Combat System

Tick engine, PER cycle, resolution, reactions, cancellation, movement

- Tick Engine and PER Cycle
- Resolution, Initiative, and Reactions
- Cancellation and Readability
- Movement and Terrain

Tick Engine and PER Cycle

Tick Engine and PER Cycle

The combat clock is a **global discrete tick**, not real time. Player-paced: the world advances only when the player commits an action or explicitly waits. "Whenever it's time for the user to enter something, everything stops."

Granularity

- Ticks are deliberately tiny: roughly **10 ticks per foot** on even terrain. Integer tick durations everywhere.
- Fine granularity minimizes same-tick ties between actors (though ties still happen - see Resolution).

The PER Cycle

Every action - attacks, spells, movement, cancels, reactions - is modeled identically:

Phase	Meaning
Prep	Wind-up: gathering reagents, mental focus, shifting grip. The action has not started. This is the **readable, interruptible window** - only intelligent/experienced observers can read prep as an intention.
Exec	The action is *active*: casting, swinging. You can't do anything else. May include zone windows (e.g. a whirlwind that slices anyone entering range). The action can be mitigated here, but it cannot be cancelled.
Recover	The action is *done*; the actor is winding down - regaining footing, recovering balance. Nothing can affect the completed action at this point, but enemies can **take advantage of the window** while the actor is not fully recovered.

- **Zero phases allowed**: a fighter feat can zero prep/recover on light attacks to enable combos. Readability and anti-spam live in the prep window, not recover.
- **One action state machine**: every entity in the sim runs the same Action state machine - PREP, then EXEC, then RECOVER. Attacks, spells, movement, cancels, and reactions are all instances of one Action type; they differ only in parameters (phase durations, exec windows, completion effects), never in code path. A cancel or a reaction is itself an action with its own PER cycle, so the same rules compose.

Movement is an action

Movement uses the same PER model; terrain is paid at the borders (see Movement and Terrain).

Resolution, Initiative, and Reactions

Resolution, Initiative, and Reactions

Tick resolution

Each tick runs one loop over every actor, in initiative order:

1. **Actor action processing.** Does the actor have an action in progress?
   - **No** - select the next action (monsters: from the profile's ranked triggers; player: input - the game stops until they commit) and process its first tick. A zero-prep action flows straight through to the next phase.
   - **Yes** - process the current tick of the PER cycle.
2. **Reaction collection.** After the actor's processing, loop the other actors: did this tick trigger a reaction? If yes, add it to the queue - an actor may collect multiple reactions in the same tick. The reactor-side gate applies at collection: an actor in its own exec adds nothing (it is committed, not observing).
3. **Reaction pass.** Loop the reaction queue in initiative order:
   - An actor with several queued reactions evaluates them and uses the best one for the current circumstances; the rest are discarded. Monsters rank by profile; the player is offered the choice.
   - If the actor executes the reaction: pause or cancel their current action (if any), per that action's interruptibility, then process the first tick of the reaction's own PER cycle. A short reaction may complete its exec within this tick and spend subsequent ticks in recover.

Movement is not a special phase - it is an action like any other and runs through the same actor loop. Within a movement action, "leaving" reaction triggers queue before "entering" triggers; initiative breaks ties within each kind.

Deterministic, seeded, logged. This is what makes fights replayable for debugging and tuning.

Initiative

- Rolled at combat start (Perception or Stealth style, as in Pathfinder), **fixed for the whole fight**.
- Used as **priority for same-tick, world-affecting resolutions only** (actor order, reactions, movement events) - not for internal phase transitions.
- Tiebreakers: dex, then level, then PCs-over-monsters, then a **coin flip pre-rolled once per combat** (so identical actors stay consistent all fight).
- **Initiative as a speed stat**: a small random jitter is added to PER cycles; initiative reduces prep and recover time (never exec - exec durations stay consistent so active windows remain readable). High initiative therefore produces more actions over a fight. The jitter is seeded per combat so replays reproduce.

Reactions

Reactions are short actions taken out of sequence, triggered by observing another actor's phase.

Can I react at all? - gated by your own phase:

My phase	React?	Cost
Prep	Yes	Pauses or cancels my prep, per the action's interruptibility
Exec	No	Committed - this is the point
Recover	Yes	The reaction's own PER handles the cost

What can my reaction do? - gated by the observed target's phase:

Observed phase	What a reaction can do
Prep	Can **cancel** the target's action (interrupt the windup). Knowledge-gated: only actors who can read prep as prep can target it.
Exec	Can **mitigate** the action (reduce or disrupt its effect) - or do something entirely unrelated.
Recover	Cannot affect the completed action at all - but the target is vulnerable, so this is an **exploit window**: act while they wind down.

Reactions never *need* to counter the triggering action. They trigger on phase events, not as restricted counters: a magehunter reacts to a mage's casting by throwing a knife; a mage under attack reacts by casting shield for extra AC.

Reactor cost - the reaction's own PER. The tax lives in the reaction's own cycle, never the parent action's. Reactions should be built with short or zero prep and short exec; the **recover is the throttle** - pad it to tax use. This composes with the exploit window: reaction spam leaves the actor in a long recover, where enemies can take advantage.

Interruptibility (pause vs cancel). When an actor reacts, what happens to their in-progress action is an action property: actions that demand deep concentration are cancellable on reaction; lighter actions pause and resume once the reaction fully completes (including its recover). Set per action, like any other parameter.

Baiting is a core player skill: a monster that has already spent its reaction window has no answer to a maneuver.

Cancellation and Readability

Cancellation and Readability

Cancel actions

Cancelling is itself an action with its own PER cycle - not a special-case mechanic. There will be multiple actions that implement cancels:

- **Boundary cancel** - the deliberate cancel at the prep-to-exec edge (below).
- **Deep-prep cancel** - the mid-prep bail action (below).
- **Cancel-on-reaction** - reactions that explicitly cancel the triggering actor's parent action as their effect.
- **Cancel-as-part-of-another-action** - feats or abilities that cancel the parent action as a side effect of doing something else.

Each cancel implementation is its own action with its own prep/exec/recover parameters and its own readability profile. Whether a cancel from a reaction or another action actually lands is subject to the target action's interruptibility (see Resolution). The one action state machine applies to cancels too.

Cancellation policy

Two windows, both in force:

- **Boundary cancel (primary)**: cancels allowed at the prep-to-exec boundary, never between exec and recover. Cheap but never free - even a minimal PER cycle, because a free cancel means commitment means nothing.
- **Deep-prep cancel (fallback)**: cancelling mid-prep is allowed but costs scale with invested prep: `cancel_time = min_cancel + invested_prep * factor` A minimum cancel time prevents instant-cancels whenever an interrupt appears.

Readability - the public/private state layer

Every actor's phase state has two layers:

- **Public**: what the world observes (and the UI shows).
- **Private**: what is actually happening in the sim.

Feats and abilities modify the broadcast, not the sim:

- Fake a cancel (broadcast "recovering", actually still prepping).
- Pretend to continue an action while actually cancelling.
- Read others' cancels more accurately (requires Lore/Int investment).

Cancelling must telegraph (the flinch): hands relax, grip shifts. Else cancel-spam has no counterplay. An experienced enemy reads the flinch and punishes it; a wolf does not.

This is also where the game's knowledge system lives: what you and the monsters can *read* about each other's intentions is a stat, not a given.

Movement and Terrain

Movement and Terrain

Movement is a PER action like any other. Terrain is paid **at the borders**:

Phase	Source of cost
Prep	Departure difficulty of the tile you're on (getting off the log is hard)
Exec	The transfer itself - physical or mental traits
Recover	Arrival difficulty of the destination tile (landing on the log is hard)

Physical traits for exec: dex for balance, strength and encumbrance for load. Mental traits can make the same transfer harder too - fighting through an illusion, or opposing a strong will to continue.

Worked examples (log crossing)

1. **Grassy bank -> log**: minimal prep (solid footing), traits-driven exec (simple step), significant recover (getting stable on the log).
2. **Log -> log**: harder prep (concentrating on footing before shifting weight), same exec (physical or mental traits), same recover (log is still hard to settle on).
3. **Log -> bank**: hard prep (leaving the log), same exec, easy recover.

Rules that fall out of this:

- A multi-tile crossing (long bridge) pays at **every edge** - each tile-to-tile transition is its own PER cycle.
- Difficult terrain can add to all three phases (departure and arrival both hard).
- **Reactions during movement follow the normal rules.** Each movement action is a single tile-to-tile step (not a multi-tile path), so there is no special "mid-path" case - the reactor-side phase table applies exactly as with any other action.

Danger tiles and 3D

Proposed direction: **tiles are 3D, so there are no "open" tiles.** Stepping into a space that is open in the current level means you **fall** until you hit a solid tile, then you're on that level. With a fly or levitate effect (equipment or spell), you stay at your current height unless you take an action to ascend or descend.

Why this stays an open question: ASCII display. Possible techniques: shading on terrain tiles to indicate space above, blank tile to indicate space below.

Open decisions

- **Tile size**: leaning toward 1 tile = 1 cubic foot. A typical human is then about two wide and one deep (footprint), which roughly matches the traditional roguelike glyph (taller than it is wide). Bigger creatures occupy multiple tiles - an ancient dragon should not be the same single character as a human.
- Confirm the per-tile movement baseline at even terrain and the tick basis (see Open Questions).