Representation and Feedback Doctrine

Representation and Feedback Doctrine

Purpose

This page fixes the game's presentation stance: what game objects are allowed to look like, what the interface is allowed to tell the player, and how harm is modeled and communicated. It is a governing principle, not a tuning sheet. Tick mechanics live in the Combat System chapter; this governs how they are *shown*.

Three rules, then their known costs.

Rule 1: representation must be diegetic

A game object is represented by something that exists in the game's fiction. A sword is a sword. A spell is cast, not played. A character does not hold a hand of cards.

The test that separates a legitimate abstraction from an import: **abstractions that evolved inside the medium are fine; objects borrowed from another medium are a tell.** A health bar, a minimap, a pause menu and a level number are all non-diegetic, and all of them grew up inside computer games. A card is different in kind. It is an artifact of tabletop, and importing it means the fiction has been subordinated to a convention from a different medium. The signal the player receives is "board game in a video game suit."

The card test: strip the cards off a design and present the same multiset of options in any other form, keeping the draw rules identical. If the game plays the same, the cards were skin and should be replaced with the diegetic object.

Consequence: **no literal card objects.** Where a design reaches for cards, use the diegetic object instead (an ability, a weapon, a stance, a readied slot) and represent its state in the world.

Hidden information is not a defence for cards. The *bands solved diegetic hidden information decades ago: items carry a flavor appearance and stay unidentified until handled, used, or read with an identify spell; monster knowledge accumulates through fighting. Same information asymmetry, no cardboard.

Reference points where the same mechanic is expressed as an object in the world: Into the Breach (abilities always available, constrained by position and turn order), FTL (weapon charge times shown on the ship), Civ production queues, Darkest Dungeon (the constraint is position and stress, not a hand).

Rule 2: numberless is the target state, numbers are a dev aid

End state: the shipped player-facing UI carries no numeric readouts for health, damage or state. Feedback is textual and qualitative.

During development: numbers are permitted in the display behind a dev flag, for engineering and tuning. The flag is a debug affordance, not a crutch: numberless is the default and the flag is opt-in, never the reverse.

The reason for the end state: a number on screen substitutes for a mental model. The player reads the number instead of learning the system. Removing it forces the design to teach through description.

Build the text layer first, overlay the numbers on top

The ordering matters more than the flag. If the text vocabulary is deferred until "final product", it gets retrofitted under schedule pressure and then quietly dropped, and the game will have been tuned for a readout it no longer has.

So: **the text channel is primary from the first playable build and is exercised in every session. Numbers are an overlay drawn on top of a system that already speaks.** Every state change must emit its qualitative line before the numeric flag can be considered finished, and a build with the flag off must never be the first time anyone reads the wording.

The legibility test runs with the flag OFF

The two playtest questions at the bottom of this page only measure the text layer if the numbers are hidden. Run them with the dev flag off even mid-development, then flip the flag on afterwards to check the player's answer against the real numbers. Numbers on for engineering, numbers off for legibility testing.

Numbers stay a calibration channel

Even at the end state, numbers do real work off-screen: they let a designer verify that the model matches the game's actual state. That channel must be *replaced*, not merely deleted, in the player-facing UI. **Every qualitative phrase must sit on a consistent, learnable, ordinal scale, globally.** "A little", "a lot" and "much" must mean the same magnitudes everywhere, and the same phrasing must mean the same thing whether the source was a sword or a fireball.

Text that is decorative rather than ordinal fails this. "The goblin looks annoyed" conveys nothing the player can calibrate against, so nothing is learned. If feedback cannot be ranked, it is flavour, not information.

Precedents: Angband's textual feedback (monotonic phrasing the player learns), Brogue (deliberately playable with no numeric readouts, bars and description only), Dead Space (health on the character's spine, ammo on the weapon) for the diegetic HUD case.

Rule 3: fixed body, scaling avoidance

Health does not scale with level.

- Bodily capacity is essentially fixed, varying slightly between individuals and over a life, not by experience.
- What experience buys is **a buffer of avoidance**: honed reflex, knowing when to duck, tucking and rolling, rolling with the punch, deflecting with a blade.
- The buffer depletes through a fight, accumulating nicks and scratches in proportion to the character's experience.

- When the buffer is exhausted, real damage lands, and real damage has **real effects**: on attack, on movement, on breathing.

Progression therefore shows up as *you get harder to hit*, not *you have a bigger pool*. Survivability still grows. It grows as competence.

Precedents: Into the Odd (hit protection depletes first, then damage goes to attributes), Blades in the Dark (harm levels that reduce effect), Sekiro (posture and vitality as separate tracks).

Failure mode: damage must degrade verbs, not subtract quantity

If the buffer is a hit point pool with a new name, the design has rebuilt HP and bolted text onto it. The model only pays off when real damage changes **capability**:

- broken rib: cannot draw a bow, breathing is audible, stealth suffers
- cut tendon in the leg: slower movement, cannot sprint
- blood in the eye: attacks miss more

These are categorical state changes. A wound system is a set of disabled or degraded verbs. A hit point pool is a subtraction.

Test: if resting restores a character completely, it is HP regardless of what it is called. Wounds have to change how the player plays.

Accepted cost: variance

Fixed body plus an avoidance buffer means an experienced character takes fewer hits, but each real hit is a larger fraction of their life. The distribution widens: most fights are comfortable, then one unlucky exchange nearly kills a veteran.

This is deliberate. It buys tension and costs computability. In a numbers game a careful player can prove they are safe; here they cannot compute survival at all. That is the trade, and it defines the audience. The min-max players lost here are lost on purpose, not by accident.

Feedback as rising action

A depletion buffer lets combat carry a narrative arc instead of a subtraction: fresh, working, winded, guard failing, exposed, wounded. Each stage is a beat with its own vocabulary. This is the main creative payoff of Rule 3, and it is only available if the feedback is ordinal (Rule 2).

Production cost: text at volume. Every ability, monster, wound and branch needs its own line, with enough variation to avoid becoming wallpaper. This is the main expense of the whole stance, and it is writing, not code.

Validation: two playtest questions

Neither requires a UI mockup.

1. After a fight, stop the player: **"How hurt are you, and what is wrong with you?"** Compare their answer to actual state. If they cannot answer, the feedback vocabulary failed.
2. **"Would you fight another one of those right now?"** Compare their answer to the real odds. If willingness does not track actual risk, the representation is lying to the player.

The two catch different failures. (1) catches a vocabulary that cannot describe the state. (2) catches a vocabulary that describes it inaccurately.

Where this collides with existing decisions

- **Open Question 6** (UI/information design) is now governed by this page. The knowledge-gated ASCII question stands, but any answer must satisfy Rules 1 and 2, which means a numberless player-facing UI in the target state and numbers available behind a dev flag during development.
- **Open Question 13** previously allowed the 0-1 roll to be shown as 1-100 percentages "for readable tuning". Rule 2 disallows numeric display in the **player-facing UI**. Percentages remain legitimate in dev tooling and tuning logs, where the audience is the designer and the seeded replay log rather than the player.
- Downstream consequences: monster knowledge display, wound vocabulary, level-up signalling, and how ASCII glyphs convey phase state all become children of this page.

Related

- Combat System > Tick Engine and PER Cycle
- Combat System > Cancellation and Readability (readability is diegetic signalling, the same principle applied to intent)
- Planning > Open Questions (6, 13)
- Overview (pillars)

Revision #2
Created 2026-10-21 03:21:37 UTC by Rowan Flora
Updated 2026-10-21 03:25:02 UTC by Rowan Flora