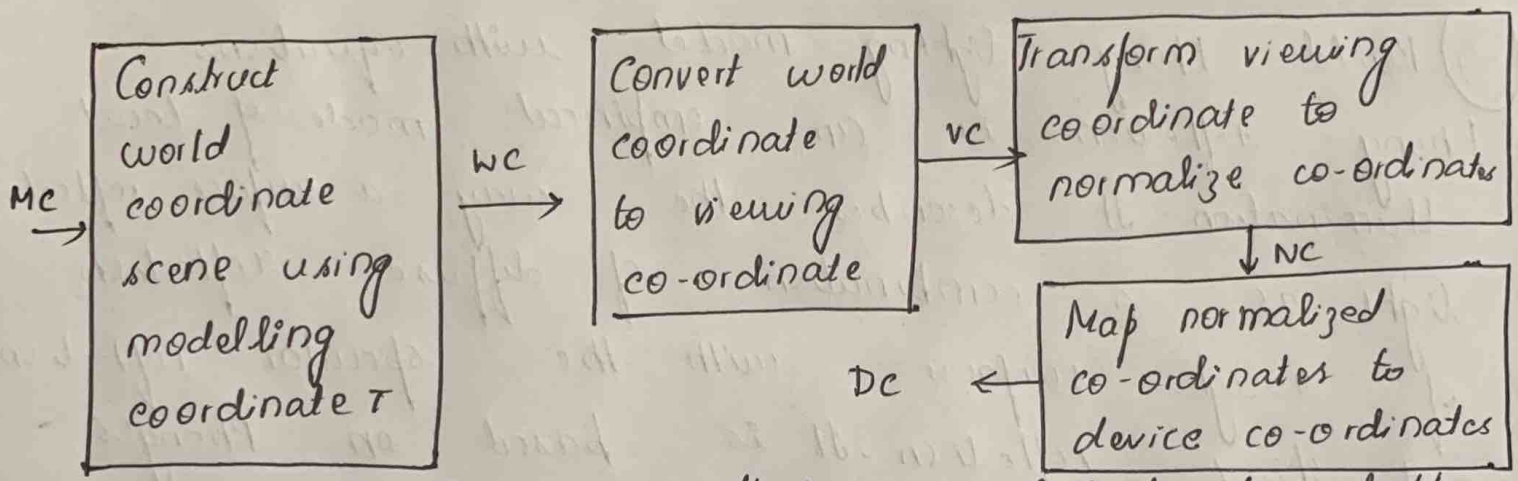
Name: Nithya. H.S
USN: 1BY20CS130
CSE-B SEC

(1) Build a 2D viewing transformation pipeline and also explain OPENGL 2D Viewing function



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

The mapping of a 2D world co-ordinate (wc) description to device co-ordinates (DC) is called a 2D Viewing transformation. Sometimes the transformation is simply referred to as window to viewport transformations

Once the world co-ordinate scene has been constructed we could set up a seperate 2D viewing co-ordinate system for specifying the clipping window.

Depending upon graphics library, the viewport is defined in normalized co-ordinates as screen co-ordinates. At the final step normalized step

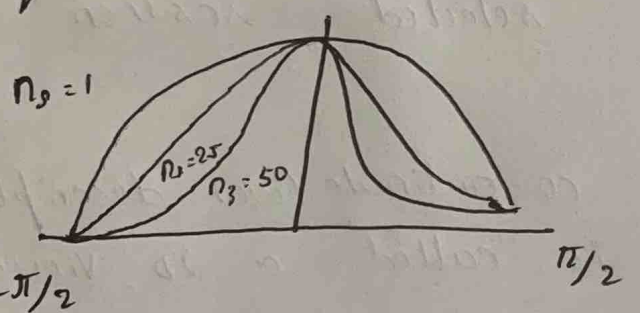
viewing transformation the contents of viewport are transferred to partitions within the display window

The OpenGL 2D viewing function in OpenGL projection mode. GLU clipping window function :-

glViewport(xrmin, yrmin, Vpwidth, Vpheight);

(2) Build Phong lighting model with equations

Phong reflection is an empirical mode of local illumination - It describes the way a surface reflects light as a combination of diffuse reflection of rough surfaces with the specular reflection of shiny reflection - It is based on Phong's informal observation.



Phong model sets the intensity of specular reflection to $\cos^{n_s}\phi$

$I_l, \text{specular} = w(\theta) I_l \cos^{n_s}\phi$

$0 \leq w(\theta) \leq 1$ is called specular reflection coefficient. If light direction & viewing direction v are on some side of normal v or if L is behind the surface, specular effects do not exist

We have 3 functions in GLUT for display window

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

(3) Apply Homogenous co-ordinates for translation, rotation and scaling via matrix representation (2)

A standard technique for accomplishing 2D or 3D transformation is to expand each two-dimensional co-ordinate-position representation (x, y) to a 3D representation (x_h, y_h, z_h) called homogenous co-ordinates

where $x = \frac{x_h}{h}$ $y = \frac{y_h}{h}$

Translation

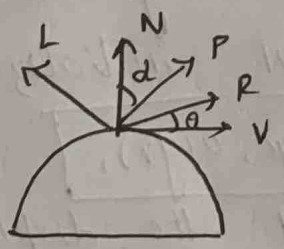
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad P' = T(t_x, t_y) \cdot P$$

Scaling

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad P' = S(S_x, S_y) \cdot P$$

Rotation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad P' = R(\theta) \cdot P$$



$$I_s, \text{specular} = \begin{cases} k_s I_l (V \cdot R)^{n_s} & , V, R > 0 \ \& \ N \cdot L > 0 \\ 0 & \text{otherwise} \end{cases}$$

$R = (2N \cdot L)N - L$ The normal N may vary at each point. To avoid N computations, angle ϕ is replaced by angle α defined by a halfway vector H

between α and γ

$$H = \frac{\alpha + V}{|\alpha + V|}$$

If light source and viewer are relatively far from object, α is constant.

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

Random Scan	Raster Scan
* It has high resolution	* Its resolution is low
* It has more expense	* less expensive
* Easier to modify	* Modification is tough
* Solid pattern, tough to fill	* Does not depend on pictures
* Refresh rate depends on resolution	* Easy to fill pattern
* Bean penetration technology comes under it	* Shadow mark technology comes under it

(5) Demonstrate OpenGL function for displaying window management using GLUT

We perform the GLUT initialization with statement

```
glutInit(&argc, argv);
```

Next, we can state that the display window is to be created on screen with a given title ~~equation~~ for title bar. This is accomplished with function

```
glutCreateWindow("An Example");
```

When the single argument for this function can be any character string that we want to use for

This function calls passes the line segment description to display the window

glutDisplayFunc(lineSegment);

This function must be last one in the program. Displays / Keeps the result displayed continuously

glutMainLoop();

glutInitWindowPosition(50,100);

Specifies ~~that~~ a single window and its position.

(6) Explain OpenGL visibility Detection functions

a. OpenGL polygon Curring functions:-

Back face removal with functions

glEnable(GL_CULL_FACE); glCullFace(mode);

mode can be GL_Back, GL_FRONT, GL_FRONT_AND_BACK

Disable with glDisable(GL_CULLFACE);

b. OpenGL DepthBuffer functions:-

To use OpenGL depth Buffer Visibility detect, on function we need to modify GLUT initialization function

glutInitDisplayMode(GLUT_SINGLE | GLUT_DEPTH);

glClear(GL_DEPTH_BUFFER_BIT);

Disable with glDisable(GL_DEPTH_TEST);

We can set states of Depth Buffer so that it is only in read state

c. OpenGL Wireframe surface visibility method

A wireframe display can be obtained in OpenGL by representing that only its edges are generated

glPolygonMode(GL_FRONT_AND_BACK, GL_LINE)

d. OpenGL Depth Cuing functions :-

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

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

applies to depth i.e $d_{min} = 0.0$ and $d_{max} = 1.0$ and set different values for d_{min} and d_{max}

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

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

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

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

Cases

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

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

(ii) When projection reference point is at co-ordinate assign $(x_{prp}, y_{prp}, z_{prp}) = (0, 0, 0)$

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

(iii) If view plane is uv plane and no restriction on placement of projection reference point

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

(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 and propert properties

Developed by French engineer Pierre Bezier for use in design. It can be filled to any number of control points

Equation: $P_k = (x_k, y_k, z_k)$ $\quad P_k$ = generate $(n+1)$ control point position

P_k = position vector that describes path

$P(t) = \sum_{k=0}^{n} P_k B \in \Sigma_{k/n}(y)$ $\quad B \in Z_{k/n}(y) C_{(n,k)} u^k (1-u)^{n-}$

is Bezier polynomial

Properties :- curve lies within numbes of control points

Basic functions are real, polynomial degree is one less than control points.

(9) Explain Normalization transformation for Orthogonal projection

We assume that orthogonal projection view volume to mapped into symmetric normalization cube within left-handed reference frame. Also z-coordinate position for handed reference frame. Also z-co-ordinate for near & far position is denoted at z_{near} & z_{far}. This position $(x_{min}, y_{min}, z_{near})$ is mapped to (1,1,-1) normalization transformation for view volume

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

(10) Explain Cohen-Sutherland line Clipping algorithm

Every line end point in picture is designed within 4 digits binary code called region code and each bit is used to indicate where the point lies inside

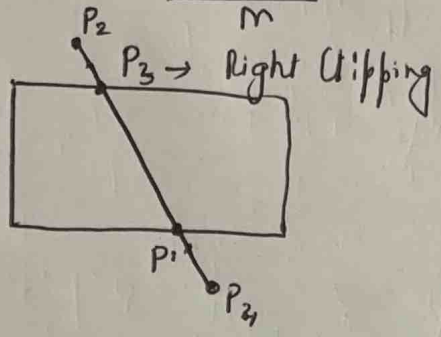
Once we establish the region code for all lines end-point we determine where the line lies Intersection P_2' & $P_2'P_2''$ is clipped off for line P_0 to P_4. we find that point is outside. left boundary P_4 is inside. Therefore intersection

P_3 & P_3' to P_3' is clipped off

By clipping the code R_3

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

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



1001	1000	1010
0001	0 00	0 010
0101	0100	0110

4 digit code for representation