(sorry for my writing quality)

4 – Arbitrary Rotations, Stacks and Graphs

If my laptop disconnects, DON'T leave. I will switch to sharing directly from the ipad.
(should be fine, just tested for 1 hour before class :))

Readings

- Review **Math** (chapter 2) as needed
- **Quaternions**: 16.2 (ways of representing arbitrary rotations)
  - "get big picture"

A1(a): Transformation Matrices
A1(b): Projection and Line Drawing

A1(a): implement common transformations and matrix multiplication

A1(b): implement common projections and transform lines by projection and matrix transformations

A1(a) released this evening, due next Saturday midnight

Arbitrary Rotations

basic R_x, R_y, R_z

could break down in R_x, R_y, R_z

→ "gimble lock"

unit Vector = $|V|$ length 1

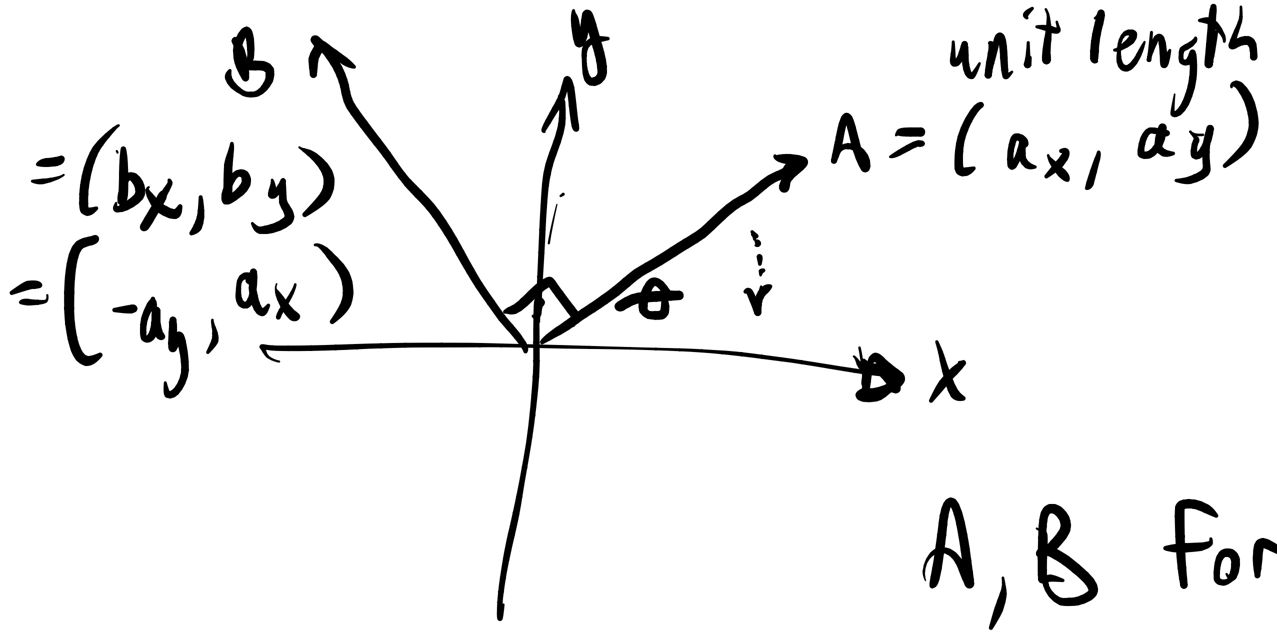
$$\hat{V} = \frac{V}{|V|} = \left(\frac{x}{L}, \frac{y}{L}, \frac{z}{L}\right) \qquad L = \sqrt{x^2+y^2+z^2}$$

$$\hat{V} \cdot \hat{V} = 1$$

If A & B are orthogonal (perpendicular)

$$A \cdot B = 0$$

$$V_1 \times V_2 \Rightarrow V_3$$



want to rotate θ to have A lie on x-axis

A, B form orthonormal basis

$$R = \begin{bmatrix} a_x & a_y & 0 \\ b_x & b_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_x & a_y & 0 \\ -a_y & a_x & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

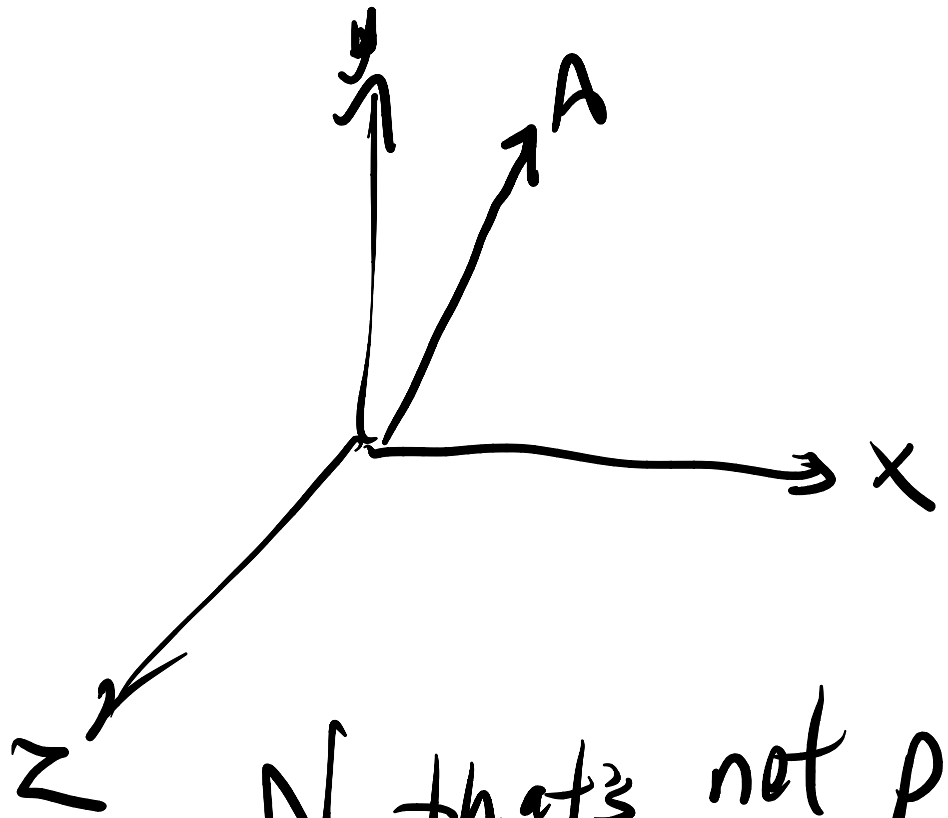
$$R \cdot A = \begin{bmatrix} a_x \\ a_y \\ 1 \end{bmatrix}$$

$$R \begin{bmatrix} a_x & a_y & 0 \\ -a_y & a_x & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_x^2 + a_y^2 + 0 \\ -a_x a_y + a_x a_y + 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$R^T \cdot [1, 0, 1] \quad \begin{bmatrix} a_x \\ a_y \\ 1 \end{bmatrix}$$

$$R \cdot A \rightarrow [1, 0, 1]$$
$$R \cdot B \rightarrow [0, 1, 1]$$

$$R^{-1} = R^T = \begin{bmatrix} a_x & -a_y & 0 \\ a_y & a_x & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$A = (a_x, a_y, a_z)$

Goal: rotate(theta, a_x, a_y, a_z)

$$R = R_3 R_2 R_1 \qquad R_1 = \begin{bmatrix} a_x & a_y & a_z & 0 \\ b_x & b_y & b_z & 0 \\ c_x & c_y & c_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

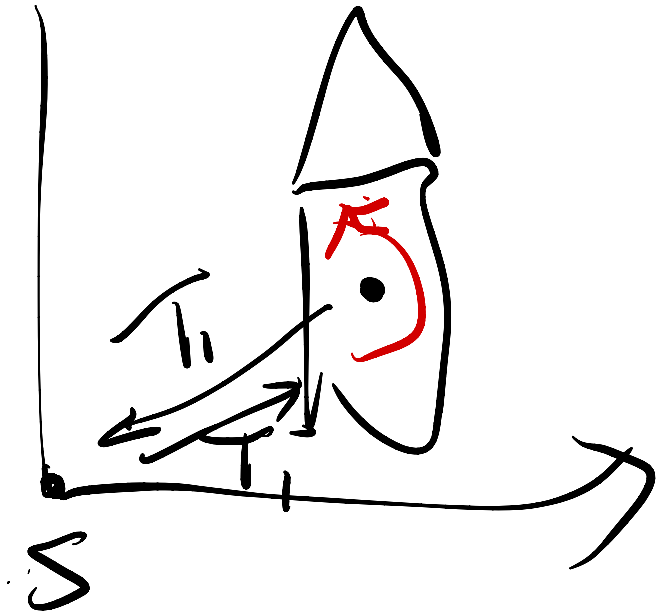
$$= R_1^{-1} R_2 R_1$$

N that's not parallel to A

$$B = \frac{A \times N}{|A \times N|} \qquad B \perp A \text{, unit length}$$

$$C = A \times B$$

A, B, C form an orthonormal basis in 3D



$$S_{orig} = T_1^{-1} S T_1$$

form N

if (dx is very small)

$N = (1, 0, 0)$

else

$N = (0, 1, 0)$

$R_2 = $ rotateX(theta)

$R_3 = R_1^{-1}$ ← matrix inversion

For Rotation Matrices $R_1^{-1} = R_1^T$

$R = R_3 R_2 R_1$

rotate(theta, A)

Rotation Interpolation: Quaternions

$Q = (s, x, y, z)$

(problems solved: interpolation of matrices, gimble lock)

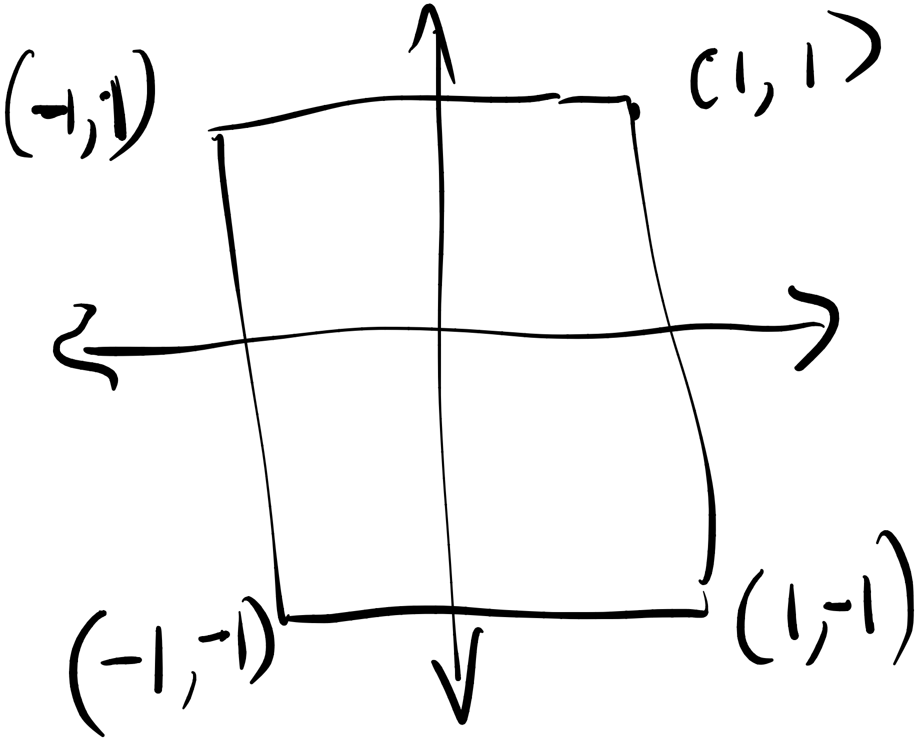
interpolation problem as well

matrices ⇒ difficult because we can't interpolate

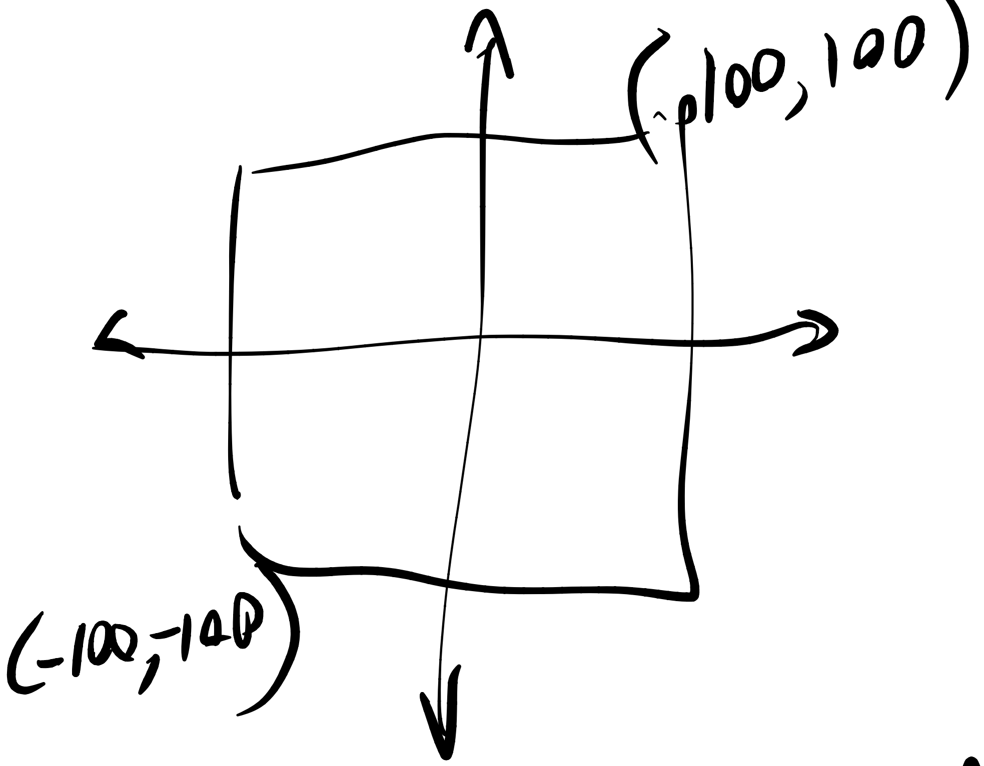
$$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & -1 \\ 0 & 1 \end{bmatrix} \quad \ldots \quad \overset{\text{at } 0.5}{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}}$$

if I do pointwise interp.

Direct Rendering with Matrix Stacks (OpenGL)

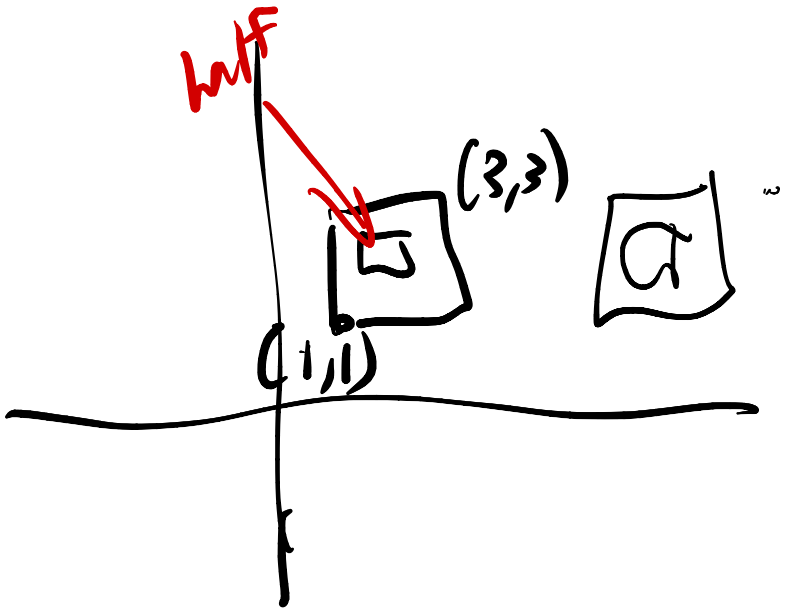



```
function Square() {
  begin shape();
  startpoint(1, 1)
  line to (-1, 1)
  line to (-1,-1)
  line to (1,-1)
  End shape()
}
```



init() // C=I
scale(100,100) // C=I•S
Square()

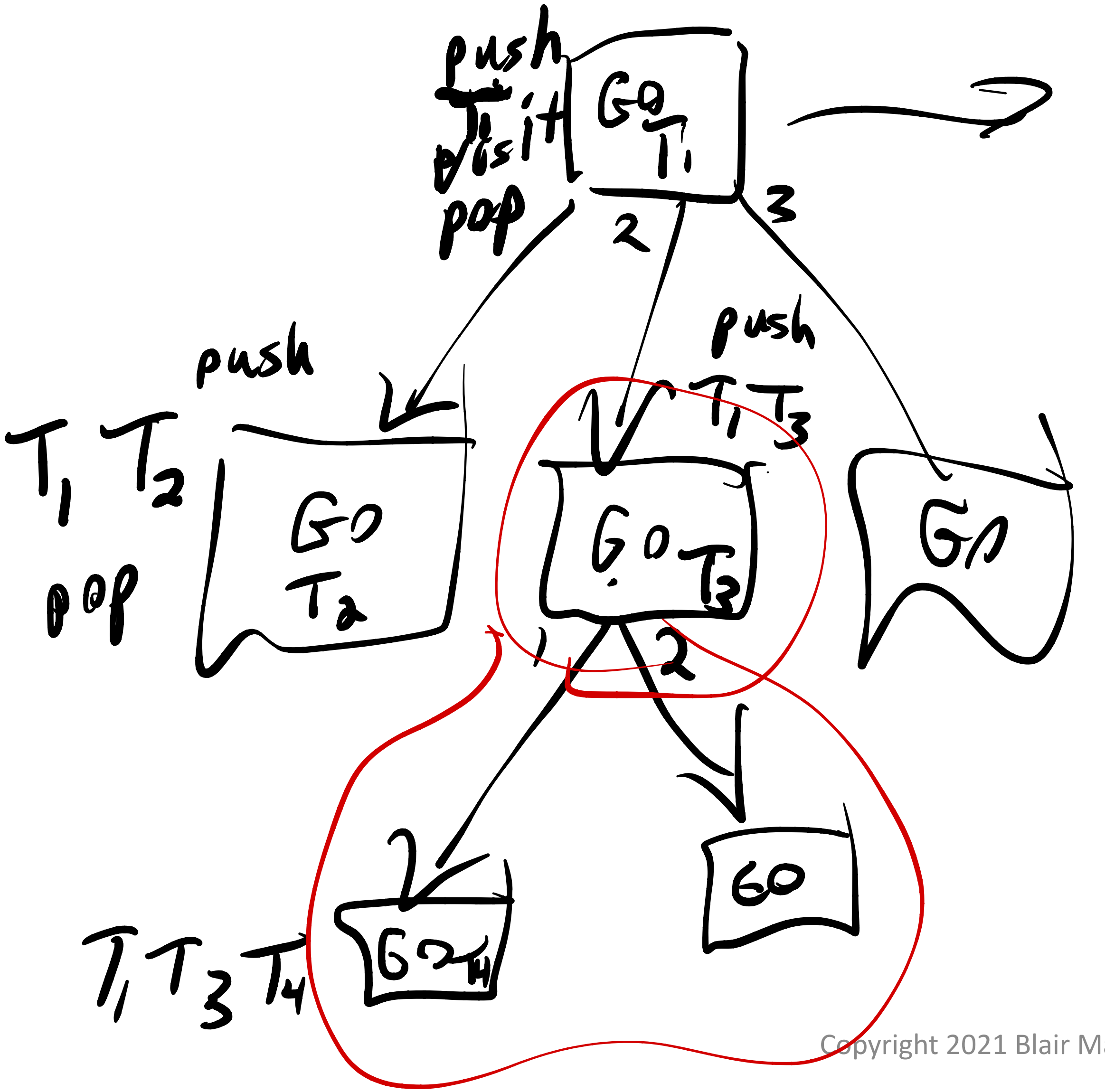
C = current transform
All vertices (x,y) are transformed by C   P' = C•P



doubleSquare
push() → copy of C & push onto a stack
scale(0.5, 0.5)
square()
pop()
square()

push
translate(2, 2)
double square
pop

Scene Graphs (Unity, Three.js, Babylon, etc)



GO's, Object3D, ...

base object
- transformation
- name, other properties, ...

children

Two Kinds of Graphics Libraries

→ Immediate Mode

→ Canvas, WebGL, OpenGL

"the comands I issue are executed immediately"

→ Retained Mode

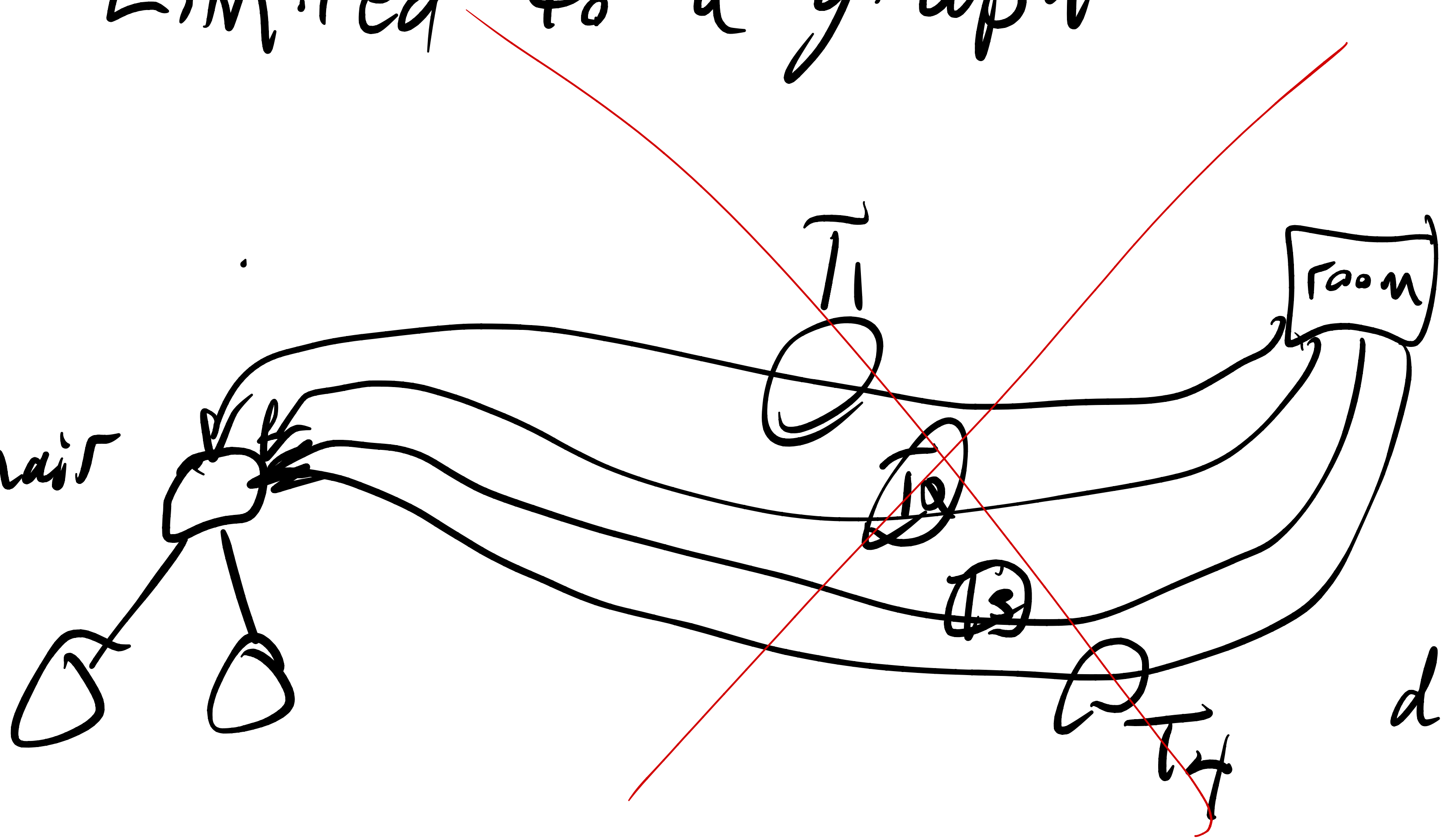
→ Three.js, Unity, ...

Build a graph

The system renders

With a S.G.

Limited to a "graph"



Want 4 chairs

different ~~x~~form to move the chairs