Cory Zimmerman

(Q1)
TODO

(Q2: 11.3)
TODO

(Q3: 11.2)
Let the projection matrix be defined as

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

With eye coordinates x_e, y_e, z_e, this yields clip coordinates

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_e \\ y_e \\ z_3 \\ 1 \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \\ 1 \\ -z_3 \end{bmatrix}$$

This produces device coordinates

$$\begin{bmatrix} \frac{x_e}{-z_e} \\ \frac{y_e}{-z_e} \\ \frac{1}{-z_e} \\ 1 \end{bmatrix}$$

Using the projection matrix PQ instead generates

$$PQ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -3 & 0 \end{bmatrix}, PQ \cdot \begin{bmatrix} x_e \\ y_e \\ z_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 3x_e \\ 3y_e \\ 1 \\ -3z_e \end{bmatrix}$$

This produces device coordinates

$$\begin{bmatrix} \frac{3x_e}{-3z_e} \\ \frac{3y_e}{-3z_e} \\ \frac{1}{-3z_e} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{x_e}{-z_e} \\ \frac{y_e}{-z_e} \\ \frac{1}{-3z_e} \\ 1 \end{bmatrix}$$

Mathematically, it appears that the coordinates projected by PQ have identical x and y values but are squished closer towards the origin in the z direction compared to coordinates projected by just P. I believe this has the effect of extending the z-length of the visual frustrum, fitting more total coordinates into the view window. In a rendering environment, I think this would make us see further.

I made a test example in `mtx_proj.py`. With eye coordinates $(0, 0, -1, 1), (0.5, 0, -0.5, 1), (0, 0.5, 0.5, 1)$, the device coordinates projected by just P are $(0, 0, 1, 1), (1, 0, 2, 1), (0, -1, -2, 1)$, while the device coordinates projected by PQ are $(0, 0, 0.33, 1), (1, 0, 0.67, 1), (0, -1, -0.67, 1)$.