HW 2

(1) A general perspective matrix in 2D has the form

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Due to invariance under multiplication, we need <u>4 points</u> to determine the matrix:

e.g.:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} 1 \\ -3 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} -w_1 \\ -w_1 \\ w_1 \end{bmatrix}$$

Note that, since the ~~Matrix~~ matrix is not normalized, we cannot expect the w coordinate to be normalized to 1 after transformation. Thus, we have to assume a general w component w_1 ~~on~~ the right.

So, specifying 1 point has given us 3 equations, but also a new unknown w_1

$$a - 3b + c = -w_1$$

$$d - 3e + f = -w_1$$

$$g - 3h + i = w_1$$

If you specify 3 more points in general position (ie no 3 points on one line) you get enough equations to determine the matrix entries $a - i$ <u>up to a constant multiplier</u>. You can set that multiplier to any <u>non zero</u> value.

The solving of the system is exactly like for the affine case (HW 1)

HW 2

(2) There are many solutions. One is this:

```
pushMatrix()
  scale (3,2)
  drawSquare()          // first square
popMatrix()

translate (3,2)         // frame in top right corner of box 1
rotate (α)              // would draw box as pictured here ->
pushMatrix()
  scale (5,-2)          // note negative scaling -> flips along this axis
  drawSquare()
popMatrix()

translate (5,0)
scale (4,-2)            // same idea...
drawSquare()
```

Alternatively: use combination of positive ~~translation~~ scaling and translation!