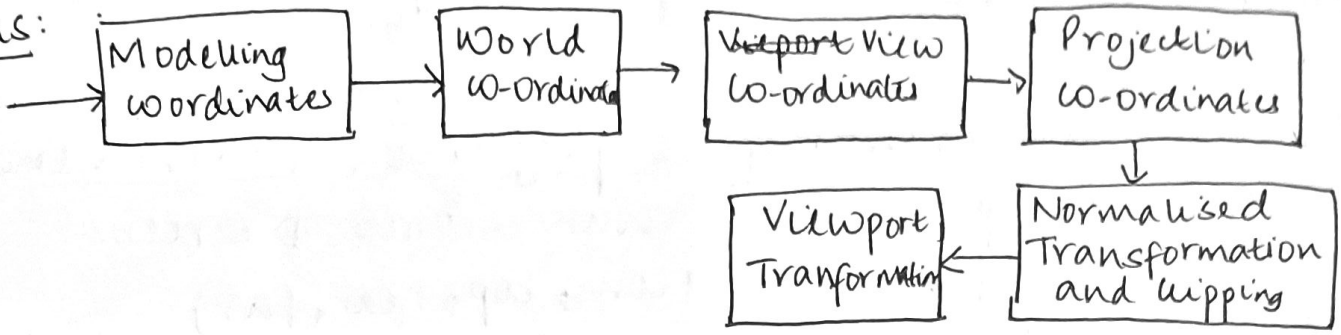
CG Assignment

Name: Megha
USN: 1BY20CS108
sem: 6th 'B'

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

Ans:



The 2D viewing transformation pipeline in computer graphics involves several steps to transform and display 2D objects on a 2D screen. Here are the main steps involved:

1) Object co-ordinate system: objects are initially defined in their own object co-ordinate system.
2) World co-ordinate system: The object co-ordinates are transformed into the world co-ordinate system. This step involves translation, rotation and scaling operations to position the objects appropriately in the world.
3) View co-ordinate system: The view co-ordinate system represents the viewers' perspective.
4) Projection co-ordinate system: Determines how the 3D view is projected onto a 2D screen.
5) Normalized co-ordinates: The projection co-ordinates are transformed into normalized device co-ordinates.

6) Viewport Transformation: The NDC co-ordinates are mapped onto the usual screen or viewport co-ordinates.

OpenGL 2D Viewing Functions:

In openGL, several functions are available to setup and control the 2D viewing transformation. Here are some commonly used functions:

- gluOrtho(left, right, bottom, top, near, far): Defines an orthographic projection matrix that maps the specified 3D volume onto 2D screen.
- glFrustum(left, right, bottom, top, near, far) Defines a perspective projection matrix using a frustum shaped viewing volume. Objects outside will be clipped.
- gluPerspective(fovy, aspect, near, far): specifying a perspective projection based on the vertical field of the viewing angle and aspect ratio.
- glViewport(x, y, width, height): Sets the viewport specifying the screen region where the rendering will appear.

2. Build Phong Lighting model with equations.

Ans: The Phong Lighting model is a widely used shading model in computer graphics that simulates interaction of light with surfaces. It consists of ambient, diffuse and specular lighting.

Ambient Lighting: The ambient components represents the overall illumination of a scene and its independent light source. It is calculated using the following equation:

$$La = Ka * La$$

La - resulting ambient light intensity
Ka - ambient reflection
La → ambient lighting color.

Diffuse Lighting: The diffuse component simulates the even scattering of light on rough services. It depends on the angle between the surface normal, L is the normalized light direction vector and dot(N, L) is the dot product of N and L.

Specular Lighting: The specular component represents the reflection of light off shiny surfaces. It depends on the angle between direction of perfect reflection and viewer's direction.

$$Is = Ks * Is * max(0, dot(R, V))^{n}$$

The overall lighting intensity is the sum of ambient, diffuse and specular intensities.

$$I = La + Ld + Ls$$

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

Ans: Translation: To represent translation using homogenous co-ordinates, a 3X3 transformation matrix is used.

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

where tx and ty are translation factors

The original co-ordinates (x, y) are extended to homogenous co-ordinates (x', y', 1) by appending the extra co-ordinate w = 1.

Rotation: Rotation in 2D can be representated using a 3X3 rotation matrix. The rotation matrix for rotating an angle θ counter-clockwise is:

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

Scaling: Scaling in 2D can be represented using 3X3 scaling matrix:

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

Where sx and sy are scaling factors.

4. Outline the differences between raster and random scan displays.

Ans: Raster scan displays:

- They are the most common type of displays used today.
- The screen is divided into a grid of pixels and the image is generated by scanning pixels sequentially from left to right, top to bottom.

Raster scan	Random scan
• Raster scan displays use a grid of pixels and scan them sequentially.	• Random scan displays draw lines and curves based on specified co-ordinates
• Raster scan displays assign color or intensity values to individual pixels	• Do not have individual pixels & draw lines and curves directly
• Raster scan displays can handle complex graphics and support color.	Random scan displays are simpler and typically monochromatic.
• Raster scan displays are commonly used in modern displays like LCDs and LEDs	• Used in older systems like vector displays.

5. Demonstrate OpenGL functions for displaying window management using GLUT.

Ans: • glutInit(argc, argv): Initializes the GLUT library and processes any command line arguments.
• glutInitDisplay(mode): Specifies the initial display mode for the window.
• glutInitWindowSize(width, height): Specifies the initial size of the window in pixels.
• glutCreateWindow(title): creates a window with the specified title.
• glutDisplayFunc(func): Sets the display callback function.
• glutReshapeFunc(func): sets the reshape callback function.
• glutKeyboardFunc(func): Sets the keyboard callback function.

6. Explain OpenGL Visibility Detection Functions.

Ans: Visibility detection functions are used to determine which parts of a scene are visible from a given viewport.

- glClear(mask): Clears the specified buffers, it is commonly used with GL_COLOR_BUFFER_BIT and GL_DEPTH_BUFFER_BIT.
- glEnable(GL_DEPTH_TEST): Enables Depth Testing, which compares the depth values of fragments to determine visibility.
- glDepthFunc(func): Depth comparison function used in depth Testing.

7. Write special cases with respect to Perspective projection transformation co-ordinates:

Ans: 1. Division by zero: If the w-co-ordinate of a transformed vertex becomes zero after the perspective division, it indicates that the vertex is at infinity and lies on perspective projection's focal plane.

2. Near clipping plane: When objects come too close to the camera, their transformed co-ordinates might have w-co-ordinates close to zero.

3. Far clipping plane: It used to discard objects too far away from the camera.

c 8. Explain Bezier curve equations along with properties.

Ans: Bezier curve is a type of curve used in computer graphics and modelling. It is defined by a set of points that influence its shape. The equation is as follows:

$$B(t) = \sum_{n=0} P_i \, B_i^n(t)$$

$$B_i^n(t) = {}^nC_i (1-t)^{n-t} t^i$$

where B(t) is the Bernstein's polynomial:

Properties:

1) Interpolation: A bezier curve passes through its first and last control points.
2) The degree of a Bezier curve is determined by the number of control points minus one.
3) The Parameter t determines the position on the curve. As t varies from 0 to 1, the curve is traced from starting to ending point.

9. Explain Normalization transformation for orthogonal Projection.

Normalization transformation is applied to convert orthographic projection of a 3D scene to a normalized device co-ordinate.

1. Define the clipping volume:

The first step to define a clipping volume which determines the visible portion of a 3D scene.

2. Translate and scale:

Once the clipping volume is defined, translation and scaling transformations are applied to map the clipping volume to a canonical volume.

The translation transformation shifts the clipping volume to align its center with the origin. This is achieved by calculating translation values as (-right, -left)/2, (-top-bottom)/2 and (-far-near)/2.

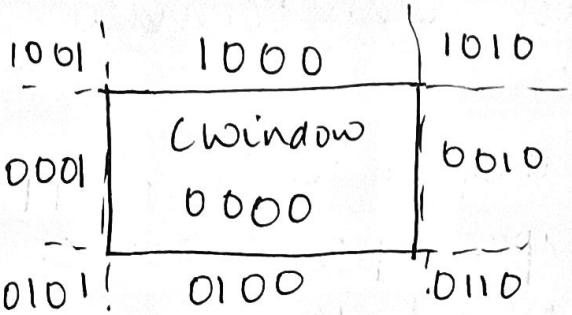
10. Explain cohen sutherland line-clipping Algorithm.

This Algorithm works on region code.

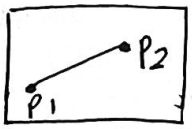
Region code 4bit code:

(ABRL) Above, Below, Right, Left

(TBRL) Top, Bottom, Right, Left.

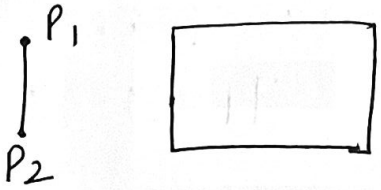


case 1: If both endpoint region code is zero, completely inside & visibu.



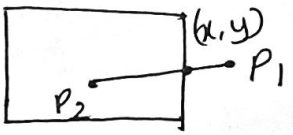
P_1 0000
P_2 0000
0000

case 2: If both endpoint Region code is nonzero, then apply Logical AND, and the result is non-zero



P_1 = 0001
P_2 0001
0001

case 3: If one of P_1 and P_2 is zero



Then find Intersection point

Intersection pt. (x, y).

$$m = \frac{y - y_1}{x - x_1} \; ; \; y - y_1 = m(x - x_1)$$

$$y = y_1 + m(x - x_1).$$

$$x - x_1 = \frac{y - y_1}{m}$$

$$x = \frac{x_1}{m}(y - y_1).$$