CSCI 480 / 580 – February 10, 2020 – Viewing Transformations

1. The viewport matrix takes points in *normalized device coordinates* and maps their x, y coordinates into image (pixel) space. Write a 4x4 transformation matrix that performs the following steps:
   2. Scale – x and y coordinates begin in the range [-1, 1] – a range of size 2
      a. x should be scaled to fit in a range of size W-1
      b. y should be scaled to fit in the range [H-1, 0]
      c. z should be left alone; no scale is applied
   3. Translate – the origin (0,0) should get mapped to x = (w-1)/2, y = (h-1)/2

$$\mathbf{M_{vp}} = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

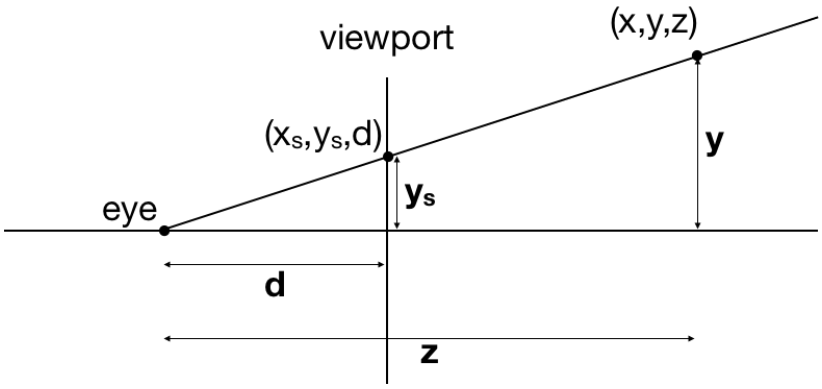
CSCI 480 / 580 – February 10, 2020 – Viewing Transformations

1. The viewport matrix takes points in *normalized device coordinates* and maps their x, y coordinates into image (pixel) space. Write a 4x4 transformation matrix that performs the following steps:
   1. Scale – x and y coordinates begin in the range [-1, 1] – a range of size 2
      d. x should be scaled to fit in a range of size W-1
      e. y should be scaled to fit in the range [H-1, 0]
      f. z should be left alone; no scale is applied
   2. Translate – the origin (0,0) should get mapped to x = (w-1)/2, y = (h-1)/2

$$\mathbf{M_{vp}} = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

2. Consider the side-view of a perspective camera shown below: the image looks at the yz plane *from* the +x direction. The scene point at (*x*, *y*, *z*) appears in the viewport at (x_s, y_s, -*d*). Based on this view, calculate the value of y_s, the y coordinate of the viewing ray's intersection with the viewport, in terms of the pixel's 3D coordinates (*x*, *y*, *z*) and the camera's focal length *d*.

   Hint: no trig necessary! Try similar triangles.



2. Consider the side-view of a perspective camera shown below: the image looks at the yz plane *from* the +x direction. The scene point at (*x*, *y*, *z*) appears in the viewport at (x_s, y_s, -*d*). Based on this view, calculate the value of y_s, the y coordinate of the viewing ray's intersection with the viewport, in terms of the pixel's 3D coordinates (*x*, *y*, *z*) and the camera's focal length *d*.

   Hint: no trig necessary! Try similar triangles.