_{min}} & 0 & \frac{-y_{wmax}+y_{wmin}}{y_{wmax}-y_{wmin}} \\ 0 & 0 & \frac{-2}{z_{near}-z_{far}} & \frac{z_{near}+z_{far}}{z_{near}-z_{far}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrix is multiplied on right by composite viewing transformation R.T to produce complete transformation from world coordinates to normalize orthogonal projection coordinates.



normalized view volume

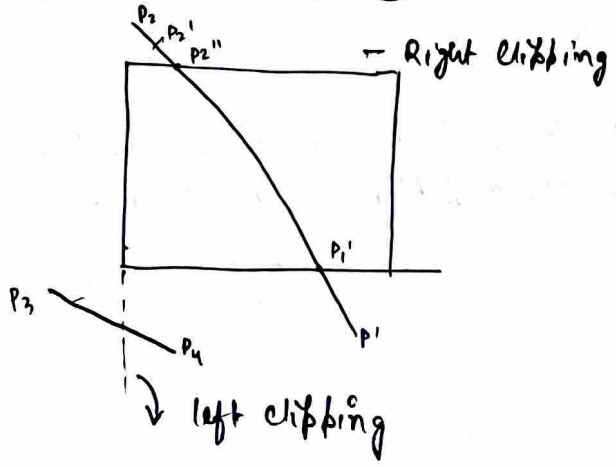
(10) Explain Cohen Sutherland line clipping algorithm.

⇒ Every line endpoint in picture is designed with four digit binary code called region code and each bit is used to indicate where point lies inside or outside

1001	1000	1010
0001	0000	0010
0101	0100	0110

Once we established region code for all line endpoint we determine where they lie completely inside or not.

When or operation b/w 2 end points is false, line is inside window AND operation b/w 2 endpoints is true, completely outside clipping window If it is not completely inside, it lies partially outside



Intersection P_2 and P_2' or P_2'' P_5 clipped off. For line P_0 to P_4 we find that point is outside left boundary P_4 is inside. Therefore intersection is P_3 and P_3' to P_3' is clipped off.

By ducking the region code P_3' and P_4 we find the remainder of line is below clipping window & can be eliminated. To determine a boundary for line eqn the y-condinate intersection point with vertical clipping line can be obtained by

$$y = y_0 + m(x - x_0)$$

$$m = y_{end} - y_0 / x_{end} - x_0$$

$$x = x_0 + \left(\frac{y - y_0}{m} \right)$$