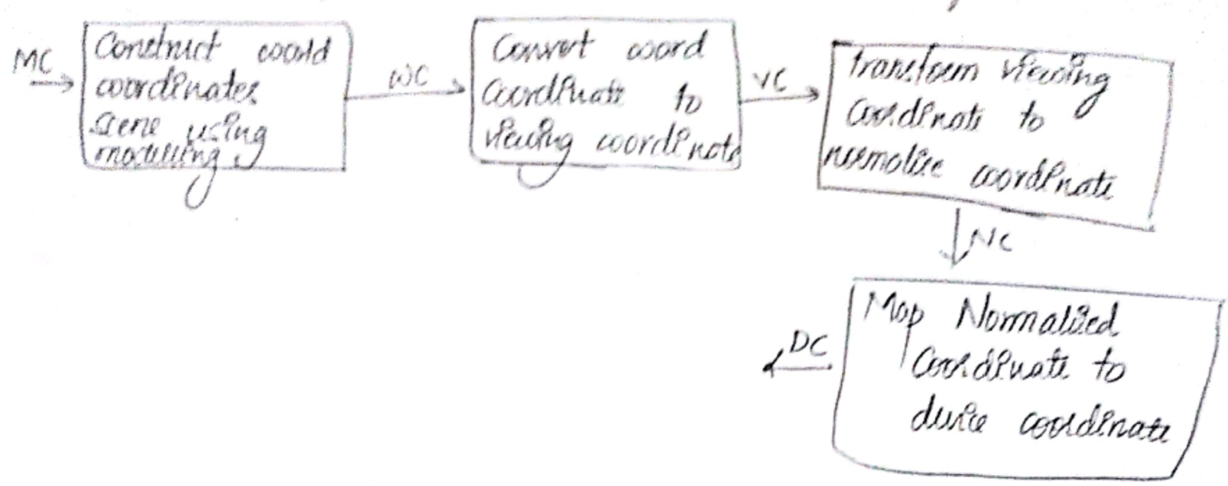
CGV assignment

1BI20CS101
M. Shashank.

1) Build a 2D viewing transformation pipeline & also Explain OpenGL 2D viewing



* a section of 2D scene that is selected for display is called as clipping window.

* Mapping of a 2D world coordinate description to device coordinate is called 2D viewing transformation.

* Once the world coordinate scene has been constructed, we would set up a seperate 2D viewing coordinate reference frame.

* Depending upon graphics library, the viewport is defined as normalised coordinates or screen coordinate reference frame.

* The OpenGL 2D viewing function in OpenGL projection mode before we should start a viewport to OpenGL

glMatrixMode (GL_PROJECTION);

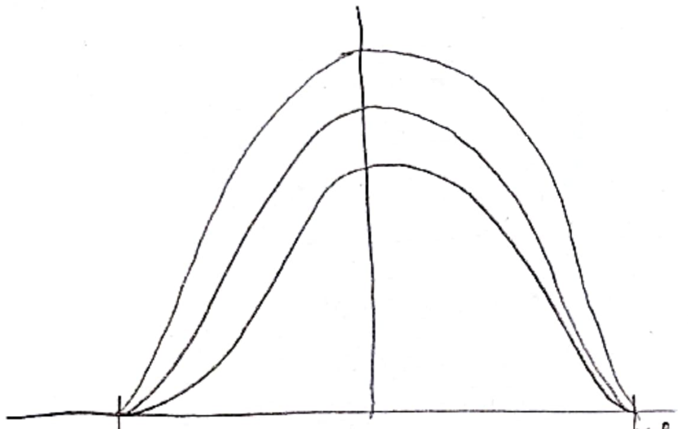
glu Clipping window Function:-

glViewPort (Xvmin, Ymin, Vpwidth, Vpheight);

② Build Phong lighting model with Equations

Ans

Phong reflection is an emphirical mode of local illumination. It describes the way of suface reflects light as a Combination of diffuse reflection of rough surfaces.



$-\pi/2$ $\pi/2$

Specular = $\omega(\theta) I_e \cos^n \phi$.

$0 < \omega \leq 1$, is called Specular reflection coefficient of light Viewing direction v are on the normal v or if L is behind the surface, specular exists do not exists.

we have 3 functions in GLUT:-

glut Init Window Postion (XTopLeft, Y TopLeft);

glut ~~Dlete~~ Create Window ("Title of window")

③ Apply Homogeneous Coordinate for transformation, rotation and scaling via matrix representation.

Ans: A standard technique for accomplish 2D or 3D transformation. is to Expand each two dimensional. coordinate position

(xh, yh, zh) is called homogeneous coordinate.

Translation:

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

$P' = T(t_x, t_y) P$

Scaling

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

$P' = S(S_x, S_y). P$

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

④ Outline difference b/w Raster Scanes & Random scan display

Random Scan	Raster Scan
* High Resolution	* Low Resoulution
* More Expensive	* less Expension
* Easily to modify	* Tough to modify.
* Solid patter tough to fill	* Easy to fill tough patteen

⑤ Demonstrate OpenGL function for display window management

Ans:

① we perform:

glutInit (&argv, &argc);

② to Create a Screen with a given caption for title bar,
we use: glutCreateWindow ("An Example");

③ When the single argument for this function can be any character string that we want to use for displaying windows

glutDisplayFunc (lineSegment);

④ To put device into infinite loop

glutMainLoop();

⑥ Explain OpenGL visibility Detection Function

a. OpenGL polygon Culling Function

* Back-face removal function

glEnable (GL_COLOR_FACE);

glCullFace (mode);

* mode values can be:-

GL_BACK

GL_FRONT

GL_FRONT_AND_BACK

* to disable the function:

glDisable (GL_CULLFACE);

b. OpenGL DepthBuffer functions

To use OpenGL depthBuffer visibility detection function we need to modify GLUT installaction function

glutInitDisplayMode (GLUT_SINGLE | GLUT_DEPTH);

glClear (GL_DEPTH_BUFFER_DEPTH);

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

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

Equation : $P_K = (x_K, y_K, z_K)$

P_K = generate $(n+1)$ control point position

P_K = position vector that describes path.

$P(K) = \sum_{K=0}^{n} P_K \, BEZ_{K,n}(u)$ $BEZ_{K,n}(u)$ $C(n,K)$

$u^K (1-u)^{n-K}$ is Bezier polynomial.

(9) Explain Normalization transformation for orthogonal projection.

Ans) 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. co-ordinate for near & far position is denoted as z_{near} & z_{far} respectively.

Matrix, norm =

$$\begin{bmatrix} \frac{2}{x_{wmax} - x_{wmin}} & 0 & 0 & -\frac{x_{wmax} + x_{wmin}}{x_{wmax} - x_{wmin}} \\ 0 & \frac{2}{y_{wmax} - y_{wmin}} & 0 & -\frac{y_{wmax} + y_{wmin}}{y_{wmax} - y_{wmin}} \\ 0 & 0 & \frac{-2}{z_{near} - z_{far}} & \frac{z_{near} + z_{far}}{z_{near} - z_{far}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

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

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

Cases

i) projection reference point is limited along z-view

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

ii) when projection reference point is at co-ords 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 and no restriction on placement of reference point.

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

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

iv) If view plane is in view plane and restriction on reference

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

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