Grassland Iron Tide

A five-mission browser RTS chapter keeping selection grammar, battlefield sightlines, and defeat recovery strictly unified.

Complex Systems · Real-Time Strategy

2026 · Chapter One playable

Lewis Dong · 24kplo@gmail.com

---

01 · Core Design Tradeoff

Enforce one command grammar across five missions instead of per-level special controls

Human tension: Real-time strategy players manage multiple squads under a ticking clock; when designers patch in custom buttons or override selection rules for later levels, escalating challenge turns into interface betrayal.

Alternatives rejected:

- **Per-mission custom shortcuts and bespoke UI buttons** — Forces players to unlearn muscle memory mid-chapter and obscures whether a mistake came from strategy or misread controls.
- **Commercial-RTS command density (dozens of micro-abilities and stances)** — Overloads new browser players before core tactical timing and terrain reading ever register.

Chosen tradeoff: Lock click, box-select, Shift add/remove, Ctrl grouping, and right-click commands into one immutable grammar—missions may restrict which commands are available, never change what an action means.

02 · Key Interaction Rule

Pair a fixed battlefield camera with one unified failure-and-checkpoint save chain

Fix camera zoom so the battlefield claims the largest screen area while objectives, minimap, and command grid sit cleanly on edges; pause, defeat reason, checkpoint, and unit restoration share one 20 Hz simulation save chain.

03 · Key Interaction Rule

Load Phaser atlases per mission via a data-driven manifest instead of preloading all five chapters

Express missions 1–5 as typed manifest definitions consumed by a shared runtime scene loader: only the active mission's assets mount into memory while retries reset state purely via in-memory checkpoints.

Role & Scope

Solo · Systems design, specs & QA sign-off

- Immutable 5-mission command model
- TypeScript · Phaser 4 · 20 Hz Sim
- 430/430 tests · Dual-res playthrough

Honest Scope & Limits

No structured usability testing with outside players.

Run sessions with outside players and record where they get stuck in the first two missions.

Scan for Live Prototype

https://lewisdong.vercel.app/en/projects/grassland#interactive-demo

Walkie

An M5StickS3 wrist voice remote for Claude Code—keeping every high-risk tool approval locked to deliberate physical key presses.

Physical AI · Tactile Control

2026 · Oct

Lewis Dong · 24kplo@gmail.com

---

01 · Core Design Tradeoff

Bind tool approvals to physical watch buttons rather than spoken 'yes'

Human tension: Stepping away from the desk leaves an autonomous coding agent stalled on a single permission prompt, yet speaking commands aloud in shared spaces creates privacy discomfort and makes accidental approval feel effortless.

Alternatives rejected:

- **Hands-free voice confirmation ('Yes, run it')** — Speech-to-text cannot distinguish deliberate user intent from background conversation or media audio, offering zero tactile boundary before a destructive shell command executes.
- **Mirror the full terminal session on a phone screen** — Pulling out a phone re-exposes notifications and drags a two-second decision back into full-screen desktop context switching.

Chosen tradeoff: Asymmetric physical key grammar on M5StickS3: hold the Front Key to record press-to-talk audio, press Front Key to approve, press Side Key to deny (with an optional spoken reason), and require a 30-second physical Front Key press to pair any new PC.

02 · Key Interaction Rule

Compress context into spoken summaries and explicit offline diagnostics

The Node 24 bridge distills replies into short spoken summaries, USB always preempts the AES-256-GCM Fly.io relay when plugged in, and OFFLINE screens distinguish no-Wi-Fi, relay-unreachable, and PC-bridge-offline explicitly.

03 · Key Interaction Rule

Enforce sibling-partition flash guards and button-confirmed key pairing across shared hardware

firmware/guard.py blocks -t erase and -t uploadfs pre-flash, pairing locks an AES-256-GCM shared key to pair.key only after a 30-second physical Front Key confirmation, and Fly.io routes opaque binary envelopes without decrypting them.

Role & Scope

Solo · Product direction, interaction rules & on-device checks

Dual-boot S3 key-locked approvals

ESP32 C++ · AES-256-GCM · Node 24

3 manual hardware/relay gates

Honest Scope & Limits

Single developer (Lewis) has used it; no external user study has been run.

Have 2–3 Claude Code users test an approval-heavy session away from their desk.

Scan for Live Prototype

https://lewisdong.vercel.app/en/projects/walkie#interactive-demo

Resume Autofill

Chrome and Android resume autofill that visibly separates certain matches from guesswork—and never submits on your behalf.

Trustworthy AI · Local Privacy Tool

2026 · used on real applications

Lewis Dong · 24kplo@gmail.com

---

01 · Core Design Tradeoff

Split certainty into three visible tiers and leave ambiguous answers blank

Human tension: Re-typing one resume across Workday, Beisen, and Moka is exhausting, yet AI autofillers that guess visa status, salary expectations, or motivation essays to look '100% automatic' shift catastrophic error risk onto the applicant.

Alternatives rejected:

- **Cloud LLM auto-generation for every blank and open-ended prompt** — Produces plausible hallucinations on high-stakes compliance fields and requires sending private resume history to external servers.
- **Color-only highlight for uncertain fields** — Fails color-blind accessibility and leaves users unsure whether a field was skipped intentionally or broken.

Chosen tradeoff: Deterministic 1:1 local matches fill immediately (solid), ambiguous dropdowns pause for review (dashed), open prompts stay blank (3px solid), and every tier pairs stroke style, word badge, and explicit side-panel reason.

02 · Key Interaction Rule

Hard-lock final Submit to human action and keep resumes strictly on-device

Extension logic advances Next across pages and stops dead before Submit (stated explicitly in the side panel), while Chrome CSP blocks network calls and Android encrypts resumes in the device Keystore (verified by a unit test rejecting network imports).

03 · Key Interaction Rule

Match by UI component library first and pair with a network-free Android Resume Keyboard

Adapter matching abstracts controls by underlying library (Ant Design, Element Plus, react-select, ARIA) plus ATS presets, while on Android a Keystore-encrypted custom IME lets applicants tap-insert local resume fields cleanly—verified by unit tests rejecting any network import.

Role & Scope

Solo · Full-stack architecture & rules

3-tier certainty & Submit lock

Chrome MV3 · Capacitor Android (Java) · Zero net

6/6 visual checks · Used on Lewis's own applications

Honest Scope & Limits

No one but Lewis is on record as using it, and there has been no user testing.

Watch three or four graduates use it on one real application each.

Scan for Live Prototype

https://lewisdong.vercel.app/en/projects/autofill#interactive-demo