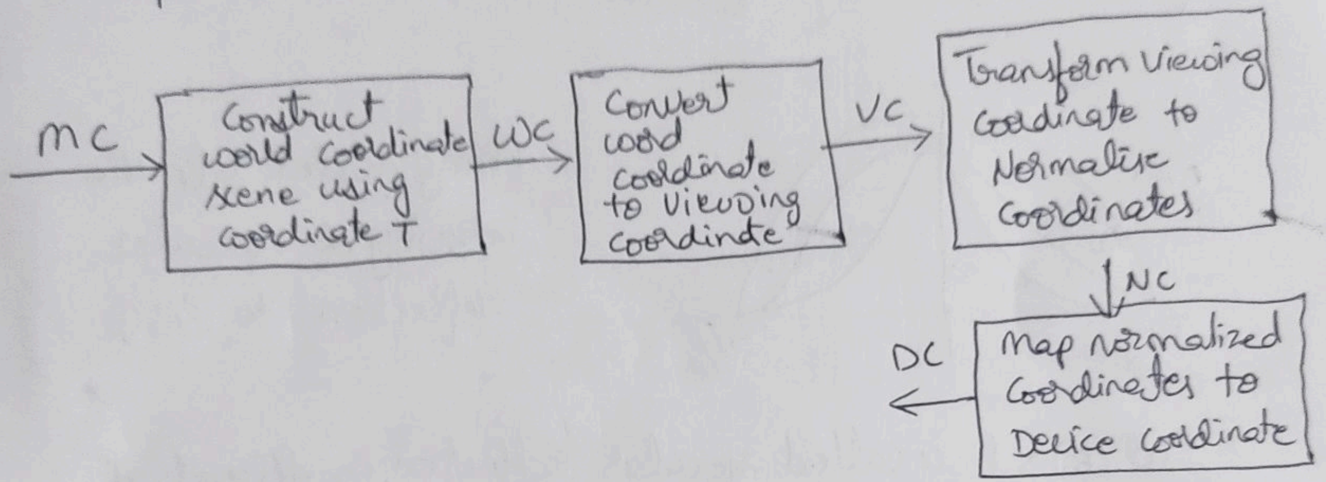
CGV ASSIGNMENT

Name :- Nandan N kopplu USN: 1BY20CS122 CSE 6 'B'

① Build a 2D viewing transformation pipeline & also explain OPEN GL 2D viewing functions



A section of 2D scene that is selected for display is called a clipping window because all parts of scene outside the selected section are "clipped off

Depending upon graphics library, the viewport is defined in normalised coordinates (or) screen co-ordinates. At the final step of viewing transformation the confronts of viewport are transfered to portions within the display window.

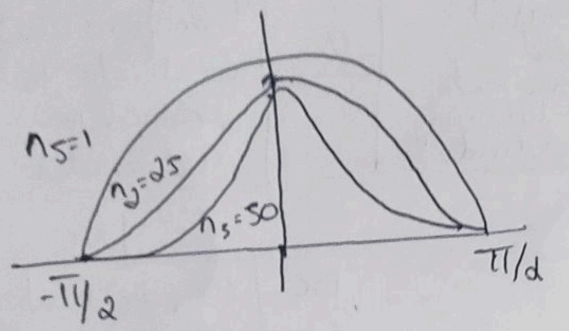
The openGL 2D viewing function in openGL projection mode Before we should start a Viewport to OpenGL.

glMatrixMode (GL-PROJECTION);

glViewport (X_{vmin}, Y_{vmin}, V_{pwidth}, $V_{pheight}$);

2. Build phong lighting model with equations

Phong reflection is a empirical mode of local illumination. It describes the way of surface reflects light as a diffuse reflection of rough surfaces with the specular reflection of shining reflection



Phong model sets the intensity of specular reflection to $\cos\phi$

$0 \le \omega(\theta) \le 1$ is called specular reflection co-efficent of light direction & viewing direction

```
glutInitWindowPosition (xTopleft, yTopleft);
glutInitWindowSize ( dwidth, dheight);
glutCreateWindow ("Title of Window");
```

3. Apply homogenous co-ordinates for translation, rotation and scaling via matrix representation.

→ A standard technique for accomplish 2D (or) 3D Transformation is to expand each two-dimensional coordinate

where ⇒ $x = \dfrac{x_h}{h}$, $y = \dfrac{y_h}{h}$

Translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \\ 1 \end{bmatrix}$$

$$P' = T(tx, ty) . P$$

Scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} . \quad P' = (Sx, Sy) . P$$

Ration -

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

$$P' = R(\theta) . P$$

4. Difference between raster scan display and random display

Random scan	Raster scan
• It has high resolution	• It is low resolution
• It is more expensive	• It is less expensive
• Easier to modify	• Modification is Tough
• Solid pattern tough to fill	• Easy to fill solid pattern
• Refresh rate depends on resolution	• Does not depend on pictures.
• Beam penestration technology comes under it	• Shadow mask technology comes under it.

(5.) Demonstrate OpenGL function for display window management using GLUT

→ We perform the GLUT initialisation

glutInit (&argc, argv)

Next, we can state that display window is to be created on screen

glutCreateWindow ("AN Example");

The following function call passed the line segment

glutDisplayFunc (line-segment);

The function must be last one in program. It puts the device in infinite loops that

glutMainLoop();

(6) OpenGL Visibility functions

→ a) OpenGL polygon curving functions

Backface removal with functions

glEnable (GL-CULL-FACE);

glCullFace (mode);

Disable with

glDisable (GL-CULLFACE);

glutInitDisplayMode (GLUT-SINGLE | GLUT-DEPTH

(7) Write special cases that we discussed wrt perspective projection transformation coordinates.

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

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

Cases:

(i) Projection reference point is limited along z-view axis:

$$x_{prp} = y_{prp} = 0 \;;\quad x_p = x\left[\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right] \;;\quad y_p = y\left[\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right]$$

(ii) when projection reference point is at co-ordinate origin

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

$$x_p = x\left(\frac{z_{vp}}{z}\right), \quad y_p = y\left(\frac{z_{vp}}{z}\right)$$

(iii) If view plane is uv plane:

$$z_{vp} = 0, \quad x_p = x\left[\frac{z_{prp}}{z_{prp} - z}\right] - x_{prp}\left[\frac{z}{z_{prp} - z}\right] \quad y_p = [y]\left[\frac{z_{vp}}{z_{prp} - z}\right]$$

(iv) If view plane is in uvw plane and restriction on reference point:

$$x_{prp} = y_{prp} = z_{vp} = 0 \;;\quad x_p = x\left[\frac{z_{prp}}{z_{prp} - z}\right] - x_{prp}\left[\frac{z}{z_{prp} - z}\right]$$

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

---

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

Sol^n Developed by French engineer Pierre Bezier for use in 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 position P_K = Position vector that describes path

$$P(u) = \sum_{k=0}^{n} P_k \cdot BEZ_{k/n}(u) \quad BEZ_{k/n}(u)\, C_{(n,k)}\, u^k (1-u)^{n-k}$$

is Bezier polynomial.

(Q) Explain Normalization transformation for orthogonal projection

we assume that orthogonal projection view volume to mapped into symmetric normalization cube with left-handed reference frame. Also Z-co-ordinate position for handed.

reference frame

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