CH23 — Mastery & Gameplans

This chapter defines how Handz models "mastery" and "gameplans" for strike decision training, including goal types, planning logic, user trust controls, and how mastery integrates with Practice Mode logging.

Doc ID:	HZ-V1-CH23_Mastery_And_Gameplans_R1
Revision:	R1 (2026-01-02)
Status:	Draft
Depends on:	CH03 (Concepts & Glossary), CH08 (Entitlements), CH20 (Practice Setup), CH21 (Active Session), CH22 (Logging & History), CH24 (Maintenance), CH26 (Scientific Claims), CH27 (Notifications), CH30 (Safety/Abuse)
Related:	CH04 (Navigation), CH05 (Screen Inventory), CH15 (Library), CH16 (Flow Detail), CH17-19 (Sharing/Inbox/Import Conflicts), CH25 (Paywalls/Upsells), CH31 (Error States), CH33 (Analytics)
Supersedes:	None
Owned Decisions:	Mastery model & states; Gameplan object & composition rules; Goal types; Trust controls (downgrade/override); Planning outputs (recommended sessions/maintenance); UX requirements for low-friction tracking during training (gloves-on).
Open Questions / Placeholders:	Exact mastery thresholds; exact spaced-repetition schedule parameters; exact "science" citations & wording (owned by CH26); whether some mastery planning is Pro-only vs Free-limited (owned by CH08/CH25).

1. Purpose & Scope

Mastery & Gameplans exist to convert "flows" (decision trees) into actionable training plans that help users: (a) learn a set of responses until they feel automatic, (b) maintain those responses over time without overload, and (c) see concrete progress (what was memorized / what is on track) rather than only time spent.

This chapter defines logic and UX for:

- Selecting one or many paths (across one or many flows) and bundling them into a named Gameplan.
- Choosing a goal type (e.g., "Remember", "React faster", "Fight-camp readiness") that changes plan intensity and maintenance cadence.
- Generating a recommended practice schedule and maintenance schedule that the user can adjust.
- Tracking mastery state for each path and for the overall gameplan (including user-controlled downgrades/overrides).
- Showing value for Pro (practice + mastery tracking) without requiring gloves-off rating prompts during sets.

- Preventing maintenance overload (handoff to CH24 for scheduling and notifications).

Out of scope for CH23 (owned elsewhere):

- Exact copy and citations used for scientific claims (CH26).
- Exact paywall placements, trial mechanics, and upgrade flows (CH25).
- Practice timer mechanics, early-end logic, interruptions, and logging fields (CH20-CH22).
- App-wide navigation structure and page inventory (CH04-CH05).

2. Definitions & Core Objects

CH03 owns the global glossary. This chapter introduces concrete objects used by mastery planning and references CH03 terms.

2.1 Path

A Path is a specific ordered route through a flow: Move A → Sequence (optional) → Move B → ... until a chosen end node. Paths can cross branch nodes. A path may terminate at a leaf or intentionally stop mid-tree. Paths are the atomic unit of mastery and maintenance (users practice paths).

2.2 Gameplan

A Gameplan is a named bundle of one or more Paths selected by the user. Gameplans may include paths from multiple flows. Gameplans are designed to match user intent: an upcoming opponent, a curriculum block, or a personal "set of reactions" the user wants automatic.

Gameplan composition rules:

- A Gameplan can include: (a) multiple individual paths, (b) an entire flow, or (c) a saved filter-set that expands into paths (e.g., "all branches from Jab root").
- When a user selects an entire flow, Handz must automatically enumerate its paths in the background and present them as editable items inside the Gameplan.
- Users can rename any Gameplan at any time.
- A Gameplan can be edited later: add/remove paths, reorder drill priority, and change goal type.

2.3 Mastery

Mastery is an app-level model of how "locked in" a path is for the user, based on logged practice and time since last reinforcement. It is explicitly not a guarantee of fight performance. It is a planning and memory-maintenance proxy.

Mastery must satisfy two constraints:

- Low friction: It cannot require constant rating while the user is gloved or mid-round.
- Trust: Users must be able to correct/override mastery if the app feels wrong, without losing their entire library.

2.4 Mastery States (Path-Level)

Each Path has a MasteryState. The exact numerical thresholds are placeholders; the state machine and UI are defined here.

- Not Started — user has never practiced this path in Practice Mode (or has no qualifying logs).
- In Progress — user is actively training; plan has remaining sessions before target readiness.
- On Track — user is meeting the recommended cadence; forgetting risk is low.

- Needs Review — user is drifting; forgetting risk is rising; maintenance is recommended.
- Maintained — user has completed initial plan and is currently maintained at a defined cadence.
- Stale — user has not reinforced for a long period; should be treated like partially forgotten.
- User-Adjusted — a flag that indicates the user manually adjusted mastery up/down (see §6).

3. Goal Types & Intensity Profiles

When creating a Gameplan, users pick a Goal Type. Goal Type changes recommended volume, cadence, and what "success" means.

3.1 Goal Type menu (V1)

- Remember (Technique Recall) — primary objective is accurate recall of the sequence and decision path.
- React Faster (Decision Speed) — objective is fast selection of correct response; volume/cadence higher than Recall.
- Fight Camp (Readiness by date) — user sets a target date; plan schedules to reach readiness by that date.
- Coach Curriculum (Teach & retain) — similar to Recall, optimized for assigning to students; includes share/export hooks (owned by CH17-19).
- Just Drill (Simple) — minimal setup: user selects paths, picks timer settings, and practices; mastery tracking still applies but uses default intensity.

Note: Goal types are user-facing marketing language; exact in-app copy belongs to CH15/CH26. CH23 owns the behavior differences.

3.2 Intensity Profiles

Each Goal Type maps to an Intensity Profile. An Intensity Profile defines default parameters for: sessions per week, sets per path, target repetitions or time, and maintenance cadence. Exact numbers are placeholders and should be tuned later without changing the structure.

- Profile: Light — for hobbyists or low frequency training.
- Profile: Standard — default for most users.
- Profile: Aggressive — for fighters in camp or users seeking faster automation.
- Profile: Custom — user can override defaults (within safety constraints; see CH30).

Intensity override rules:

- User can always decrease intensity (less volume) without warnings.
- Increasing intensity above defaults may trigger soft warnings (e.g., "This may be hard to maintain") and should present a one-tap confirmation (CH30).

- Intensity overrides must be remembered at Gameplan level, and may be overridden per-path (advanced).

4. UX: Mastery & Gameplans Surfaces

CH04/CH05 own route maps and page inventory, but CH23 defines the required surfaces and their behavior.

4.1 Mastery Hub (entry surface)

Primary purpose: show concrete progress and next actions without overwhelming. Users should see: (a) what they are mastering, (b) what is due for review, (c) how many items are at risk, and (d) one-tap actions to practice.

Required modules on Mastery Hub:

- Top summary strip
- • Counts: Gameplans Active, Paths On Track, Paths Needing Review, Paths Stale.
  - Optional streak/cadence indicators are secondary; primary is accomplishment counts.
- Next Up card
- • Shows the single most important recommended action: e.g., "Review 3 paths from Gameplan: Southpaw Counter Plan".
  - Buttons: [Start Practice] (routes to CH20 with preselected paths), [View Gameplan], [Snooze] (if allowed, see CH24/CH27).
- Due for Review list
- • Sortable list of paths or grouped by gameplan.
  - Each item shows: Path name (derived), Source flow name, Status badge, Last practiced date, and a small "Why" tooltip (owned by CH26 phrasing).
- Active Gameplans list
- • Cards with: name, goal type, percent progress (derived), and quick actions: [Practice], [Edit], [View].

4.2 Gameplans List

Shows all gameplans. Must support search and basic sorting. V1 must allow renaming and deletion. Deletion must have clear choice: remove gameplan only vs remove gameplan and associated custom path configs (never delete flows).

Buttons/actions:

- [+ New Gameplan]
- Per item: [Practice], [View], [Edit], overflow (Rename, Duplicate, Delete)
- Empty state: explain what a gameplan is in one sentence + CTA to create (copy owned by CH15).

4.3 Gameplan Builder (Wizard)

The builder must support both beginner simplicity and advanced control. It must start by asking what the user wants to do first, then guide them (as previously agreed).

Wizard Step A — Choose scope

- Option 1: Select a whole flow (then auto-breakdown into paths).
- Option 2: Select specific paths from one or more flows.
- Option 3: Select paths across different flows (advanced) — allowed.

Wizard Step B — Name + goal type

- Name field (required, editable later).
- Goal Type picker (Recall / React Faster / Fight Camp / Coach Curriculum / Just Drill).
- If Fight Camp: ask for target date (calendar).

Wizard Step C — Intensity & constraints

- Select intensity profile: Light / Standard / Aggressive / Custom.
- Custom allows adjusting: sessions per week, sets per session, time per set, rest defaults. Actual session mechanics owned by CH20-CH21; this step only sets defaults.
- Confirm "I want as simple/complex as I want" philosophy: provide "Keep it simple" vs "Tune details" branching.

Wizard Step D — Review & create plan

- Show a preview of the first 7 days (or first 3 sessions) of the recommended schedule.
- Allow toggling the order of path priority (drag reorder).
- CTA: [Create Gameplan]. Secondary: [Back], [Save Draft].

5. Planning Logic: From Gameplan to Sessions

Planning logic must produce a schedule that feels actionable and not overwhelming. The system should: (a) generate recommended sessions, (b) map sessions to practice setups, and (c) define maintenance after initial mastery.

5.1 Planning Inputs

- Selected paths (each has a difficulty proxy and length).
- Goal type and intensity profile.
- User training frequency preference (self-reported).
- User available days/times (optional; full scheduling in CH24/CH27).
- Existing practice history for the same paths (CH22).
- User caps/entitlements (CH08) that may limit practice sessions (e.g., Free credits).

5.2 Path Difficulty Proxy (V1)

Because we cannot measure real performance in V1, difficulty is estimated using simple heuristics. These heuristics must be transparent ("estimated") and never presented as medical/scientific certainty.

- Path length: number of move nodes (and sequence nodes if present).
- Branch complexity: number of decision points in path.
- Move types included (e.g., spins/head kicks) — optional; placeholder until move taxonomy is finalized (CH09/CH10).
- User-tagged difficulty override (optional advanced): user can label a path as Easy/Medium/Hard; used for scheduling.

5.3 Recommended Plan Output

The plan output must generate: (a) an "Initial Mastery Phase" schedule and (b) a "Maintenance Phase" schedule. Both are editable and can be paused without punishment messaging.

- Initial Mastery Phase
- - A finite set of planned sessions. Each session contains an ordered list of selected paths with configured sets/time.
  - A target completion condition for each path (placeholder thresholds).
  - A projected completion date range (if user provides frequency).
- Maintenance Phase
- - After a path reaches Maintained, it enters a recurring review cadence.
  - Cadence adapts based on recency: if user misses reviews, the path moves to Needs Review or Stale.

- Maintenance schedule must cap weekly load (handoff to CH24 overload prevention).

5.4 Converting a Plan into Practice Setup

Each planned session must be representable as a pre-filled Practice Setup (CH20): paths selected, order fixed, and per-path configuration applied. When the user taps "Start Practice" from Mastery Hub or a Gameplan card, Handz routes to CH20 with a preloaded session preset.

Preload requirements:

- Selected paths list is prechecked.
- Order is pre-applied (user can reorder unless "Strict camp" mode is enabled; placeholder).
- Timer settings and assumed reps are prefilled from plan.
- Session is labeled with origin metadata: "From Gameplan " for logging (CH22).

6. Trust Controls & Manual Adjustments

To prevent users losing trust when the app feels unrealistic, Handz must allow mastery adjustments at multiple granularities.

6.1 Adjust Mastery (Path-level)

- Action: "Adjust mastery" available from Path Detail and from Gameplan Detail (per path).
- Options: Mark as Not Started / In Progress / Maintained / Needs Review / Stale (exact labels may vary but must map to states).
- Reason capture: optional quick-select reasons ("Haven't drilled in real sparring", "Injured", "Too easy", "App overestimated"). Must be one-tap and skippable.
- When adjusted, add a User-Adjusted flag and show subtle indicator in UI.

Guardrails:

- Never block the user from adjusting (no hard restriction).
- If user increases mastery upward, show a non-judgmental note: "Handz will use this to reduce reminders."
- If user downgrades, immediately surface a helpful action: "Want a quick review session now?"

6.2 Revert Model (Granularity)

Users may want to revert a single adjustment without reverting the entire library. Provide revert at three scopes.

- Scope A — Single path
- • Revert mastery state to system-estimated state (based on logs).
  - Revert custom difficulty tag.
  - Revert custom per-path plan config (sets/time).
- Scope B — Single gameplan
- • Revert all user adjustments within this gameplan (paths remain selected).
  - Recompute schedule using current goal type/intensity.
- Scope C — Global (rare)
- • Reset all mastery estimates (keeps logs). This should be behind a confirmation and likely placed in Settings (CH04/CH05).

7. Progress Metrics: Concrete Results

Users asked for results that feel concrete: not just minutes, but what was accomplished. V1 metrics must emphasize mastery progress and maintenance health.

7.1 Path Detail metrics

- Mastery state badge + explanation ("On Track" etc.).
- Sessions completed toward target (e.g., 4/10) — placeholder until thresholds defined.
- Last practiced timestamp + "days since".
- Next recommended review (date or "due now").
- Total completed sets and assumed reps (from CH22 logs).
- If user is in a gameplan: show which gameplans include this path.

7.2 Gameplan-level metrics

- Paths: total, on track, needs review, stale.
- Projected readiness (if Fight Camp) — computed from planned sessions remaining.
- Consistency indicator: "Sessions completed this week" (secondary).
- Maintenance load indicator: "X reviews scheduled" with overload warning if above cap (CH24).

7.3 No gloves-on rating requirement

Handz must not require users to rate performance mid-set. Instead, use logged completion signals from Practice Mode: set completed (timer ended or user tapped completed), interruptions, and early-end actions (CH21). Optional post-session reflection is allowed but must be skippable and low effort (CH22).

8. Entitlements & Funnel Considerations (No Contradictions)

CH08/CH25 own final gating. CH23 defines constraints to avoid contradictions between importing/sharing, saved flow caps, and mastery planning.

8.1 Known global locks to respect

- Practice is paywalled; Free receives 3 monthly practice credits usable only on saved flows (CH00 global lock).
- Free saved flows cap: 2. Free inbox cap: 10 items; Free can view but cannot practice inbox items (CH00 global lock).
- Share links are unlisted; imports should not be blocked in a way that kills the sharing funnel (CH17-19).

These locks come from CH00 decision log and must not be changed here. See CH00 §5 for the authoritative list.

8.2 Mastery visibility vs Mastery action

To preserve the funnel: Free users may be allowed to view mastery concepts and gameplan previews, but actions that generate value (Practice sessions, automated plans) may be Pro-gated or limited. Final rule belongs to CH08/CH25; CH23 provides a safe default suggestion.

- Suggested default (subject to CH08/CH25)
- - Free can: create Gameplans, add paths, view due-for-review list, and view a limited plan preview (e.g., first session).
  - Free cannot: start planned sessions unless they spend a practice credit and the session uses saved flows only.
  - Pro can: generate full plan schedules, start unlimited practice, and use maintenance notifications.

8.3 Imported content contradiction avoidance

If Free can import unlimited flows but cannot save beyond 2, the system must ensure coaches can still send value without bypassing Pro. This chapter enforces: (a) inbox view-only, (b) practice disallowed on inbox items for Free, (c) credits usable only on saved flows.

- If a Free user receives a flow via link/import: it appears in Inbox (cap 10).
- They can view it, but Practice CTA is disabled with explanation: "Save to your library to practice" + upsell if at saved-flow cap.
- To practice, they must move it into Library (requires account; and counts toward 2 saved flows cap).
- If at cap, they must delete an existing saved flow or upgrade (CH25) before saving the imported flow.

9. Data Model (Product-level, implementation-agnostic)

This section defines the minimal data objects required. Actual storage and API details live in CH29 (storage) and implementation docs.

9.1 Entities

Gameplan

Fields (suggested):

- id (UUID)
- owner_user_id
- name
- goal_type (enum)
- intensity_profile (enum)
- target_date (nullable)
- created_at, updated_at
- status (active/paused/archived)
- notes (optional)
- default_session_template_id (pointer to a generated practice preset)

GameplanItem (Path binding)

- id (UUID)
- gameplan_id
- source_flow_id
- path_id (stable path signature; see below)
- priority_order (integer)
- difficulty_override (nullable: easy/medium/hard)
- config_override (nullable: per-path sets/time overrides)
- created_at, updated_at

PathSignature (stable path identity)

A path must be referencable even if the canvas layout changes. Define a stable identity as an ordered list of node IDs (move nodes and optional sequence nodes) plus branch labels used. If a flow is edited such that the path no longer exists, the signature becomes "orphaned" and requires user resolution (see §10).

- path_node_ids: [nodeId1, nodeId2, ...]
- branch_conditions: ["if leans back", ...] (optional)

- version_tag: increments when the source flow changes shape

9.2 MasteryRecord

- id (UUID)
- owner_user_id
- path_signature
- mastery_state (enum)
- user_adjusted_flag (bool)
- system_estimated_state (enum)
- last_practiced_at
- sessions_completed_count
- sets_completed_count
- assumed_reps_total
- next_review_due_at (computed)
- maintenance_cadence_profile (computed/selected)
- created_at, updated_at

10. Edge Cases & Conflict Handling

CH19 handles import conflict resolution. CH23 defines mastery/gameplan-specific conflicts.

10.1 Orphaned paths after flow edits

- If a user edits a flow and removes nodes referenced by a Gameplan path, that path becomes orphaned.
- Orphaned paths must be clearly labeled in Gameplan Detail: "Needs update".
- Provide actions: [Fix Path] (reselect a new path), [Remove from Gameplan], [Keep as reference] (view-only).
- MasteryRecord for orphaned paths becomes frozen; it does not generate maintenance reminders until resolved.

10.2 Duplicated flows and path identity

- When duplicating a flow (CH16), duplicated nodes receive new IDs; paths in the duplicate are new identities.
- If user duplicates a flow specifically to create a variant of the same gameplan, offer: "Copy mastery progress into duplicate?" (optional; placeholder).
- Default: do not copy mastery automatically; copying can create false confidence. Provide manual tools instead.

10.3 Multi-gameplan inclusion

- A path can belong to multiple gameplans. Mastery is path-level (global) unless explicitly overridden.
- If two gameplans have different intensity configs for the same path, use the higher intensity only when practicing from that gameplan; mastery record aggregates all logs.
- UI must show when a path's state is influenced by multiple gameplans (e.g., "Included in 3 gameplans").

11. Maintenance & Notifications Hooks

CH24 defines overload prevention and scheduling; CH27 defines notification systems. CH23 defines what signals Mastery produces for those systems.

11.1 Signals emitted by Mastery

- DueReview(path_id, due_at, severity)
- OverloadRisk(weekly_reviews_count, recommended_cap)
- StaleRisk(path_id, days_since_last_practice)
- GameplanReadiness(gameplan_id, projected_ready_at, behind_schedule_flag)

11.2 Snooze behavior

- Snooze is always optional and never shaming.
- Snooze options: Later today / Tomorrow / This weekend / Custom (if CH24 allows).
- Snoozing a path updates its due date but does not change mastery state until practiced.

12. Acceptance Tests (Given/When/Then)

- Given a user has zero mastery records, when they open Mastery Hub, then the empty state explains mastery/gameplans and provides a single CTA to create a gameplan.
- Given a user creates a gameplan from a full flow, when the gameplan is created, then the flow is automatically expanded into a list of editable paths inside the gameplan.
- Given a user selects paths from multiple flows, when they save the gameplan, then the gameplan includes all selected paths with preserved priority order.
- Given a gameplan exists, when the user taps Practice from the gameplan card, then Practice Setup opens with those paths preselected and ordered.
- Given a path is practiced and logged, when the log is saved, then the associated mastery record updates last practiced time and increments sessions/sets totals.
- Given a user manually downgrades mastery for a path, when they confirm, then the path shows a User-Adjusted indicator and immediately offers a quick review CTA.
- Given a Free user views an inbox item, when they tap Practice, then the button is disabled and the UI explains that inbox items cannot be practiced on Free.
- Given a Free user is at saved-flow cap, when they attempt to save an imported flow to Library, then they must delete one saved flow or upgrade before saving.
- Given a flow edit removes nodes used by a gameplan path, when the gameplan detail loads, then the affected path is marked Orphaned and no maintenance reminders are generated for it until resolved.
- Given the user snoozes a due review, when the snooze completes, then the due date changes and the hub reflects the updated schedule without changing mastery state.

Checklist

- Gameplan create/edit/delete works without affecting flows.
- Paths can be selected across flows and re-ordered.
- Mastery state machine exists and is visible at path and gameplan levels.
- User adjustments are possible and reversible at multiple scopes.
- Preloaded Practice Setup integration works.
- Orphaned-path handling is implemented.
- Free vs Pro restrictions do not allow practicing inbox items; credits only apply to saved flows.

13. Replit Build Prompt (Chapter-specific)

```
You are implementing Handz V1 PRD Bundle chapter CH23 (Mastery & Gameplans) only.
Follow CH00 rules: no guessing, no summarizing, and use cross-references instead of
redefining other chapters.

Goal:
Implement the product surfaces and data model for Mastery & Gameplans:
- Gameplans list, gameplan builder wizard, gameplan detail, path detail mastery UI, and
mastery hub summary.
- MasteryRecord update logic that consumes practice logs from CH22.
- Route to Practice Setup (CH20) with a prefilled preset when user taps Practice from
mastery/gameplan.

Scope limits:
- Do not implement notifications scheduling (CH27) or overload prevention engine (CH24)
beyond placeholders.
- Do not implement paywalls or purchase flows (CH25) beyond calling provided gating
helpers (isPro, creditsRemaining).
- Do not implement import conflict resolution beyond receiving 'resolved path/move IDs'
from CH19.

Required steps:
1) Define data models/interfaces:
- Gameplan, GameplanItem, PathSignature, MasteryRecord (as specified in CH23 §9).
2) Create storage layer (local + remote stubs):
- CRUD for gameplans and items.
- Read/update mastery records keyed by path signature.
3) Build screens/components:
- MasteryHubScreen: summary strip, Next Up card, Due for Review list, Active Gameplans
list.
- GameplansListScreen: search, sort, create CTA.
- GameplanBuilderWizard: Step A scope, Step B name+goal, Step C intensity, Step D review.
- GameplanDetailScreen: list of paths with statuses, edit actions, orphaned flags.
- PathDetailScreen: mastery state, sessions toward target, last practiced, adjust mastery,
revert options.
4) Integrate with Practice Setup (CH20):
- When user taps Practice, generate a PracticePreset object: selected paths + order +
default timers from intensity profile.
- Navigate to PracticeSetup with preset.
5) Integrate with Practice Logging (CH22):
- On practice log save, update MasteryRecord: lastPracticedAt, counts, compute
nextReviewDueAt placeholder.
6) Implement gating:
- If content is in inbox and user is Free, disable practice CTA with explanation.
- If user is Free and at saved-flow cap, prevent saving imported flow; route to manage
flows or upgrade.
7) Add unit tests / integration tests for key flows (see CH23 §12).
8) Add a 'PRD Assumptions' comment block if anything is unclear, and stop implementation
of that part until resolved.

Deliverables:
- Working UI for mastery/gameplans.
- Clean internal state management.
- Test coverage for core flows.
```

14. Troubleshooting Notes (Chapter-specific)

- Mastery states feel wrong to users
- - Ensure Adjust Mastery is easy to find (Path Detail + Gameplan Detail).
  - Ensure user-adjusted indicator is visible but not shaming.
  - Confirm logs are being attributed to the correct PathSignature; mismatches make mastery appear broken.
- Gameplan paths disappear after flow edits
- - Validate orphan detection: if node IDs changed, mark orphaned instead of silently dropping.
  - Provide Fix Path workflow and freeze reminders until resolved.
- Practice launches with wrong paths or order
- - Confirm practice preset generation uses gameplan priority_order.
  - Confirm CH20 expects the same path identifier format; if not, map through a PathSignature adapter.
- Free users can practice imported/inbox items
- - Confirm you are checking source=Inbox and plan state=Free at the Practice CTA layer.
  - Credits must be usable only on saved flows; enforce at session start, not only at UI.
- Performance issues with large gameplans
- - Virtualize lists, memoize path rows, and compute derived counts off main thread if needed.
  - Avoid recomputing entire plan on every render; cache plan outputs and only recompute on edits.