The **World-to-Screen** transformation is the process of mapping an object's coordinates in a 3D world space (world coordinates) to 2D screen space. This involves several transformations: Model transformation, View transformation, Projection transformation, and Viewport transformation. Here's a step-by-step breakdown of the procedure, including the matrices and an example for each transformation.1. **Model Transformation**
The model transformation maps the object's local coordinates to world coordinates. This transformation allows the object to be positioned, rotated, and scaled in the world.

- **Matrix:**

$$M_{\text{model}} = \begin{bmatrix} s_x & 0 & 0 & t_x \\ 0 & s_y & 0 & t_y \\ 0 & 0 & s_z & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where:

- $$s_x, s_y, s_z$$

  are the scaling factors along each axis.
- $$t_x, t_y, t_z$$

  are the translation components along each axis.
- **Example:**
  Suppose we want to scale an object by a factor of 2 and translate it by

  $$t_x = 3, t_y = 4, t_z = 5$$

  . The model matrix would be:

$$M_{\text{model}} = \begin{bmatrix} 2 & 0 & 0 & 3 \\ 0 & 2 & 0 & 4 \\ 0 & 0 & 2 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. **View Transformation (Camera Transformation)**
   The view transformation defines the camera's position and orientation in the world. This transforms the world coordinates to the camera (view) coordinates.

- **Matrix:**

$$M_{\text{view}} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} & -\text{eye}_x \\ r_{yx} & r_{yy} & r_{yz} & -\text{eye}_y \\ r_{zx} & r_{zy} & r_{zz} & -\text{eye}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where:

- $$r_{xx}, r_{xy}, \ldots, r_{zz}$$

  are the components of the rotation matrix that defines the camera's orientation.

- $$\text{eye}_x, \text{eye}_y, \text{eye}_z$$

  are the camera's position in world coordinates.

- **Example:**
  If the camera is at the origin

  $$(0, 0, 0)$$

  and looking along the z-axis, the view matrix might be an identity matrix:

  $$M_{\text{view}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. **Projection Transformation**
   The projection matrix transforms the 3D coordinates into a 2D plane. There are two common types of projection:

- **Orthographic projection** (no perspective distortion)
- **Perspective projection** (objects further away appear smaller)
- **Matrix for Perspective Projection:**

$$M_{\text{proj}} = \begin{bmatrix} \frac{1}{\tan(\theta/2)} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\theta/2)} & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & \frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Where:

- $$\theta$$

  is the field of view angle.

- $$f$$

  is the far clipping plane.
- $$n$$

  is the near clipping plane.
- **Example:**
  Suppose the field of view is

  $$90^\circ$$

  and the near and far planes are at

  $$n = 1$$

  and

  $$f = 100$$

  . The perspective projection matrix could be:

  $$M_{\text{proj}} = \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & \frac{101}{99} & \frac{2 \times 100 \times 1}{99} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

4. **Viewport Transformation**
   The viewport transformation maps the normalized device coordinates (which range from -1 to 1 after the projection transformation) to screen coordinates. The matrix for viewport transformation is used to scale and translate the coordinates to the actual screen resolution.

- **Matrix:**

  $$M_{\text{viewport}} = \begin{bmatrix} \frac{w}{2} & 0 & 0 & \frac{w}{2} \\ 0 & \frac{h}{2} & 0 & \frac{h}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where:

- $$w$$

  is the width of the screen.

- $$h$$

  is the height of the screen.
- **Example:**
  If the screen width is

  $$800$$

  pixels and the height is

  $$600$$

  pixels, the viewport matrix would be:

  $$M_{\text{viewport}} = \begin{bmatrix} 400 & 0 & 0 & 400 \\ 0 & 300 & 0 & 300 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Final Transformation

To compute the final coordinates, you need to multiply all these matrices in the following order:

$$P_{\text{final}} = M_{\text{viewport}} \times M_{\text{proj}} \times M_{\text{view}} \times M_{\text{model}} \times P_{\text{object}}$$

Where

$$P_{\text{object}}$$

is the object's coordinates in its local space.

Example Walkthrough:

Let’s say we have an object with local coordinates

$$P_{\text{object}} = (1, 2, 3, 1)$$

.

1. **Model Transformation:**

- Apply the model matrix to the object:

$$M_{\text{model}} \times P_{\text{object}} = \begin{bmatrix} 2 & 0 & 0 & 3 \\ 0 & 2 & 0 & 4 \\ 0 & 0 & 2 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 2(1)+0(2)+0(3)+3 \\ 0(1)+2(2)+0(3)+4 \\ 0(1)+0(2)+2(3)+5 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \\ 11 \\ 1 \end{bmatrix}$$

2. **View Transformation:**

- Apply the identity view matrix:

$$M_{\text{view}} \times P_{\text{object}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 5 \\ 8 \\ 11 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \\ 11 \\ 1 \end{bmatrix}$$

3. **Projection Transformation:**

- Apply the perspective projection matrix (assuming values as before):

$$M_{\text{proj}} \times P_{\text{object}} = \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & \frac{101}{99} & \frac{2\times100\times1}{99} \\ 0 & 0 & -1 & 0 \end{bmatrix} \times \begin{bmatrix} 5 \\ 8 \\ 11 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \\ \frac{101}{99} \times 11 + \frac{2\times100\times1}{99} \\ -11 \end{bmatrix}$$

4. **Viewport Transformation:**

- Finally, apply the viewport matrix (assuming the screen size as before):

$$M_{\text{viewport}} \times P_{\text{object}} = \begin{bmatrix} 400 & 0 & 0 & 400 \\ 0 & 300 & 0 & 300 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 5 \\ 8 \\ \text{calculated z value} \\ -11 \end{bmatrix}$$

This gives the final 2D screen coordinates for the object.