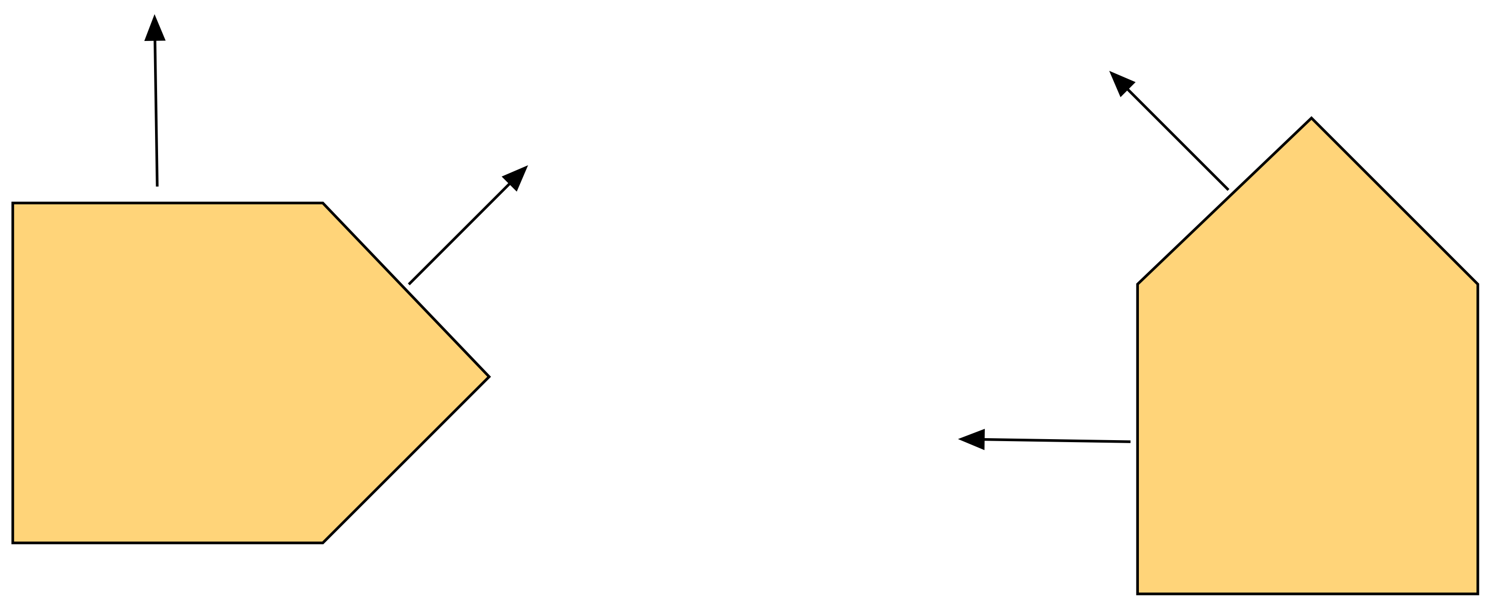
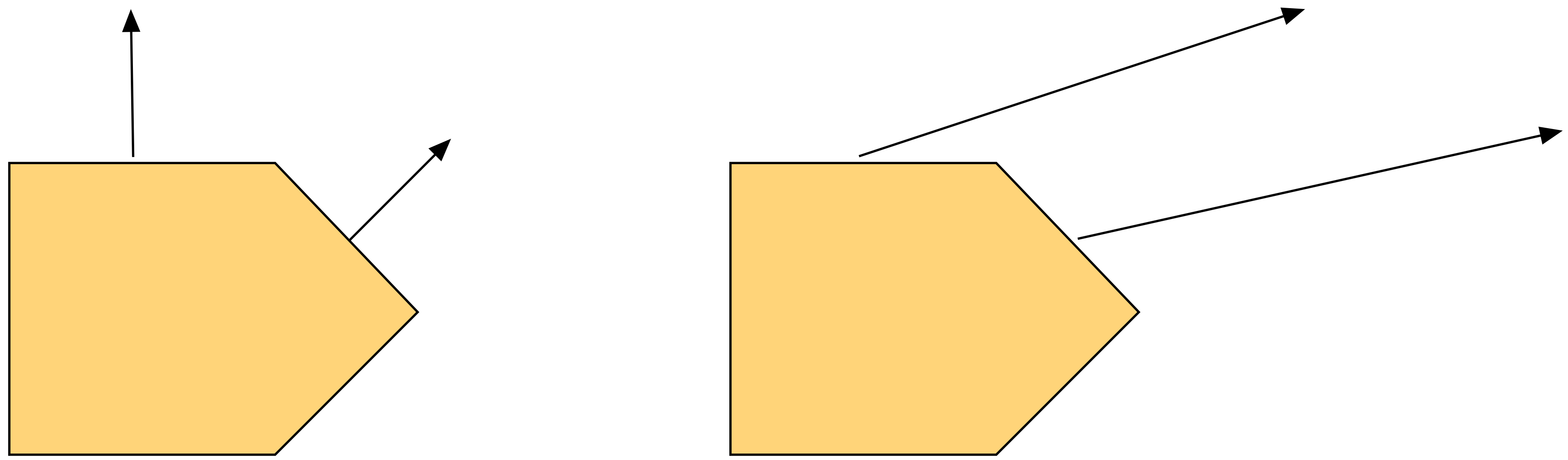
Now, lets return to that normal matrix...

The Normal matrix

When placing a model in the world, normals must be rotated...



but they must not be translated...



so we just cast to mat3, right?

$$\begin{bmatrix} r & r & r & t_x \\ r & r & r & t_y \\ r & r & r & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} r & r & r \\ r & r & r \\ r & r & r \end{bmatrix}$$

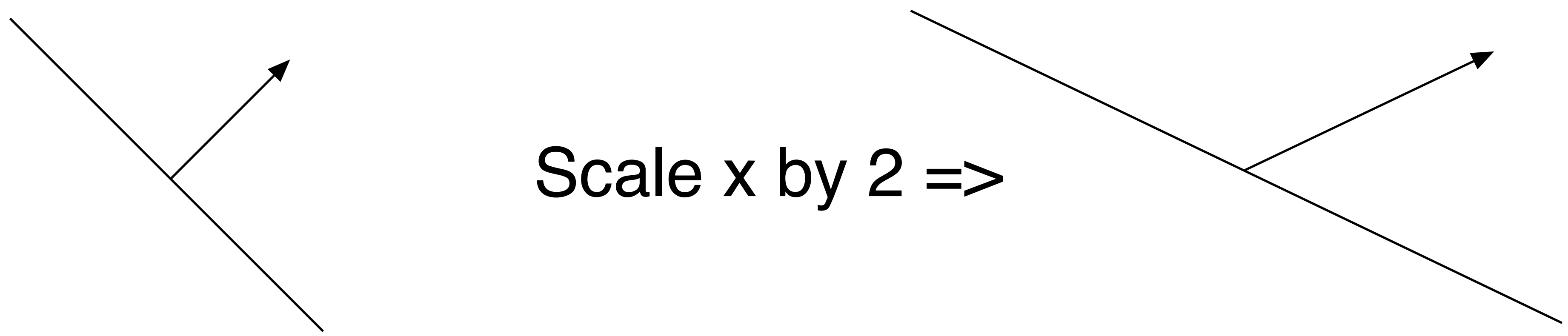
or we zero the translation part:

$$\begin{bmatrix} r & r & r & t_x \\ r & r & r & t_y \\ r & r & r & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} r & r & r & 0 \\ r & r & r & 0 \\ r & r & r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It works most of the time... but...

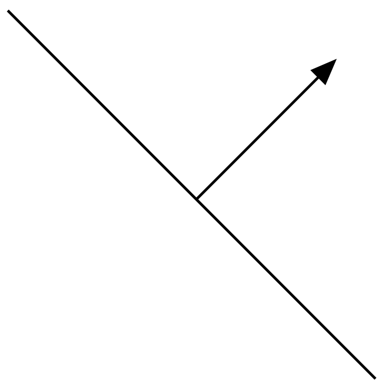
But wait!

For *non-uniform scaling*, this does not work!

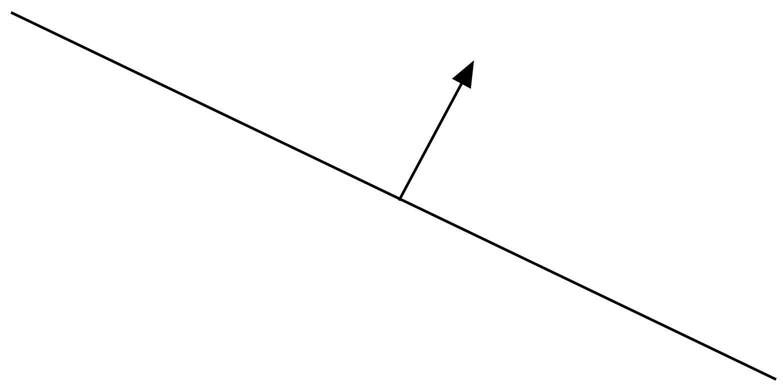


The normal vector is no longer perpendicular to surface!

But what if we do the *opposite*...



Scale geometry by 2 along x
Scale normal by 1/2 along x
=>



Suddenly things look better...

but what happens if we mix in rotations?

Normal matrix, full solution

Invert scaling, keep rotation

1) Invert to reverse both
2) Transpose to reverse rotation

=> Use *inverse transpose* of rotation part

$$N = (M^{-1})^T$$

Normal matrix, full solution

Invert scaling, keep rotation

1) Invert to reverse both
2) Transpose to reverse rotation

=> Use *inverse transpose* of rotation part

$$N = (M^{-1})^T$$

Please don't miss this!