Roseanne Feng
Belinda Zeng

Pset 9 – Written Questions

1. B. Intuitively, the camera would be a bad camera if it didn't preserve the ratio of distance between points. Additionally, if we plug in actual points, say [1,1,1,1] and [1,1,-1,1] to the normal matrix and the projection matrix affected by tan(θ), we see that the ratio of the first x to the second x remains the same.
2. No change because even though the x, y, z coordinates are all scaled by a factor of 3, v_n is also increased by a factor of 3 so the changes cancel out when we are converting to normalized device coordinates. The x and y screen coordinates and clipping is not affected; the image looks the same.
   The normal projection matrix looks like:
   $$\begin{bmatrix} x_n\,w_n \\ y_n\,w_n \\ z_n\,w_n \\ w_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix}$$
   Such that $x_n = x_e / z_e$ and $y_n = y_e / z_e$ and $w_n = -z_e$

   When we change the projection matrix to equal PS where S =
   $$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$
   Then the new equation becomes:
   $$\begin{bmatrix} x_n\,w_n \\ y_n\,w_n \\ z_n\,w_n \\ w_n \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & -3 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix}$$
   Such that $x_n = x_e / z_e$ and $y_n = y_e / z_e$, just like the original matrix, since w_n has become -3z_e .

3. The x and y coordinates are scaled by a factor of 3, as well as v_n so when the image is drawn we do not perceive a change in size of objects drawn. However, since z is not scaled, when we calculate z_n less objects are clipped and we see more objects drawn because their z_n is more likely to satisfy $-1 < z_n < 1$.
   Here, we use PQ in place of P:
   $$\begin{bmatrix} x_n\,w_n \\ y_n\,w_n \\ z_n\,w_n \\ w_n \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -3 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix}$$

such that $x_n = x_e / z_e$ and $y_n = y_e / z_e$, just like in matrix, since w_n has become -3z_e. However, $z_n = -1/3z_e$ which is why we see the change in clipping coordinates.

4. The x, y, and z coordinates are all scaled by a factor of 3, but v_n does not also scale, so we see the image changes by basically zooming into the picture so that objects appear 3x bigger and we do not see the same amount of objects because the z increases so objects are less likely to satisfy $-1 < z_n < 1$ and get clipped as a result. This affects the graphics pipeline when converting from clipping coordinates to normalized device coordinates as we end up with less overall information from the original image.
   Here, we use QP in place of P:
   $$\begin{bmatrix} x_n\, w_n \\ y_n\, w_n \\ z_n\, w_n \\ w_n \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix}$$
   such that $x_n = 3x_e / z_e$ and $y_n = 3y_e / z_e$, since w_n is still -z_e but x_n and y_n have clearly increased by a factor of 3. Additionally, $z_n = -3/z_e$, which is why we see the change in clipping coordinates.
5. The x and y coordinates are scaled by a factor of 3, z and v_n do not change. This makes the image appear bigger by a factor of 3, but nothing changes with respect to clipping, so we still see the same amount of objects, unlike in 11.4, which clips more objects. Because of the way the window coordinates are defined, it scales with respect to the [-0.5, -0.5] point instead of [0, 0] so the half-pixel offset causes the lower left corner to stay at [-0.5, -0.5] and the upper right corner is at [3W – 0.5, 3H – 0.5] instead of [3(W – 0.5), 3(H – 0.5)].
6. [0.45 * 512, 0.63 * 512] = [230.4, 322.56]
   Subtract by 0.5 to account for half-pixel offset → [229.9, 322.06] so the closest pixel values are [230, 322].
   To express this in canonical square coordinates, add 0.5 → [230.5, 322.5]
   [230.5 / 512, 322.5 / 512] = [0.4502, 0.6299]
7. So we can define $\begin{bmatrix} v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}$
   We can then divide these values by w_n : $\begin{bmatrix} v/w_n \\ 1/w_n \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} \frac{1}{w_n}$
   This gives us v/w_n and $1/w_n$ which are affine wrt screen space