Finger Exercises 9

Frames, Changing Coordinates, Scene Graphs

≈ 65 minutes, plus a 10-minute bonus
CS 453 — Computer Graphics

How to do it: On **paper** first, no computer, squared paper beside you. Sketch the frame *before* you compute. Conventions are the lecture's: x right, y up, positive angles counterclockwise, points get a 1 in the third slot and directions a 0, products read right to left. If an item takes longer than 5 minutes, skip it, finish the rest, come back.

Short on time? Do Parts C, D, E, G, J and K first (about 30 minutes). They are the heart of the lecture, and of A3.

Part A • Do You Remember?

RECALL ≈ 6 min

A1. A coordinate frame is an ____________ plus a ____________. The basis must be *orthonormal*: each arrow has ______________, and $\mathbf{u} \cdot \mathbf{v} =$ _____. It is *right-handed* when $\mathbf{v}$ is __________________________ from $\mathbf{u}$.

A2. The book writes the *thing measured* in the subscript and the *axis* as the main letter. So x_p is ________________, u_p is ____________________, and x_u is ____________________.

A3. Frame to world: $\mathbf{p} =$ _______________________. World to frame: first $\mathbf{b} =$ __________, then $u_p =$ _________ and $v_p =$ _________.

A4. The frame-to-canonical matrix has columns ______, ______, ______ and bottom row _________. It takes ___________ numbers in and gives ___________ numbers out.

A5. $\mathbf{M}_{TB}$ takes _________ numbers in and gives _________ numbers out. $\mathbf{M}_{WT}\mathbf{M}_{TB} =$ ________, and $\mathbf{M}_{TB}^{-1} =$ ________.

A6. To turn by θ about a point $\mathbf{p}$: $\mathbf{M} =$ __________ __________ __________. The factor that acts on the point first is _______________.

A7. In a scene graph each node stores one _________ matrix: its frame seen from its _________. An object is drawn with the product of _____________________________, with _________ on the left.

A8. To hit a world ray $\mathbf{a} + t\mathbf{b}$ against an instance, multiply by _______; $\mathbf{a}$ gets a ___ in the third slot and $\mathbf{b}$ a ___. The t you find is _______________, provided you never _______________.

A9. In a manifold mesh every edge is used by _________ triangles and every vertex has _________________ around it. Two neighbours agree on the front when their shared edge's two vertices appear in _____________ orders.

A10. Lecture 9 used the letter $\mathbf{b}$ for three different things. Say what it meant in section 2 (changing coordinates), in section 3 (the pendulum) and in section 4 (instancing).

Part B • Wednesday's Toolbox

COMPUTE ≈ 4 min

Lecture 9 runs on Lecture 8's 3×3 matrices. Four quick ones to warm up.

B1. Write translate$(3, -1)$ as a 3×3 matrix. Apply it to the *point* $(2, 2)$ and to the *direction* $(2, 2)$. Why do the answers differ?

B2. Write rotate$(90°)$ as a 3×3 matrix and apply it to the point $(3, 1)$. Sketch both points.

B3. Let $\mathbf{M} =$ translate$(3, -1)$ rotate$(90°)$. Which move acts first? Where does $(1, 0)$ go? Then swap the order: where does $(1, 0)$ go under rotate$(90°)$ translate$(3, -1)$?

B4. Write $\mathbf{M}$ from B3 as one matrix. Then write $\mathbf{M}^{-1}$ as two moves (in which order?) and check that it sends $(3, 0)$ back to $(1, 0)$.

Lecture 9 on one card — check Part A against it, then keep it open

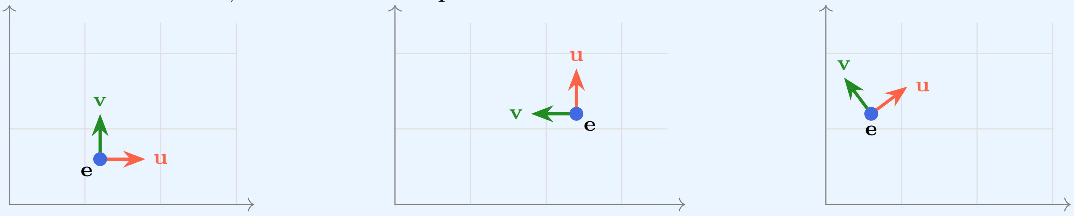
Frame: origin $\mathbf{e}$ plus arrows $\mathbf{u}, \mathbf{v}$ with $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$, $\mathbf{u} \cdot \mathbf{v} = 0$, and $\mathbf{v}$ a quarter turn counterclockwise from $\mathbf{u}$.
Frame → world: $\mathbf{p} = \mathbf{e} + u_p\mathbf{u} + v_p\mathbf{v}$.
World → frame: $\mathbf{b} = \mathbf{p} - \mathbf{e}$, then $u_p = \mathbf{u} \cdot \mathbf{b}$, $v_p = \mathbf{v} \cdot \mathbf{b}$.
A *direction* skips $\mathbf{e}$, both ways.
As matrices, for a frame C:

$$\mathbf{M}_{WC} = \begin{bmatrix} x_u & x_v & x_e \\ y_u & y_v & y_e \\ 0 & 0 & 1 \end{bmatrix} \quad \text{columns } \mathbf{u}, \mathbf{v}, \mathbf{e}$$

$$\mathbf{M}_{CW} = \begin{bmatrix} x_u & y_u & -\mathbf{u}\cdot\mathbf{e} \\ x_v & y_v & -\mathbf{v}\cdot\mathbf{e} \\ 0 & 0 & 1 \end{bmatrix} \quad \text{rows } \mathbf{u}, \mathbf{v}$$

Names: $\mathbf{M}_{XY}$ takes Y numbers to X numbers: $\mathbf{p}_X = \mathbf{M}_{XY}\mathbf{p}_Y$. $\mathbf{M}_{XY}\mathbf{M}_{YZ} = \mathbf{M}_{XZ}$, $\mathbf{M}_{XY}^{-1} = \mathbf{M}_{YX}$.
Turn about p: translate($\mathbf{p}$) rotate(θ) translate($-\mathbf{p}$).
Piece on a piece: each drawn with its hinge at its own origin; child to world = parent to world × child to parent, e.g. $\mathbf{M}_{WF} = \mathbf{M}_{WU}\mathbf{M}_{UF}$.
Scene graph: $\mathbf{M}_{\text{root}} \cdots \mathbf{M}_{\text{parent}}\mathbf{M}_{\text{local}}$; stack: push, draw, visit children, pop.
Instance: $\mathbf{M}_{OW}\mathbf{a} + t\,\mathbf{M}_{OW}\mathbf{b}$; same t; never normalize.
Mesh: 2 triangles per edge, 1 loop per vertex; a shared edge appears in opposite orders.

The three frames on this sheet. Parts C–G keep coming back to them. All three are orthonormal and right-handed: in each one, $\mathbf{v}$ is $\mathbf{u}$ turned a quarter counterclockwise.



frame S: shifted only
$\mathbf{e} = (2, 1)$
$\mathbf{u} = (1, 0)$, $\mathbf{v} = (0, 1)$

frame Q: a quarter turn
$\mathbf{e} = (4, 2)$
$\mathbf{u} = (0, 1)$, $\mathbf{v} = (-1, 0)$

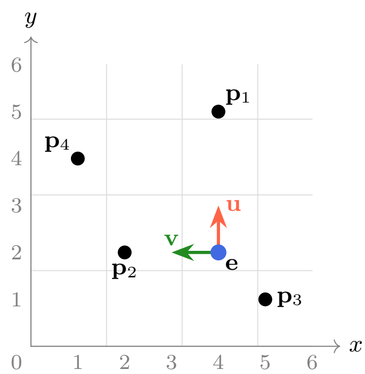
frame T: tilted (3-4-5)
$\mathbf{e} = (1, 2)$
$\mathbf{u} = (0.8, 0.6)$, $\mathbf{v} = (-0.6, 0.8)$

Part C • Count Steps on the Grid

TRACE ≈ 5 min

Frame Q: $\mathbf{e} = (4, 2)$, $\mathbf{u} = (0, 1)$ points **up**, $\mathbf{v} = (-1, 0)$ points **left**. A *world* address is read off the grid as usual. A Q *address* (u_p, v_p) is counted: start at $\mathbf{e}$, take u_p steps along $\mathbf{u}$ (up), then v_p steps along $\mathbf{v}$ (left). Walking an arrow backwards counts as negative.

point	world (x_p, y_p)	Q address (u_p, v_p)
$\mathbf{e}$	$(4, 2)$	$(0, 0)$
$\mathbf{p}_1$	________	________
$\mathbf{p}_2$	________	________
$\mathbf{p}_3$	________	________
$\mathbf{p}_4$	________	________



C1. Fill in the table: read each world address, then count each Q address.

C2. Check $\mathbf{p}_4$ with the recipe: $\mathbf{b} = \mathbf{p}_4 - \mathbf{e}$, then dot with $\mathbf{u}$ and with $\mathbf{v}$. Does it agree with your count?

C3. $\mathbf{p}_3$ has two negative numbers. In words, where does a point with $u_p < 0$ and $v_p < 0$ sit relative to $\mathbf{e}$?

C4. What is the Q address of the world origin $\mathbf{o} = (0, 0)$? Count it on the grid, then check with the recipe.

Part D • Frame to World: Walk the Arrows

COMPUTE ≈ 3 min

D1. Frame S. The point with S address $(3, 2)$ is at world __________. The world origin has S address __________. For a frame that is only shifted, converting is just __________________________.

D2. Frame Q. Which world point has Q address $(2, 3)$? Compute $\mathbf{e} + 2\mathbf{u} + 3\mathbf{v}$ and compare with Part C.

D3. Frame T. Convert the T addresses $(2, 1)$ and $(5, 0)$ to world addresses. Sketch both on the frame T picture.

D4. In frame T, the pair $(0, 2)$ could be a *point* or a *direction*. Convert both to world numbers. Why do they differ by exactly $\mathbf{e}$?

Part E • World to Frame: Subtract, Then Dot

COMPUTE ≈ 6 min

All in frame T: $\mathbf{e} = (1, 2)$, $\mathbf{u} = (0.8, 0.6)$, $\mathbf{v} = (-0.6, 0.8)$.

E1. Find the T address of the world point $(4, 3)$. Then walk back, $\mathbf{e} + u_p\mathbf{u} + v_p\mathbf{v}$, to check.

E2. Find the T address of $(5, 5)$. One of the two numbers is 0: what does that tell you about where $(5, 5)$ is?

E3. A classmate skips step one and dots $(4, 3)$ directly with $\mathbf{u}$ and $\mathbf{v}$. What do they get? Whose address is that, really?

E4. Find the T components of the world *direction* $\mathbf{d} = (2, -1)$. Do you subtract $\mathbf{e}$? Check your answer by rebuilding $\mathbf{d}$ from $\mathbf{u}$ and $\mathbf{v}$.

E5. A tile is drawn in frame T with $0 \le u \le 4$ and $0 \le v \le 2$. Which of the world points $(4, 3)$, $(5, 5)$, $(2, 4)$ are on it? Use your answers above: no new arithmetic.

E6. A3's `point_in_box` asks the same question about a box *centred* on its frame's origin: $|u_p| \le$ `half_w` and $|v_p| \le$ `half_h`. For this tile, give `half_w`, `half_h`, and the origin $\mathbf{e}'$ that frame would need (same $\mathbf{u}$, $\mathbf{v}$). Redo the test for $(4, 3)$.

Part F • Is It a Good Basis?

CLASSIFY ≈ 4 min

Three quick tests: **unit** ($x^2 + y^2 = 1$ for each arrow), **right angle** ($\mathbf{u} \cdot \mathbf{v} = 0$), **right-handed** ($\mathbf{v} = (-y_u,\ x_u)$, which is A2's `u.perp()`). Tick what holds, then give a verdict.

	u	v	unit	right angle	right-handed	verdict
(i)	$(0, -1)$	$(1, 0)$	☐	☐	☐	________
(ii)	$(1, 0)$	$(0, -1)$	☐	☐	☐	________
(iii)	$(2, 0)$	$(0, 2)$	☐	☐	☐	________
(iv)	$(0.6, -0.8)$	$(0.8, 0.6)$	☐	☐	☐	________
(v)	$(1, 0)$	$(1, 1)$	☐	☐	☐	________
(vi)	$(0.8, 0.6)$	$(0.6, -0.8)$	☐	☐	☐	________

F1. Use basis (iii) with $\mathbf{e} = (0, 0)$. The point with address $(1, 1)$ sits at world $(2, 2)$. What does "dot with each axis" return for $(2, 2)$? Which of $\mathbf{u} \cdot \mathbf{u}$ and $\mathbf{u} \cdot \mathbf{v}$ is to blame?

F2. Use basis (v) with $\mathbf{e} = (0, 0)$. The address $(0, 1)$ is the world point $(1, 1)$. What does the dot recipe return, and why?

F3. A3's `to_local` must work for *any* object matrix, including one that scales. How can it, if the dot recipe needs an orthonormal basis?

Part G • The Matrix Writes Itself

COMPUTE ≈ 6 min

G1. Write $\mathbf{M}_{WQ}$, frame Q's frame-to-canonical matrix, by copying columns. Push the Q address $(2, 3)$ through it and compare with D2.

G2. Write $\mathbf{M}_{WT}$. Push the T address $(3, -1)$ through it and compare with E1.

G3. Read backwards: $\mathbf{M} = \begin{bmatrix} 0 & 1 & 5 \\ -1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix}$ is some frame's frame-to-canonical matrix. Read off $\mathbf{e}$, $\mathbf{u}$, $\mathbf{v}$ and sketch the frame. Which basis from Part F is it? Where does the address $(2, 0)$ land?

G4. Write $\mathbf{M}_{TW}$ with the card's rule: rows $\mathbf{u}$ and $\mathbf{v}$, last column $-\mathbf{u} \cdot \mathbf{e}$ and $-\mathbf{v} \cdot \mathbf{e}$. Push the world points $(2, 4)$ and $(4, 3)$ through it; compare with D3 and E1.

G5. Column 3 of $\mathbf{M}_{TW}$ is the world origin, written in T's numbers. Check it with subtract-then-dot. What are columns 1 and 2?

G6. Which matrix, $\mathbf{M}_{WT}$ or $\mathbf{M}_{TW}$? (a) To draw E5's tile on screen from its corners. (b) To decide whether a mouse click at a world point lands on the tile.

Part H • Names That Cancel

CLASSIFY ≈ 3 min

A desk lamp: frames W (the desk, the world), B (the lamp's base), A (its arm), H (its head). The scene stores three local matrices: $\mathbf{M}_{WB}$, $\mathbf{M}_{BA}$, $\mathbf{M}_{AH}$.

H1. In $\mathbf{p}_A = \mathbf{M}_{AH}\,\mathbf{p}_H$, which numbers go in, and which come out? Say it in one plain sentence.

H2. Which products make sense (neighbouring letters match)? Name each result. (a) $\mathbf{M}_{WB}\mathbf{M}_{BA}$ (b) $\mathbf{M}_{BA}\mathbf{M}_{WB}$ (c) $\mathbf{M}_{WB}\mathbf{M}_{BA}\mathbf{M}_{AH}$ (d) $\mathbf{M}_{AH}\mathbf{M}_{BA}$ (e) $\mathbf{M}_{BA}\mathbf{M}_{AH}$

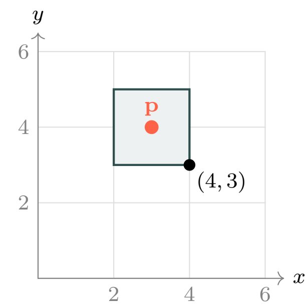
H3. Write $\mathbf{M}_{HW}$ (a click, in world numbers, turned into head numbers) using only the stored matrices and inverses. Check the letters.

H4. In the book's words, $\mathbf{M}_{WH}$ is the head's __________________ matrix and $\mathbf{M}_{HW}$ its __________________ matrix.

Part I • Turn About a Point

TRACE ≈ 5 min

A square has corners $(2,3)$, $(4,3)$, $(4,5)$, $(2,5)$ and centre $\mathbf{p} = (3,4)$. We want to spin it in place about $\mathbf{p}$. Sketch every answer on the grid: a turn that keeps the square on top of itself is right, anything else is a bug.



I1. Write the three matrices of the sandwich for a quarter turn (90°) about $\mathbf{p}$, in the order you multiply them. Which one touches the point first?

I2. Multiply them out. Lecture shortcut: the turn block stays as it is, and the last column is $\mathbf{p} - \mathbf{R}\mathbf{p}$.

I3. Check your $\mathbf{M}$: where do the pivot $(3,4)$ and the corner $(4,3)$ go? Does the sketch show a quarter turn counter-clockwise about the centre?

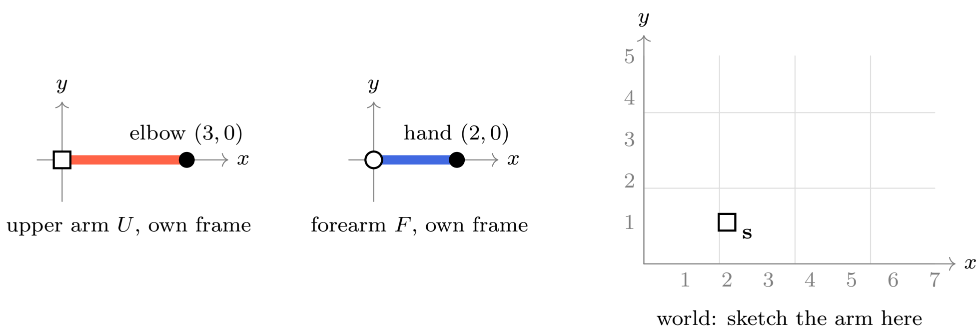
I4. A half turn (180°) about $\mathbf{p}$: here $\mathbf{R} = -\mathbf{I}$, so the last column is $2\mathbf{p}$. Write $\mathbf{M}$ and find where $(4,3)$ goes.

I5. Two broken versions: (a) rotate(90°) alone; (b) translate$(-3,-4)$ rotate(90°) translate$(3,4)$, the outer factors swapped. Where does the pivot $(3,4)$ go in each? Which single check catches both?

Part J • A Two-Piece Arm

COMPUTE ≈ 6 min

The lecture's pendulum with new numbers. Each piece is drawn *once*, in its own frame, with its hinge at its own origin and its rod along its own $+x$. The **upper arm** U runs from its hinge to the elbow at $(3,0)$; the **forearm** F runs from its hinge to the hand at $(2,0)$. The shoulder sits at world $\mathbf{s} = (2,1)$. Pose: the upper arm is turned $\theta = 90°$, and the forearm is turned $\phi = 90°$ *relative to the upper arm.*



J1. The elbow is at $(3,0)$ in whose numbers: U's, F's or the world's? Why is that the useful choice?

J2. Write $\mathbf{R}_\theta$, $\mathbf{T}_\mathbf{s}$ and $\mathbf{M}_{WU} = \mathbf{T}_\mathbf{s}\mathbf{R}_\theta$. Where is the elbow in the world? Sketch the upper arm.

J3. Write $\mathbf{M}_{UF} =$ translate$(3,0)$ rotate$(90°)$. Where is the hand, in U's numbers?

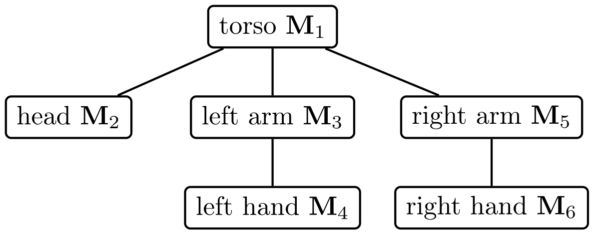
J4. Compute $\mathbf{M}_{WF} = \mathbf{M}_{WU}\mathbf{M}_{UF}$ and push the hand through it. Check in two steps: apply $\mathbf{M}_{WU}$ to your answer to J3. What is the forearm's total turn in the world?

J5. The shoulder relaxes to $\theta = 0$; the elbow stays bent at $\phi = 90°$. Which of $\mathbf{M}_{WU}$, $\mathbf{M}_{UF}$, $\mathbf{M}_{WF}$ do you rebuild by hand, and which just follow? Where are the elbow and the hand now?

Part K • The Tree and the Stack

TRACE ≈ 5 min

A small robot as a scene graph. Each box holds a part and its local matrix, relative to its parent. Children are visited left to right: head, left arm, right arm.



K1. Write the product that draws each of the six parts.

K2. Which *one* matrix do you change to (a) wave the right hand only, (b) lift the whole right arm, hand and all, (c) walk the whole robot across the screen, (d) nod the head?

K3. Traverse the tree with push, draw, visit children, pop. Complete rows 4 to 12.

step	action	stack, bottom → top	drawn now, with
1	push $\mathbf{M}_1$	$\mathbf{M}_1$	torso, $\mathbf{M}_1$
2	push $\mathbf{M}_2$	$\mathbf{M}_1$ $\mathbf{M}_2$	head, $\mathbf{M}_1\mathbf{M}_2$
3	pop	$\mathbf{M}_1$	—
4	______	______	______
5	______	______	______
6	______	______	______
7	______	______	______
8	______	______	______
9	______	______	______
10	______	______	______
11	______	______	______
12	______	______	______

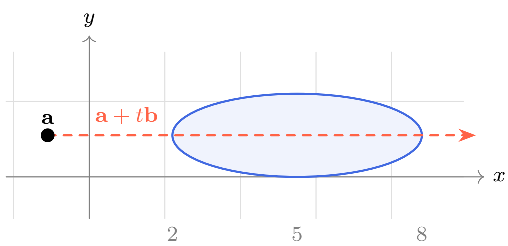
K4. Now the *left arm* forgets its pop (its hand still pops). Which product draws the right arm? What is left on the stack at the end, and what happens on the next animation frame?

K5. A3's `TransformStack.push(m)` makes the new top `current @ m`. Starting from the identity, push translate$(0, 4)$, then push rotate$(90°)$. Where is the point $(2, 0)$ drawn? Where would it be drawn if `push` stored `m` alone?

Part L • Move the Ray, Not the Shape

COMPUTE ≈ 4 min

The base object is the unit circle, $x^2 + y^2 = 1$. One instance places it with $\mathbf{M}_{WO} = \text{translate}(5, 1)\,\text{scale}(3, 1)$, so in the world it is an ellipse. A ray starts at $\mathbf{a} = (-1, 1)$ with direction $\mathbf{b} = (6, 0)$: its points are $\mathbf{a} + t\mathbf{b}$.



L1. Describe the ellipse: its centre, its leftmost and rightmost points, its top. Then write $\mathbf{M}_{OW}$ as two moves (in which order?) and as one matrix.

L2. Transform the ray: $\mathbf{M}_{OW}\mathbf{a}$ and $\mathbf{M}_{OW}\mathbf{b}$. Which third slot does each one get?

L3. The object-space ray $(-2, 0) + t\,(2, 0)$ first meets the unit circle at $x = -1$. Find t, then put the same t into the *world* ray. Is that point on the ellipse? Where does the ray leave it?

L4. Two bugs. (a) The code normalizes the object-space direction to $(1, 0)$: which t does it find, and which world point? (b) The code transforms $\mathbf{b}$ as a point: which direction comes out?

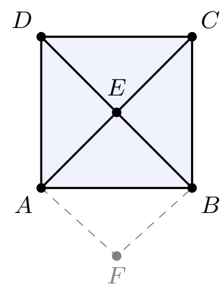
Part M • Mesh Rules

CLASSIFY ≈ 3 min

A flat patch: the square A, B, C, D with centre E, cut into four triangles, each listed counterclockwise:

$$(A, B, E), \quad (B, C, E), \quad (C, D, E), \quad (D, A, E).$$

The dashed point F below the square is used in M4 only.



M1. List the eight edges and count the triangles on each. Is the patch a manifold, a manifold with boundary, or neither?

M2. Find the shared edge BE in (A, B, E) and in (B, C, E): in which order does each list it? Do the two triangles agree on the front?

M3. A file lists the third triangle as (C, E, D) instead. Same triangle? Consistent with its neighbour (B, C, E)?

M4. Add a flap: one triangle on the edge AB with its third corner at F. Should it be listed (A, B, F) or (B, A, F)? And what would a second flap on AB break?

Part N • Find the Bug

FIX ≈ 3 min

Each line is one small change away from right. A3's names are used as in the handout. Name the bug and give the fix; the symptom's numbers come from the parts above.

N1. Symptom: for frame T, `frame_address(Vec2(4, 3), e, u, v)` returns $(5, 0)$; Part E says $(3, -1)$.

```
def frame_address(p, e, u, v):
    return Vec2(u.dot(p), v.dot(p))
```

Bug & fix: ______________________________

N2. Symptom: the square of Part I should spin in place but flies off towards $(-11, 2)$.

```
spin = compose(Mat3.translation(-3, -4), Mat3.rotation(math.pi / 2),
               Mat3.translation(3, 4))
```

Bug & fix: ______________________________

N3. Symptom: a wheel whose local transform is rotate(90°), on a car at translate(4, 0), has its centre drawn at $(0, 4)$ instead of $(4, 0)$.

```
world = node.transform @ parent_world
```

Bug & fix: ______________________________

N4. Symptom: every ray misses an instance that has been moved, even rays aimed straight at it.

```
b_obj = m_ow.apply_point(ray_dir)
```

Bug & fix: ______________________________

Part O • One Sentence Each

EXPLAIN ≈ 4 min

O1. A point on the page has no numbers of its own. Why does it get a different address in every frame?

O2. Why do we subtract $\mathbf{e}$ for a point, but not for a direction?

O3. Read as a motion, $\mathbf{M}_{WT}$ carries a tile from the world origin out to $\mathbf{e}$. Read as a conversion, it turns T numbers into world numbers. Why are these the same thing?

O4. Why does a node's own matrix sit at the *right* end of its product?

O5. Why does a ray tracer move the ray into the object's frame instead of moving the object into the world?

Part P • Bonus: Find the Lost Pivot ★

DESIGN ≈ 10 min

For the brave. Part I built a turn about a pivot and got a matrix whose last column is $\mathbf{p} - \mathbf{R}\mathbf{p}$. Go backwards: given a matrix that turns, find the one point it leaves where it is.

P1. Someone built $\mathbf{M} = \begin{bmatrix} 0 & -1 & 5 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix}$ with a sandwich and lost the pivot. Find the point with $\mathbf{M}\mathbf{p} = \mathbf{p}$: write the two equations and solve them. Then rebuild the sandwich from your $\mathbf{p}$ and check its last column.

P2. Do the same for the half turn $\begin{bmatrix} -1 & 0 & 6 \\ 0 & -1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$. There is a one-line shortcut: what is it?

P3. Why does translate$(2, 0)$ have no such point? For which angles θ can the pivot fail to exist?

P4. Chain two quarter turns: first rotate(90°) about the origin, then Part I's quarter turn about $(3, 4)$. Multiply, say what kind of move the product is, and find its pivot. Check the pivot by following it through both turns.

P5. Part J's forearm matrix $\mathbf{M}_{WF} = \begin{bmatrix} -1 & 0 & 2 \\ 0 & -1 & 4 \\ 0 & 0 & 1 \end{bmatrix}$ is a half turn. About which point? Where does it send the forearm's hinge $(0, 0)$, and why is your pivot exactly halfway?

Stuck on more than three items? Re-read the Lecture 9 slides — everything above Part P comes straight from them. Start with "World to frame: subtract $\mathbf{e}$, then dot with each axis", "Multiply: $\mathbf{u}$, $\mathbf{v}$, $\mathbf{e}$ land in the columns", "Naming a matrix by its two frames" and "Bottom piece rides along with the top". Most items here are one of those four slides with new numbers.

Answer Key

Check your work *after* attempting everything. If an answer surprises you, that item is worth re-reading in the slides.

A1. An **origin e** plus a **basis u, v**. Each arrow has **length 1**, and $\mathbf{u}\cdot\mathbf{v}=\mathbf{0}$ (a right angle). Right-handed: $\mathbf{v}$ is **a quarter turn counterclockwise** from $\mathbf{u}$, the way $\mathbf{y}$ sits relative to $\mathbf{x}$.

A2. x_p: the x coordinate of the point $\mathbf{p}$, a world number. u_p: how many steps of $\mathbf{u}$ the point $\mathbf{p}$ is from $\mathbf{e}$, a frame number. x_u: the x coordinate of the *arrow* $\mathbf{u}$, the world number we store for it. Three different things; the subscript always names what is being measured.

A3. $\mathbf{p}=\mathbf{e}+u_p\mathbf{u}+v_p\mathbf{v}$. Then $\mathbf{b}=\mathbf{p}-\mathbf{e}$, $u_p=\mathbf{u}\cdot\mathbf{b}$, $v_p=\mathbf{v}\cdot\mathbf{b}$.

A4. Columns $\mathbf{u}$, $\mathbf{v}$, $\mathbf{e}$; bottom row 0 0 1. **Frame** numbers in, **world** numbers out. ("Canonical" is the book's word for the world frame.)

A5. B numbers in, T numbers out: the subscript reads *to, from.* $\mathbf{M}_{WT}\mathbf{M}_{TB}=\mathbf{M}_{WB}$ (the inner T's cancel), and the inverse swaps the letters: $\mathbf{M}_{BT}$.

A6. translate($\mathbf{p}$) rotate(θ) translate($-\mathbf{p}$). The rightmost, translate($-\mathbf{p}$), acts first: it carries the pivot to the origin, where rotate can turn about it.

A7. A **local** matrix, relative to its **parent**. Drawn with the product of the matrices on the **path from the root down to it**, the **root** on the left and its own matrix on the right.

A8. $\mathbf{M}_{OW}$ (world to object, the inverse of the instance's matrix $\mathbf{M}_{WO}$). $\mathbf{a}$ is a point: 1. $\mathbf{b}$ is a direction: 0. The t is **the same in both spaces**, provided you never **normalize** the transformed direction.

A9. **Exactly two** triangles per edge; **one complete loop** of triangles around every vertex. (With a boundary: one or two per edge, one edge-connected fan per vertex.) The shared vertices appear in **opposite** orders.

A10. Section 2: the arrow $\mathbf{b}=\mathbf{p}-\mathbf{e}$ from the frame's origin to a point. Section 3: the point where the bottom piece hangs, in the top piece's numbers. Section 4: the ray's direction in $\mathbf{a}+t\mathbf{b}$. The letter $\mathbf{p}$ was just as busy: the plane's frame origin (section 1's 3D picture), any point being converted, the pivot of a turn, and the pendulum's top hinge. When a formula confuses you, first ask which section it came from.

B1. $\begin{bmatrix}1&0&3\\0&1&-1\\0&0&1\end{bmatrix}$. Point $(2,2,1)\to(5,1,1)$. Direction $(2,2,0)\to(2,2,0)$, unchanged. The third slot multiplies the last column: a place moves when the world slides, a heading does not.

B2. $\begin{bmatrix}0&-1&0\\1&0&0\\0&0&1\end{bmatrix}$ $(\cos 90^\circ=0,\ \sin 90^\circ=1)$. $(3,1)\to(-1,3)$: a quarter turn counterclockwise about the origin.

B3. The rotation acts first (it is next to the point). $(1,0)\to(0,1)\to(3,0)$. Swapped: $(1,0)\to(4,-1)\to(1,4)$. Different answers: order matters.

B4. $\mathbf{M}=\begin{bmatrix}0&-1&3\\1&0&-1\\0&0&1\end{bmatrix}$: the turn block from the rotation, the shift in the last column. $\mathbf{M}^{-1}=$ rotate(-90°) translate($-3,1$): undo the last move first. $(3,0)\to(0,1)\to(1,0)$. Keep this $\mathbf{M}$ in mind: it is a frame matrix, with $\mathbf{u}=(0,1)$, $\mathbf{v}=(-1,0)$ and $\mathbf{e}=(3,-1)$ in its columns — Part G.

C1. $\mathbf{p}_1=(4,5)$, Q address $(3,0)$: three steps up, none left. $\mathbf{p}_2=(2,2)$, $(0,2)$: two steps left. $\mathbf{p}_3=(5,1)$, $(-1,-1)$: one step down (against $\mathbf{u}$), one step right (against $\mathbf{v}$). $\mathbf{p}_4=(1,4)$, $(2,3)$: two up, three left.

C2. $\mathbf{b}=(1,4)-(4,2)=(-3,2)$. $\mathbf{u}\cdot\mathbf{b}=2$, $\mathbf{v}\cdot\mathbf{b}=3$: $(2,3)$, the same as the count. The dot product *is* the counting, done with numbers.

C3. Behind $\mathbf{e}$ along $\mathbf{u}$ (below it, here) and on the side $\mathbf{v}$ points away from (to its right, here). Negative only means "walk that arrow backwards".

C4. $\mathbf{b}=(0,0)-(4,2)=(-4,-2)$: $\mathbf{u}\cdot\mathbf{b}=-2$, $\mathbf{v}\cdot\mathbf{b}=4$, so $(-2,4)$. On the grid: from $\mathbf{e}$, two steps down to $(4,0)$, then four steps left to $(0,0)$. To frame Q the world's origin is just another point.

D1. $(5,3)$; $(-2,-1)$; adding $\mathbf{e}$ (frame to world) or subtracting it (world to frame). The arrows are $\mathbf{x}$ and $\mathbf{y}$, so the dot products change nothing.

D2. $(4,2)+(0,2)+(-3,0)=(1,4)$: that is $\mathbf{p}_4$, found the other way round.

D3. $(2,1)$: $(1+1.6-0.6,\ 2+1.2+0.8)=(2,4)$. $(5,0)$: $(1+4,\ 2+3)=(5,5)$, five steps straight out along $\mathbf{u}$.

D4. Point: $\mathbf{e}+2\mathbf{v}=(-0.2,\ 3.6)$. Direction: $2\mathbf{v}=(-1.2,\ 1.6)$. A direction has no position, so there is no origin to start from; only the arrows are added. In matrix form, its 0 in the third slot skips the last column, which is where $\mathbf{e}$ lives.

E1. $\mathbf{b}=(3,1)$. $u_p=2.4+0.6=3$, $v_p=-1.8+0.8=-1$: $(3,-1)$. Walk back: $(1+2.4+0.6,\ 2+1.8-0.8)=(4,3)$.

E2. $\mathbf{b}=(4,3)$: $u_p=3.2+1.8=5$, $v_p=-2.4+2.4=0$, so $(5,0)$. $v_p=0$ means the point lies on the line through $\mathbf{e}$ along $\mathbf{u}$: D3 said the same thing backwards.

E3. $(\mathbf{u}\cdot(4,3),\ \mathbf{v}\cdot(4,3))=(5,0)$: E2's answer, the address of $(5,5)$, a different point. The mistake does not crash and does not look odd; it quietly measures from the world origin instead of from $\mathbf{e}$. Subtract first, and check by walking back.

E4. No: a direction is already an arrow. $(\mathbf{u}\cdot\mathbf{d},\ \mathbf{v}\cdot\mathbf{d})=(1.6-0.6,\ -1.2-0.8)=(1,-2)$. Check: $1\mathbf{u}-2\mathbf{v}=(0.8+1.2,\ 0.6-1.6)=(2,-1)$.

E5. $(4,3)\to(3,-1)$: $v<0$, off. $(5,5)\to(5,0)$: $u>4$, off. $(2,4)\to(2,1)$: on; in fact it is the tile's centre

(D3). Two comparisons per point, however the tile is tilted.

E6. `half_w` $= 2$, `half_h` $= 1$, and $\mathbf{e}' = (2, 4)$, the tile's centre. $\mathbf{b} = (4, 3) - (2, 4) = (2, -1)$: $u_p = 1.6 - 0.6 = 1$, $v_p = -1.2 - 0.8 = -2$. $|-2| > 1$: off, as before. Moving the origin changes the numbers, never the answer.

F. (i) all three: **good**. $\mathbf{u}$ points down, $\mathbf{v}$ right (it returns in G3).
(ii) unit and right angle, but $(-y_u, x_u) = (0, 1) \neq \mathbf{v}$: **left-handed**, a mirror; anything drawn in it comes out flipped.
(iii) right angle and right-handed, but each arrow has length 2: **not unit**.
(iv) all three ($0.48 - 0.48 = 0$; $(-y_u, x_u) = (0.8, 0.6)$): **good**.
(v) $\mathbf{v}$ has length $\sqrt{2}$ and $\mathbf{u} \cdot \mathbf{v} = 1$: **neither unit nor at right angles**.
(vi) unit and right angle, but $(-y_u, x_u) = (-0.6, 0.8) = -\mathbf{v}$: **left-handed**.

F1. $(\mathbf{u} \cdot \mathbf{b},\ \mathbf{v} \cdot \mathbf{b}) = (4, 4)$, four times too big. $\mathbf{u} \cdot \mathbf{u} = 4$ instead of 1, so the coordinate comes out multiplied by 4.

F2. $(1, 2)$ instead of $(0, 1)$. $\mathbf{u} \cdot \mathbf{b} = \mathbf{u} \cdot \mathbf{v} = 1$: the $\mathbf{v}$ part leaks into u_p because the arrows are not at right angles; and $\mathbf{v} \cdot \mathbf{b} = \mathbf{v} \cdot \mathbf{v} = 2$ because $\mathbf{v}$ is not unit.

F3. It does not use the dot recipe: it applies the full inverse matrix (T5), which undoes any invertible matrix, scaled or not. For an orthonormal frame that inverse has rows $\mathbf{u}, \mathbf{v}$ and last column $-\mathbf{u} \cdot \mathbf{e}, -\mathbf{v} \cdot \mathbf{e}$, so applying it *is* subtract-then-dot. The dot recipe is the orthonormal shortcut; the inverse is the general tool.

G1. $\mathbf{M}_{WQ} = \begin{bmatrix} 0 & -1 & 4 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix}$. $(2, 3, 1) \to (-3 + 4,\ 2 + 2) = (1, 4)$: the same point as D2 and $\mathbf{p}_4$.

G2. $\mathbf{M}_{WT} = \begin{bmatrix} 0.8 & -0.6 & 1 \\ 0.6 & 0.8 & 2 \\ 0 & 0 & 1 \end{bmatrix}$. $(3, -1, 1) \to (2.4 + 0.6 + 1,\ 1.8 - 0.8 + 2) = (4, 3)$, as E1 said.

G3. $\mathbf{u} = (0, -1)$ (down), $\mathbf{v} = (1, 0)$ (right), $\mathbf{e} = (5, 3)$: basis (i), a good one. $(2, 0) \to \mathbf{e} + 2\mathbf{u} = (5, 1)$.

G4. $\mathbf{u} \cdot \mathbf{e} = 0.8 + 1.2 = 2$, $\mathbf{v} \cdot \mathbf{e} = -0.6 + 1.6 = 1$, so $\mathbf{M}_{TW} = \begin{bmatrix} 0.8 & 0.6 & -2 \\ -0.6 & 0.8 & -1 \\ 0 & 0 & 1 \end{bmatrix}$. $(2, 4) \to (1.6 + 2.4 - 2,\ -1.2 + 3.2 - 1) = (2, 1)$; $(4, 3) \to (3.2 + 1.8 - 2,\ -2.4 + 2.4 - 1) = (3, -1)$. Both match: this one matrix is the subtract-then-dot recipe.

G5. $\mathbf{b} = \mathbf{o} - \mathbf{e} = (-1, -2)$: $\mathbf{u} \cdot \mathbf{b} = -0.8 - 1.2 = -2$, $\mathbf{v} \cdot \mathbf{b} = 0.6 - 1.6 = -1$. Yes, $(-2, -1)$. Column 1 is the world's $\mathbf{x}$ arrow in T's numbers, $(0.8, -0.6)$; column 2 is $\mathbf{y}$, $(0.6, 0.8)$. Both directions of conversion follow one rule: columns are the other frame's arrows and origin, in my numbers.

G6. (a) $\mathbf{M}_{WT}$: the corners are stored in T numbers and the screen wants world numbers. (b) $\mathbf{M}_{TW}$: the click arrives in world numbers and the test is easy in T numbers (E5). That is A3's `to_local`.

H1. Head numbers in, arm numbers out: given a point of the lamp's head, it says where that point is, measured in the arm's frame.

H2. (a) yes, $\mathbf{M}_{WA}$. (b) no: A meets W. (c) yes, $\mathbf{M}_{WH}$, the head's world matrix: the path from the root. (d) no: H meets B. (e) yes, $\mathbf{M}_{BH}$.

H3. $\mathbf{M}_{HW} = \mathbf{M}_{HA}\mathbf{M}_{AB}\mathbf{M}_{BW} = \mathbf{M}_{AH}^{-1}\mathbf{M}_{BA}^{-1}\mathbf{M}_{WB}^{-1}$. Read the subscripts in order, HA, AB, BW: each neighbouring pair of letters matches and cancels. It is also $(\mathbf{M}_{WB}\mathbf{M}_{BA}\mathbf{M}_{AH})^{-1}$, and inverting reverses the order.

H4. Frame-to-canonical; canonical-to-frame. Both names describe the same direction as the letters: the English name reads left to right (from, to), the subscript reads right to left (to, from).

I1. translate$(3, 4)$ rotate$(90°)$ translate$(-3, -4)$: $\begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{bmatrix}$. The rightmost, translate$(-3, -4)$, touches the point first and carries the pivot to the origin.

I2. $\mathbf{R}\mathbf{p} = (-4, 3)$, so $\mathbf{p} - \mathbf{R}\mathbf{p} = (7, 1)$ and $\mathbf{M} = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$.

I3. $(3, 4) \to (-4 + 7,\ 3 + 1) = (3, 4)$: the pivot stays. $(4, 3) \to (-3 + 7,\ 4 + 1) = (4, 5)$: the lower-right corner moves to the upper-right. Counterclockwise, in place.

I4. $\begin{bmatrix} -1 & 0 & 6 \\ 0 & -1 & 8 \\ 0 & 0 & 1 \end{bmatrix}$. $(4, 3) \to (2, 5)$: the diagonally opposite corner, as a half turn should do.

I5. (a) $(-4, 3)$: the square orbits the world origin. (b) $(3, 4) \to (6, 8) \to (-8, 6) \to (-11, 2)$: it flies off. The check: **the pivot must map to itself**. Try it on every sandwich you write in A3.

J1. U's. The elbow is a fixed spot on the upper arm, so its U address never changes however the arm moves, and the forearm's matrix never has to be rebuilt when the shoulder turns. (In the lecture, $\mathbf{b}$ lived in the top piece's frame for the same reason.)

J2. $\mathbf{R}_\theta = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\mathbf{T}_\mathbf{s} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$, $\mathbf{M}_{WU} = \begin{bmatrix} 0 & -1 & 2 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$. Elbow $(3, 0) \to (2, 4)$: the upper arm points straight up from $(2, 1)$ to $(2, 4)$.

J3. $\mathbf{M}_{UF} = \begin{bmatrix} 0 & -1 & 3 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Hand $(2, 0) \to (3, 2)$: in the upper arm's own picture, where its rod lies along $+x$, the forearm points straight up from the elbow.

J4. $\mathbf{M}_{WF} = \begin{bmatrix} -1 & 0 & 2 \\ 0 & -1 & 4 \\ 0 & 0 & 1 \end{bmatrix}$. Hand $\to (0, 4)$. Two steps: $\mathbf{M}_{WU}(3, 2) = (-2 + 2,\ 3 + 1) = (0, 4)$, the same. The turn block is rotate$(180°)$: turns along a chain add up, $90° + 90°$. The forearm points left, from the elbow $(2, 4)$ to the hand $(0, 4)$.

J5. Only $\mathbf{M}_{WU}$ changes, to translate$(2, 1)$. $\mathbf{M}_{UF}$ is untouched, and $\mathbf{M}_{WF} = \mathbf{M}_{WU}\mathbf{M}_{UF} = \begin{bmatrix} 0 & -1 & 5 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ follows by multiplying. Elbow $(5, 1)$, hand $(5, 3)$: the upper arm lies flat to the right and the forearm points straight up

$(0^\circ + 90^\circ)$.

K1. Torso $\mathbf{M}_1$; head $\mathbf{M}_1\mathbf{M}_2$; left arm $\mathbf{M}_1\mathbf{M}_3$; left hand $\mathbf{M}_1\mathbf{M}_3\mathbf{M}_4$; right arm $\mathbf{M}_1\mathbf{M}_5$; right hand $\mathbf{M}_1\mathbf{M}_5\mathbf{M}_6$. Root on the left, the part's own matrix on the right.

K2. (a) $\mathbf{M}_6$, (b) $\mathbf{M}_5$, (c) $\mathbf{M}_1$, (d) $\mathbf{M}_2$. One edit each, and nobody's world position is ever typed in.

K3. 4: push $\mathbf{M}_3$, stack $\mathbf{M}_1\,\mathbf{M}_3$, left arm. 5: push $\mathbf{M}_4$, stack $\mathbf{M}_1\,\mathbf{M}_3\,\mathbf{M}_4$, left hand. 6: pop, $\mathbf{M}_1\,\mathbf{M}_3$. 7: pop, $\mathbf{M}_1$. 8: push $\mathbf{M}_5$, $\mathbf{M}_1\,\mathbf{M}_5$, right arm. 9: push $\mathbf{M}_6$, $\mathbf{M}_1\,\mathbf{M}_5\,\mathbf{M}_6$, right hand. 10: pop, $\mathbf{M}_1\,\mathbf{M}_5$. 11: pop, $\mathbf{M}_1$. 12: pop, empty. Six pushes, six pops; the stack is never deeper than the tree (3). $\mathbf{M}_1$ was pushed once and reused by all five other parts.

K4. Right arm: $\mathbf{M}_1\mathbf{M}_3\mathbf{M}_5$, so it rides on the left arm; right hand $\mathbf{M}_1\mathbf{M}_3\mathbf{M}_5\mathbf{M}_6$. At the end $\mathbf{M}_1$ is still on the stack. The next frame starts on top of it, so the torso is drawn with $\mathbf{M}_1\mathbf{M}_1$: the robot jumps by its own offset again, and further every frame. Pushes and pops must pair up.

K5. The top is translate$(0, 4)$ rotate(90°): $(2, 0) \to (0, 2) \to (0, 6)$. Storing `m` alone leaves just rotate(90°) on top, and the point lands at $(0, 2)$: the child forgets that its parent lifted it by 4.

L1. Centre $(5, 1)$; it runs from $x = 2$ to $x = 8$; top $(5, 2)$. $\mathbf{M}_{OW} = \text{scale}(\frac{1}{3}, 1)\ \text{translate}(-5, -1)$: undo the translation first. As one matrix, $\begin{bmatrix} \frac{1}{3} & 0 & -\frac{5}{3} \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$.

L2. $\mathbf{a}$ is a point, $(-1, 1, 1) \to (-2, 0)$. $\mathbf{b}$ is a direction, $(6, 0, 0) \to (2, 0)$: the translation never touches it.

L3. $-2 + 2t = -1$, so $t = 0.5$. World: $\mathbf{a} + 0.5\mathbf{b} = (2, 1)$, the ellipse's leftmost point. It leaves at $x = 1$: $t = 1.5$, world $(8, 1)$, the rightmost point.

L4. (a) $-2 + t = -1$ gives $t = 1$, and the world point is $\mathbf{a} + \mathbf{b} = (5, 1)$: the ellipse's *centre*, nowhere near its edge. (b) $(6, 0, 1) \to (\frac{1}{3}, -1)$: the ray now heads downwards and misses the circle altogether. A direction must never pick up the translation.

M1. AE, BE, CE, DE: two each. AB, BC, CD, DA: one each, the outer edges. Every edge has one or two, E has a full loop, and each corner has one fan of two triangles joined along an edge: a **manifold with boundary**. Not watertight, which is fine for a flat patch.

M2. (A, B, E) has $B \to E$. (B, C, E), read round the loop $B \to C \to E \to B$, has $E \to B$. Opposite orders: **consistent**.

M3. Same three corners, but $C \to E \to D$ runs clockwise. Its edge CE appears as $C \to E$, the same order as in (B, C, E): **inconsistent**. The two neighbours disagree about which side is the front.

M4. (B, A, F): (A, B, E) already uses $A \to B$, so the new neighbour must use $B \to A$ (and $B \to A \to F$ is indeed counterclockwise). A second flap would give AB three triangles: a fin, **not a manifold**.

N1. It dots the point instead of the arrow from $\mathbf{e}$. Fix: `b = p - e`, then `return Vec2(u.dot(b), v.dot(b))`. (Only valid for an orthonormal frame, F3.)

N2. The outer translations are swapped: `compose` reads right to left, so this moves the pivot *away* before turning (I5b). Fix: `Mat3.translation(3, 4)` first in the list, `Mat3.translation(-3, -4)` last.

N3. The product is reversed: the car's move acts first and the wheel's turn then swings it about the world origin. Fix: `world = parent_world @ node.transform`, the parent on the left and the node's own matrix on the right.

N4. A direction transformed as a point picks up the translation (L4b). Fix: `m_ow.apply_vector(ray_dir)`, and do not normalize the result (L4a).

O1. An address counts steps along a frame's arrows from that frame's origin; change the origin or the arrows and the steps change, but the place does not.

O2. The dot products measure an arrow, and a point becomes an arrow only once we say where it starts ($\mathbf{e}$); a direction already is an arrow.

O3. The tile's corners start out with their T numbers, so carrying the tile out to where frame T sits and then reading its corners off the world grid produces exactly their world numbers ("move the point, or move the axes").

O4. It acts first: it places the node in its parent's frame, and only then do the ancestors' matrices carry that frame out to the world.

O5. In its own frame the object is the plain base shape, with an easy hit test, and moving one ray costs two matrix products; the same base shape can then be shared by many instances.

P1. $x = -y + 5$ and $y = x - 1$, so $x = -(x - 1) + 5$, giving $x = 3$, $y = 2$: $\mathbf{p} = (3, 2)$. Check: $\mathbf{Rp} = (-2, 3)$ and $\mathbf{p} - \mathbf{Rp} = (5, -1)$, the last column.

P2. $x = -x + 6$ and $y = -y + 2$, so $\mathbf{p} = (3, 1)$. Shortcut: for a half turn $\mathbf{Rp} = -\mathbf{p}$, so the last column is $2\mathbf{p}$ and the pivot is half of it.

P3. $x = x + 2$ has no solution: a slide moves every point. In general we solve $(\mathbf{I} - \mathbf{R})\mathbf{p} = \mathbf{t}$, and $|\mathbf{I} - \mathbf{R}| = (1 - \cos\theta)^2 + \sin^2\theta = 2 - 2\cos\theta$, which is zero only for $\theta = 0$. So every rigid motion that actually turns is a turn about exactly one point.

P4. $\begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 7 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$: a half turn $(90^\circ + 90^\circ)$ about $(3.5, 0.5)$. Check: $(3.5, 0.5) \to (-0.5, 3.5) \to$

$(-3.5+7,\ -0.5+1) = (3.5, 0.5)$. Two quarter turns about two different points make a half turn about a third point.

P5. Pivot $(1, 2)$, half of the last column. The hinge $(0, 0)$ goes to $(2, 4)$, the elbow, as it must. A half turn swaps each point with its mirror image through the pivot, so the pivot sits halfway between $(0, 0)$ and $(2, 4)$.