1. **Projection Transformation** (Camera Coordinates to Viewplane Coordinates)

   Let $(x_c,\, y_c,\, z_c)$ be a point in camera coordinates and let $(x_p,\, y_p,\, z_p)$ be its perspective projection onto the viewplane. Then

$$x_p = -x_c/z_c,$$

$$y_p = -y_c/z_c,$$

$$z_p = -1.$$

2. **Viewplane to Pixelplane Transformation**

   Let $(x_p,\, y_p,\, -1)$ be a point in the viewplane and let $(x_{vp},\, y_{vp})$ be its transformation to the renderer's pixel-plane. Then

$$x_{vp} = 0.5 + (w/2.001)(x_p + 1),$$

$$y_{vp} = 0.5 + (h/2.001)(y_p + 1).$$

   A point $(x_p,\, y_p,\, -1)$ from the viewplane's view rectangle will transform to a point in $(x_{vp},\, y_{vp})$ in the renderer's *logical viewport* with coordinates that satisfy

$$0.5 \le x_{vp} < w + 0.5 \qquad \text{and} \qquad 0.5 \le y_{vp} < h + 0.5$$

   where w and h are the width and height of the framebuffer's viewport. Points in the pixel-plane with integer coordinates are called *logical pixels*.

3. **Pixelplane to Pixel Transformation**

   Let $(x_{vp},\, y_{vp})$ be a point in the renderer's logical viewport (in the pixel-plane). Then

$$(\texttt{Math.round}(x_{vp}),\, \texttt{Math.round}(y_{vp}))$$

   is the logical pixel nearest to $(x_{vp},\, y_{vp})$. Let $(x,\, y)$ be its equivalent (physical) pixel in the framebuffer's viewport. Then

$$x = (\texttt{int})\texttt{Math.round}(x_{vp}) - 1,$$

$$y = h - (\texttt{int})\texttt{Math.round}(y_{vp}).$$

   Pixels $(x,\, y)$ in a framebuffer's viewport have integer coordinates that satisfy

$$0 \le x \le w - 1 \qquad \text{and} \qquad 0 \le y \le h - 1$$

   with the pixel $(0,\, 0)$ being the upper left-hand corner of the viewport.