

AI DIRECTION WORKSHEET

Bounded AI Change Brief

Define a safe implementation boundary before an AI coding partner changes a game project.

Use this brief before implementation. It turns an idea into a supervised change contract: verified context, protected surfaces, acceptance evidence, stop conditions, and a human decision owner.

USE THIS RESOURCE WHEN

- An AI agent will edit an existing repository.
- A requested feature touches more than one owner or system.
- A plausible implementation could pass tests while violating the player-facing contract.



Free Academy resource

English • US Letter edition • Fillable PDF • v1.0

[Resource page](#)

How to use it

Use this brief before implementation. It turns an idea into a supervised change contract: verified context, protected surfaces, acceptance evidence, stop conditions, and a human decision owner.

HOW TO USE IT

- 1 Write observable facts, not assumptions, in the verified-current-behavior fields.
- 2 Name what must not change before describing the implementation.
- 3 Give the AI evidence requirements and stop conditions before granting edit authority.
- 4 A human reviews the diff and evidence before accepting the change.

EXPLORE THE ACADEMY

[Martinez AI Studios](#)

[Academy](#)

[170-lesson course](#)

[Resource page](#)

USE AND SHARING

Free for personal and educational use. Share the original Martinez AI Studios URL. No resale, commercial redistribution, or re-upload.

1. Define the outcome

Goal / player effect

What must become observably different for the player or operator?

Verified current behavior

Reproduction steps, observed result, environment, and evidence location.

Source of truth

Owner, file, schema, test, specification, or runtime observation.

2. Bound the change

In-scope files / systems

Only the smallest surfaces required to produce the outcome.

Protected surfaces

Neighboring behavior, contracts, data, APIs, saves, or presentation that must remain unchanged.

Non-goals

Attractive cleanup, refactors, migrations, or features explicitly excluded.

Owner and authority boundary

Who owns each write and which system is allowed to make it?

3. Specify the safe implementation

Required change

The smallest behavior or data change that satisfies the goal.

Forbidden mutations

Writes, dependencies, shortcuts, generated assets, or side effects the AI must refuse.

Invariants to preserve

Rules that must stay true before, during, and after the change.

Acceptance criteria

Observable outcomes with a clear pass or fail result.

4. Evidence, recovery, and decision

Boundary / regression checks

Checks that demonstrate protected behavior still works.

Required output

Plan, focused diff, tests, explanation, screenshots, logs, or artifacts.

Evidence required

Commands, results, runtime observations, and file locations the reviewer must receive.

Known unknowns

What remains uncertain and how it will be resolved.

4. Evidence, recovery, and decision — continued

Ask before acting / stop if

Stop when scope expands, authority is unclear, baseline fails, or recovery is unavailable.

Checkpoint / restore reference

Commit, branch, backup, fixture, or exact restoration procedure.

Final human decision owner

Name, role, decision, date, and any conditions.