CH12 — Flow Builder: Interaction Model (R1)

Generated: 2026-01-02 • Bundle ID: HZ-V1

Doc ID:	HZ-V1-CH12_Flow_Builder_Interaction_Model_R1
Revision:	R1 (2026-01-02)
Status:	Draft
Depends on:	CH01, CH02, CH03, CH04, CH07, CH08, CH30, CH31
Related:	CH13, CH14, CH15, CH16, CH18, CH20–CH22, CH28–CH29, CH35–CH37
Supersedes:	—
Owned Decisions:	Flow builder gestures + selection model; branching UX up to 10; merges and dangling paths all
Open Questions / Placeholders:	Exact min/max zoom; default auto-layout spacing; haptic/animation tuning; performance thresh

0. Scope of This Chapter

- This chapter defines the Flow Builder’s user-facing interaction model: gestures, selection, adding/replacing/deleting nodes, connecting edges, branching (up to 10), merging, dangling paths, pan/zoom, orientation, and in-editor guest/paywall gating behavior.
- This chapter does NOT define database schema or canonical JSON structures; see CH13 for node/edge data model and CH14 for Sequence Detail Editor content.
- This chapter describes V1 behavior for iOS-only, portrait-only (global constraints owned by CH02).

1. Primary User Goal (V1)

Enable a striker to map decision paths quickly (“If opponent does X, respond with Y”), optionally add transition detail later, and keep the canvas usable even when flows become wide (many branches).

- Fast path: create a usable flow in under 2 minutes without entering deep details.
- Deep path: power users can add branches, merges, labels, and transition notes without UI overload.
- Predictable editing: every action has an obvious result and a clear way to undo or back out.

2. Canvas Fundamentals

2.1 Orientation modes (V1)

- A Flow has a per-flow “Layout Orientation” setting:
- Horizontal (default):
  - Flow grows left → right. Branches fan out vertically (up/down) as needed.
  - Best for many branches; supports side scrolling via pan/zoom.
- Vertical (optional toggle):
  - Flow grows top → bottom. Branches fan out horizontally (left/right) as needed.
  - Useful for users who think in vertical lists.
- Orientation toggle is accessible from the Builder header overflow menu: “Layout: Horizontal / Vertical”.
- Changing orientation re-runs auto-placement for nodes (does NOT change graph connections). Manual positions behavior is a placeholder.

2.2 Pan and zoom (V1)

- Pan/Zoom is required because flows can become wide with up to 10 branches per node and branches that branch again.
- Gestures:
  - Pinch with two fingers: zoom in/out around gesture center.
  - Two-finger drag: pan the canvas (moves the viewport).
  - One-finger drag on a node: drags that node.
  - One-finger drag on empty canvas: choose ONE behavior in build: (A) long-press (≈250ms) then drag to pan, or (B) dedicated pan mode toggle. Default recommendation: long-press-to-pan to avoid accidental movement while tapping.
- Zoom levels:
  - Default zoom: 100%.
  - Minimum zoom: placeholder (recommend 50%).
  - Maximum zoom: placeholder (recommend 200–250%).
  - A small zoom indicator (e.g., “125%”) appears transiently when zoom changes.
- Viewport controls:
  - Action “Fit” centers and fits the entire graph on screen.
  - Double-tap empty space: optional shortcut to zoom to 100% centered on tap point.
- Hit-target rules:
  - Primary controls meet iOS 44pt minimum touch target.
  - Small handles/labels may scale visually with zoom but keep invisible padding for reliable taps.

2.3 Node rendering basics (V1)

- Each Move Node renders as a rounded card with: move name (primary), optional small chips (stance/side/etc. if available), and an affordance for adding connections.
- Node states: default, selected, dragging, invalid/incomplete (for dangling branches), and disabled (if gated actions are unavailable).

- Canvas background: subtle grid or dotted field is optional; must not compete with nodes.

2.4 Edge rendering basics (V1)

- Two edge visual types:
- Linear continuation edge:
  - Solid line/arrow.
  - Represents “Next move” without opponent condition label.
- Branch edge:
  - Dashed line/arrow.
  - Has a label (opponent action/condition).
  - Represents decision point: “If X then Y”.
- Edges are tappable: tapping an edge opens the Sequence Detail entry point (CH14) or edge action sheet (see §5).

3. Builder Entry, Save Model, and Guest Gating

3.1 Where builder is launched from

- From Library: “New Flow” opens Builder with empty canvas.
- From Flow Detail: “Edit” reopens the existing flow in Builder.
- From Demo/Onboarding: demo flows can open in view mode; “Try editing” opens Builder but saving requires account.
- From Inbox imports: free users may preview; saving to library is gated by account + plan rules.

3.2 Saving rules inside builder (account required)

- Saving a flow (draft or final) requires an account. Guests may explore but cannot save locally or remotely (locked).
- Guest experience:
  - Persistent banner: “Guest Mode — changes won’t be saved.” + CTA “Create Account to Save”.
  - If guest taps Done/Save Draft: show account prompt with Apple/Google/Email options (CH07).
- Free plan cap awareness:
  - Free users can have 2 saved flows max (locked).
  - If user is at cap and attempts to save: show upsell screen (CH25) plus secondary option “Manage saved flows” (delete/upgrade).
- Autosave (logged-in):
  - Autosave after inactivity (placeholder interval) and on app background.
  - Status chip: “Saving…” $\rightarrow$ “Saved”.
  - Conflict handling and offline rules belong to CH28.

3.3 ‘Done’ vs ‘Save Draft’ semantics

- Header: Back, editable Title, and “Done”.
- Overflow: “Save Draft”, “Fit”, “Layout Orientation”, “Export (placeholder)”, “Delete Flow”.
- Recommended V1 semantics:
  - Done saves (if possible) and returns to Flow Detail.
  - Save Draft saves even if the flow is incomplete (dangling allowed).
  - If user cannot save (guest), both actions trigger the account prompt.

4. Core Interaction Model

4.1 Selection and focus

- Tap a node to select it. Selected node highlights and reveals contextual controls.
- Tap empty canvas to clear selection.
- Selection persists through pan/zoom.

4.2 Contextual actions for a selected node

When a node is selected, show a bottom sheet with actions:

- Primary actions:
  - Add Next (linear): solid edge to a new or existing move node.
  - Add Branch: dashed labeled edge with required label.
  - Replace Move: swap the move attached to this node.
- Secondary actions:
  - Edit Move details (CH11; respects canonical/custom rules).
  - Delete Node (see §6).
  - Manage Branches (reorder list) if node has branches.

4.3 Adding the first/root move

- Empty state shows CTA “Add First Move”.
- Selecting a move creates the root node at the origin position.
- Root move is replaceable.
- Deleting root returns to empty state.

4.4 Add Next (linear continuation)

- Select node → Add Next → Move Picker → choose move.
- System behavior:
  - Create new node (or reuse existing node if user selects a move already on canvas).
  - Connect with SOLID edge.
  - If source already has a solid “next” edge, prompt: Replace next? Replace / Cancel.

4.5 Add Branch (decision edges up to 10)

- Select node → Add Branch.
- Two-step flow:
  - Label: pick/enter opponent action/condition text.
  - Response: pick a move (existing or new).
- Hard cap:
  - Max 10 outgoing edges per node (locked).
  - Attempting 11th is blocked with message: “Max 10 branches from a move in V1.”
- Branch ordering:
  - Node maintains ordered branch list.

  - “Manage Branches” shows list with drag handles to reorder.
  - Order affects visual stacking + practice selection ordering.
- Branches can branch. Merges are allowed (multiple incoming paths).

5. Edge Interactions (Between Nodes)

5.1 Tapping edges

- Tap an edge to select it (highlight).
- Edge action sheet:
  - Edit Transition Details → opens CH14 screen/modal.
  - Edit Branch Label (branch edges only).
  - Delete Edge (removes connection; leaves nodes).
  - Jump to Target (centers viewport).

5.2 Branch label editing

- Branch labels are required and editable any time.
- Editing UI: inline edit with quick suggestions + free text.
- Validation: label cannot be empty.

5.3 Sequence Detail entry (CH14)

- Sequence details are optional; not required between moves (locked).
- Edge sheet always includes “Edit Transition Details”.
- Edge remains tappable at all zoom levels.

6. Deletion, Replacement, and Dangling Paths

6.1 Definitions

- Leaf node:
  - Node with no outgoing edges. Allowed.
- Dangling branch:
  - A branch that is conceptually incomplete (future V1/1.1).
- Orphan node:
  - Node with no incoming edges except root. Allowed but should be discoverable in cleanup tools later.

6.2 Dangling paths (V1 decision)

- Leaves are allowed in V1 and are the primary “incomplete” concept for now.
- Optionally, V1 may support branch-stub placeholders (dashed outline node) to represent “response TBD”. If not implemented, require choosing a response move during branch creation.

6.3 Node replacement (move swap)

- Replace Move keeps node position and connections; only the move reference changes.
- No automatic rule adjustments; only the label/chips update.

6.4 Deleting nodes

- If node has no edges: delete immediately with undo toast.
- If node has connections:
  - V1 minimum: confirm delete and remove node + its connected edges (Option A).
  - Option B (reconnect) is out of scope unless implemented carefully.

6.5 Undo (recommended)

- After destructive actions (delete node/edge, replace move), show an “Undo” toast for a short window.
- Full multi-step undo/redo is a stretch goal.

7. Move Picker in Builder

7.1 Entry points

- Add First Move
- Add Next
- Add Branch response
- Replace Move

7.2 Picker UI (V1)

- Sheet with search bar at top (autofocus).
- Sections:
  - Recent moves (chips).
  - Essentials (default set).
  - Browse categories (discipline/body part/defense/footwork).
- Selecting a move returns to builder and completes the originating action.
- Inline “Create new move” is available for logged-in users; guest restrictions are enforced by CH08/CH11.

7.3 Merge vs duplicate node when move already exists on canvas

- Default: selecting an already-present move reuses the existing node (creates merge).
- Offer “Duplicate node” as a secondary option if user wants separate instances for readability.

8. Layout Behavior at Scale

8.1 Auto-placement

- Horizontal:
  - Continuation node appears to the right with standard spacing.
  - Branch response nodes appear to the right and staggered up/down by branch order.
- Vertical:
  - Continuation node appears below.
  - Branch response nodes appear below and staggered left/right by branch order.
- Auto-placement avoids overlap; if overlap would occur, nudge by grid increments.

8.2 Manual repositioning

- Users can drag nodes anywhere.
- Edges re-route dynamically for readability.
- Manual reposition does not change branch order.

8.3 Readability aids

- Fit to screen action.
- Optional mini-map if needed (not required in V1).
- Optional path highlight (future).

8.4 Performance targets (placeholders)

- Aim for smooth interaction with medium-size graphs (placeholder: ~150 nodes).
- Degrade gracefully if needed: simplify edge rendering and reduce effects.

9. Error States and Microcopy (Builder-specific)

- Guest tries to save:
  - Title: “Create an account to save”
  - Body: “Guest Mode lets you explore, but your flows won’t be saved. Create an account to keep this flow.”
  - CTAs: “Sign up” (primary), “Keep exploring” (secondary).
- Free user at saved-flow cap (2):
  - Title: “You’re at your Free limit”
  - Body: “Free includes 2 saved flows. Upgrade to save more.”
  - CTAs: “Start Free Trial” (primary), “Manage saved flows” (secondary).
- Branch cap reached (10):
  - Toast: “Max 10 branches from a move in V1.”
- Empty branch label:
  - Inline: “Add a label (what the opponent does).”

10. Acceptance Tests (Given/When/Then)

- Given an empty canvas, when the user taps “Add First Move”, then the Move Picker opens and selecting a move creates a root node.
- Given a selected node, when the user taps “Add Next” and selects a move, then a solid edge connects to the chosen node.
- Given a selected node, when the user taps “Add Branch”, enters a label, and selects a move, then a dashed labeled edge is created.
- Given a node with 10 outgoing edges, when user attempts an 11th, then the action is blocked and a branch-cap message appears.
- Given a user selects an already-present downstream node, when they confirm reuse, then a merge is created (multiple incoming edges).
- Given a branch edge, when tapped, then edge actions include Edit Transition Details, Edit Label, Delete Edge.
- Given Guest Mode, when user taps Done/Save Draft, then signup prompt appears and nothing is saved.
- Given pinch gesture, when user pinches, then zoom changes and nodes/edges remain interactive.
- Given a connected node, when user deletes and confirms, then node and its edges are removed and an Undo toast appears.

11. Replit Build Prompt (Chapter-only)

```
You are implementing Handz V1 PRD Bundle (HZ-V1). Implement ONLY CH12 (Flow Builder: Interaction
Treat CH00 as authoritative for IDs and cross-references.

Build a React Native (iOS, portrait) Flow Builder with:
- Canvas pan/zoom (pinch to zoom, two-finger pan).
- Draggable Move Nodes; tappable Edges.
- Node selection opens bottom sheet with: Add Next, Add Branch, Replace Move, Delete Node.
- Add Next creates SOLID edge; Add Branch creates DASHED labeled edge (label required).
- Enforce max 10 outgoing edges per node; block 11th with toast.
- Allow merges: connect to an existing node (multiple incoming edges).
- Orientation toggle per flow: Horizontal default, optional Vertical (affects auto-placement).
- Guest gating: persistent banner “Guest Mode — changes won’t be saved.”; intercept Save/Done wi

Also implement:
- Edge action sheet: Edit Transition Details (stub route to CH14), Edit Label (branch), Delete E
- Fit-to-screen action.

Follow the CH12 acceptance tests.
```

12. Troubleshooting Notes (Chapter-only)

- Pan/zoom conflicts with node drag:
  - Use two-finger pan only, or require long-press on empty canvas to pan. Do not allow one-finger empty-canvas pan if it causes accidental moves.
- Edge label overlap:
  - Render labels as pills with background; allow truncation; show full label on tap.
- Graph rerenders too often:
  - Memoize nodes/edges; keep transforms in native layer when possible; throttle layout recalculation.
- Merge vs duplicate confusion:
  - When selecting an already-present move, show choice: Reuse (merge) vs Duplicate node.
- Guest saving confusion:
  - Banner must remain visible; prompts only on explicit save actions.