Orthographic Derivation
• scale, translate, reflect for new coord sys

\[ \begin{align*}
    x' &= ax + b \\
    y' &= ay + b \\
    z' &= cz + d
\end{align*} \]

\[ \begin{pmatrix}
    x' \\
    y' \\
    z'
\end{pmatrix} =
\begin{pmatrix}
    a & b & c & d \\
    0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
    x \\
    y \\
    z \\
    1
\end{pmatrix} \]

same idea for right/left, far/near
Projective Rendering Pipeline

- OCS - object/model coordinate system
- WCS - world coordinate system
- VCS - viewing(camera/eye) coordinate system
- NDCS - normalized device coordinate system
- DCS - device/display/screen coordinate system

**Perspective Example**

Tracks in VCS:
- left x=-1, y=-1
- right x=1, y=1
- near x=1, far = 4

View volume:
- left x=-1, y=-1
- bot = -1, top = 1
- near x=1, far = 4

**NDC Transformation**

- viewport/window location wrt actual display not available within GL
- Usually don’t care
- Use relative information when handling mouse events, not absolute coordinates
- Could get actual display height/width, window offsets from OS
- Loose use of terms: device, display, window, screen...

**N2D Transformation**

- viewpoint/window location wrt actual display not available within GL
- Usually don’t care
- Use relative information when handling mouse events, not absolute coordinates
- Could get actual display height/width, window offsets from OS
- Loose use of terms: device, display, window, screen...

**N2D Transformation**

- General formulation
  - Reflect in y for upper vs. lower left origin
  - Scale by width, height, depth
  - Translate by width/2, height/2, depth/2
  - FCG includes additional translation for pixel centers at (.5,.5) instead of (0,0)

**Coordinate Systems**

- OCS - object coordinate system
- WCS - world coordinate system
- VCS - viewing coordinate system
- NDCS - normalized device coordinate system
- DCS - device coordinate system

**Origin Location**

- Yet more (possibly confusing) conventions
  - GL origin: lower left
  - Most window systems origin: upper left
  - Then must reflect in y
  - When interpreting mouse position, have to flip your y coordinates

**General Formulation**

- Map from NDC to pixel coordinates on display
  - NDC range is x = -1...1, y = -1...1
  - Typical display range: x = 0...500, y = 0...300
  - Maximum is size of actual screen
  - Z range max and default is 0, use later for visibility
  - GL viewPort(0,0,n,f); Gil: if depth = 1 by default

**Display z range is 0 to 1.**

**Display**

- GL depthRange(n,f) can constrain further, but depth = 1 is both max and default

**View Volume**

- Left = -1, right = 1
- Bottom = -1, top = 1
- Near = 1, far = 4

- General formulation
  - Reflect in y for upper vs. lower left origin
  - Scale by width, height, depth
  - Translate by width/2, height/2, depth/2
  - FCG includes additional translation for pixel centers at (.5,.5) instead of (0,0)