

Playbook

How a non-developer directs AI from idea to launch — the method I actually used to ship 4 products solo

This isn't a course — it's a **field record**. It's the procedure I actually followed shipping THE HOUSE, MECHOOSE, and Momento and finishing Money Navigation at a hackathon. The public version carries only skeletons and checklists; detailed conditions are shared in interviews/collaboration.

1. Before you start

- Did you define the problem in one sentence (who, what pain)?
- Did you write down the folder/file structure and "it must behave like this" bar?
- Did you confirm stack/platform policy from official docs, not by guessing?

2. Set up the rules doc

- Way of working / don'ts / verification bar in one file
- Every session start: "read first and tell me the current stage" to sync
- Core rules, e.g.: no fakes (hardcoding) · test at three points · don't decide from what you alone know

3. Design first: screens and data

- Pull all design/UI and the data structure first, then start coding (doing both at once gets tangled)

4. The instruct → check → correct loop

- Instruct in small pieces — one screen, one feature
- Don't trust "it's done" — verify on a phone/simulator yourself
- When it diverges, point at the cause and re-instruct or demand a rewrite

5. Verification checklist

- Dry-run before build / static analysis, simulator, screenshot compare after build / full demo re-check per major part
- Re-run plausible successes on **changed inputs** to confirm
- Secrets only in env/secret stores; dummy data in screen captures

6. Patterns AI often gets wrong, and how to handle them

- Demo hardcoding → re-run on different inputs
- Repeated errors in generated code → keep design/data model, rewrite the code
- Version incompatibility (DI, etc.) → don't force it, weigh alternatives first
- Self-report vs reality mismatch → re-verify with actual code/artifacts

7. When stuck

- Decisively park paths outside your permission (org policy) or that are unstable
- Pivot to what works (free, on-device, a different channel) and get at least one thing working

8. Running multiple sessions (a one-person AI team)

- With several sessions, appoint one as PM to centralize order and reporting
- Prevent conflicts with file-ownership, commit-boundary, and build-lock rules
- Document state so context survives even when it changes

9. Launch checklist (per platform)

- Common: separate secrets, check for personal-data exposure, verify every link
- iOS: TestFlight / App Store Connect tax & bank, simulator captures
- Android: back up the signing keystore, internal testing track
- Web/edge: security headers, deploy automation script
- Game (Unity): split asset delivery, build automation, per-language store metadata

10. Case index

- THE HOUSE (game) · MECHOOSE (commerce) · Momento (two natives) · AttenNFC (design & security) · Money Navigation (hackathon, multi-session)
- See each project's case study for real instruction excerpts and troubleshooting.