# **Pre-Coding Essentials (Component: README.md, Version/FormulaID: VM-ENGINE v0) — 9/89**

## **1) Goal & Success**

Goal: One-page (expandable) entry that orients a new developer/user to the engine: what it does, how to run it **offline**, how determinism is guaranteed, and where specs/tests live.

Success: A first-time clone can: (1) build, (2) run a tiny Annex-B fixture, (3) verify deterministic IDs, (4) find Docs 1–7 & Annex A/B quickly.

## **2) Scope**

In scope: Purpose, quickstart, repo map, offline/determinism policy, canonical artifacts, fixtures/tests, troubleshooting.

Out of scope: Legal text (in LICENSE), contribution/security (their own files), deep design (Docs 1–7 host that).

## **3) Inputs → Outputs**

Inputs: Workspace crates, fixtures under fixtures/annex\_b/\*, schemas under schemas/\*, CLI vm\_cli.

Outputs: A single authoritative README.md with copy-pastable commands that work cross-platform (Windows/macOS/Linux).

## **4) Entities/Tables (minimal)**

## **5) Variables (content toggles)**

## **6) Functions (signatures only)**

(Markdown doc; no code functions.)

## **7) Algorithm Outline (content structure)**

**Project summary (3–4 sentences).** What the engine is, what problems it solves (tabulation, allocation, gates/frontier), and that it’s deterministic & offline.

**Determinism & offline guarantees.** Bullet points: canonical JSON (UTF-8/LF/sorted keys), seeded RNG (ChaCha20) for ties, no telemetry, no network I/O.

**Quickstart.**

Clone → toolchain → build:

Bash:  
  
 bash  
CopyEdit  
rustup show && cargo build --locked -p vm\_cli

PowerShell:  
  
 powershell  
CopyEdit  
rustup show; cargo build --locked -p vm\_cli

Run tiny fixture (Annex-B Part 0/1):  
  
 bash  
CopyEdit  
vm\_cli run --manifest fixtures/annex\_b/part\_0/manifest.json --out artifacts/run

Determinism smoke (same seed twice):  
  
 bash  
CopyEdit  
vm\_cli run --manifest fixtures/annex\_b/part\_0/manifest.json --rng-seed 0000...0001 --out artifacts/run1

vm\_cli run --manifest fixtures/annex\_b/part\_0/manifest.json --rng-seed 0000...0001 --out artifacts/run2

diff artifacts/run1/result.json artifacts/run2/result.json

**Repository map (short).**

schemas/ (JSON Schemas), fixtures/annex\_b/ (canonical tests), crates/ (vm\_core, vm\_algo, vm\_pipeline, vm\_report, vm\_cli, vm\_app), tests/, artifacts/ (outputs), dist/.

**Specs & policy links.** Point to Docs **1–7** and **Annex A/B** in this repo; state that code behavior is subordinate to those docs if conflicts arise.

**How to run tests.** cargo test --locked, then make fixtures (or listed CLI loop) to compare winners/labels with expected.

**Building reports.** Mention vm\_report JSON/HTML outputs, one-decimal presentation; note no network dependencies (fonts/tiles bundled where relevant).

**Troubleshooting.** Common pitfalls: CRLF on Windows, missing vendored deps, RNG seed formatting, WTA requires magnitude=1.

**License & security.** Pointers to LICENSE and SECURITY.md. No bug bounty, no telemetry.

## **8) State Flow**

Reader follows quickstart → produces result.json/run\_record.json → confirms identical IDs across reruns with same seed → proceeds to deeper docs/tests.

## **9) Determinism & Numeric Rules (to state explicitly)**

Canonical serialization: UTF-8, LF, sorted JSON keys; timestamps in UTC.

Integer/rational math only; **round half to even** at defined comparison points; percentages rounded once in reports.

Tie policy: deterministic by order\_index or seeded RNG when configured; seed recorded in RunRecord.

## **10) Edge Cases & Failure Policy**

Windows shell differences: provide PowerShell equivalents; advise git config core.autocrlf false.

First build without vendor/: allow cargo fetch (temporarily disable offline), then restore offline mode.

Manifest must provide exactly one of ballots or precomputed tally; missing/extra inputs → validation error.

## **11) Test Checklist (must pass)**

Copy/paste quickstart builds vm\_cli on Win/macOS/Linux.

Fixture run produces a Result with expected winners/label for VM-TST-001.

Double run with identical seed yields identical RES:/RUN: IDs.

All doc links resolve within repo; no external network required to read spec/fixtures.

**Authoring note:** keep README.md ≤ ~300 lines; move details (full CLI options, file formats, extended troubleshooting) into /docs/ subpages to preserve a tight entry point.