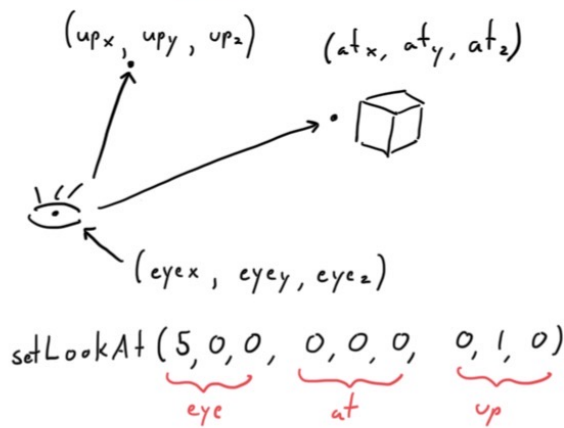


View Matrix



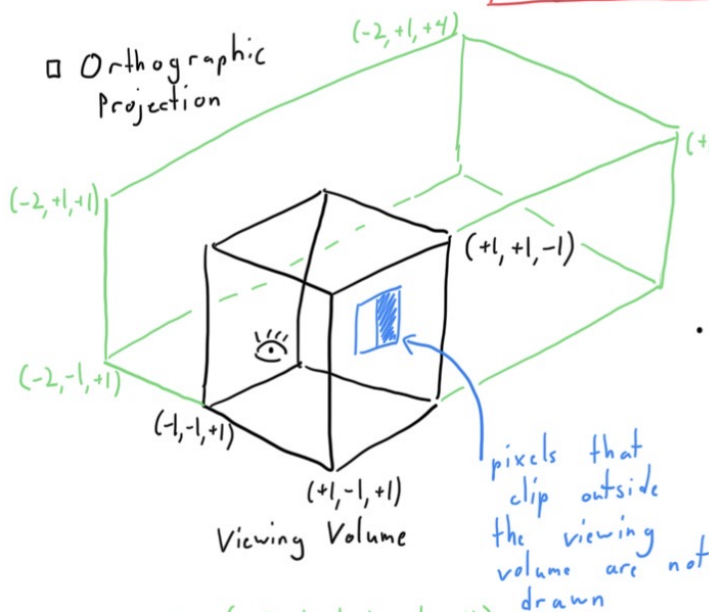
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -5 & 0 & 0 & 1 \end{bmatrix}$$

to simulate moving the "camera", we move everything else inversely!

• there is no actual camera in computer graphics !!

Projection Matrix

□ Orthographic Projection



ex: `setOrtho(-2, 1, -1, 1, -1, 4)`

• WebGL has a fixed $1 \times 1 \times 1$ viewing volume

• Orthogonal projection - "ortho" means "straight"

• We can simulate a scaled viewing volume by inversely scaling & translating everything else.

r = right t = top f = far
 l = left b = bottom n = near

$$\begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{-2}{f-n} & 0 \\ \frac{-r+l}{r-l} & \frac{t+b}{t-b} & \frac{f+n}{f-n} & 1 \end{bmatrix}$$

scale translate

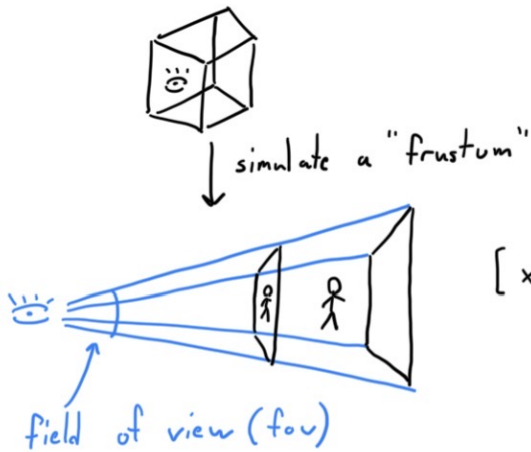
a-Position

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix}$$

"homogeneous coordinate"

created by `setOrtho(l, r, t, b, n, f)`

□ Perspective Projection - gives illusion of depth



Perspective Matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1/k \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} x & y & z & 1-z/k \end{bmatrix}$$

k being some magnitude of "fov"

$$\begin{bmatrix} \frac{x}{1-z/k} & \frac{y}{1-z/k} & \frac{z}{1-z/k} \end{bmatrix}$$

• weight (w) is used in "perspective divide"
This divide is fixed & always occurs between shaders.



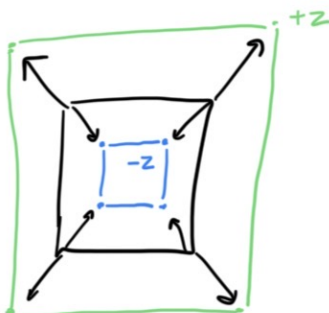
$gl_position = \text{vec3}(gl_position.xyz / gl_position.w)$
so, w divides xyz

where the pixel appears in the display

used in the z-buffer to discard pixels drawn over nearer pixels

$gl_enable(GL_DEPTH_TEST)$

• intuition - x, y values with positive z are "closer" and should be "scaled up"



• negative z → scale down
• Thus, translation is magnified for close vertices & minimized for far vertices