More on Affine Transformations

1. Suppose we have 2D frame with an origin at (2,2) and basis vectors $u = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$, $v = \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$. Suppose a point $p=(1,1)$ in $u,v$ coordinates. What is $p$ in $x,y$ coordinates?

$$p = 1(u) + 1(v) + (2,2)$$
$$= \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) + \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) + (2,2)$$
$$= \left(0, \frac{3}{\sqrt{2}}\right) + (2,2)$$
$$= (2, 2+\frac{3}{\sqrt{2}})$$

2. Construct a matrix that can convert a point in $u,v$ coordinates to $x,y$ coordinates for the basis described above.

$$\begin{bmatrix}
\frac{1}{\sqrt{2}} & 0 & \frac{3}{\sqrt{2}} \\
0 & 1 & 0 \\
\frac{1}{\sqrt{2}} & 0 & 2
\end{bmatrix} = \begin{bmatrix}
\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 2 \\
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 2 \\
0 & 0 & 1
\end{bmatrix}$$

3. What matrix would perform the window-to-viewport transformation specified by the function call `gl.viewport(0,0,960,640)`? For this question, just ignore the z-coordinate and imagine the NDC coordinates to be transformed are in the form $(x, y, 1)$. Express your answer as a 3x3 matrix and express rational numbers as fractions.

In WebGL, the window to viewport transformation is performed by the WebGL library during primitive assembly. You control it using the call `gl.viewport(x, y, w, h)`. It specifies the viewport by giving the lower-left coordinate of the viewport with $(x, y)$ and the width and height of the viewport with $w$ and $h$. 

$$\begin{bmatrix}
\frac{95}{2} & 0 & 0 \\
0 & \frac{95}{2} & 0 \\
0 & 0 & 1
\end{bmatrix} = \begin{bmatrix}
\frac{95}{2} & 0 & \frac{95}{2} \\
0 & \frac{95}{2} & 0 \\
0 & 0 & 1
\end{bmatrix}$$
4. Suppose we have the following set of transformations that map a point in view (or camera) space to world coordinates. What transformations will map world coordinates to camera space?

\[
\begin{pmatrix}
1 & 0 & 0 & e_x \\
0 & 1 & 0 & e_y \\
0 & 0 & 1 & e_z \\
0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
u_x \\
v_y \\
v_z \\
w
\end{pmatrix} = \begin{pmatrix}
u_x \\
v_y \\
v_z \\
w
\end{pmatrix}
\]

\[
\begin{pmatrix}
u_x \\
v_y \\
v_z \\
w
\end{pmatrix} = \begin{pmatrix}
u_x \\
v_y \\
v_z \\
w
\end{pmatrix}
\]

Bonus question:
What is the inverse of matrix that performs a scale transformation?

\[
\begin{pmatrix}
s_x & 0 & 0 & 0 \\
0 & s_y & 0 & 0 \\
0 & 0 & s_z & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}^{-1}
\]

\[
\begin{pmatrix}
\frac{1}{s_x} & 0 & 0 & 0 \\
0 & \frac{1}{s_y} & 0 & 0 \\
0 & 0 & \frac{1}{s_z} & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]