

Author: Jayke Paver

Module Type: Additive Module

Compatible With: Frontiers Beta 2026-1 + any system using the default Gradient Resolution and Flow of Play rules.

This document is a **modular addition** to the Frontiers Engine.

The core Frontiers rules do not assume combat as a required pillar of play. A system built on Frontiers may focus on exploration, politics, survival, intrigue, social maneuvering, or entirely nonviolent interaction. Nothing in the engine mandates structured combat.

This module exists for systems that choose to include it.

The Frontiers Combat Module provides an endorsed, example implementation of structured conflict using the existing engine architecture. It is not a core requirement. It does not redefine baseline mechanics. It does not alter the Gradient Resolution System, the Flow of Play structure, or the Attributes and Derivatives framework. It layers on top of them.

In other words, this is not “how Frontiers does combat.”

This is “a way Frontiers can do combat.”

This module serves three purposes:

First, it offers a clear and adaptable framework for resolving violent or forceful conflict using the existing engine structure.

Second, it demonstrates how a subsystem attaches cleanly to the Frontiers Engine without modifying its foundation.

Third, it provides a stable reference implementation that system designers may adopt, modify, expand, or replace.

Use of this module is optional.

It is supported and endorsed, but not mandatory.

Systems built on Frontiers may:

- Adopt this module as written.
- Modify portions of it.
- Expand it with additional combat layers.
- Replace it entirely with a different combat framework.
- Omit structured combat altogether.

This module assumes use of the default Gradient Resolution and Encounter Play rules. If a system alters those foundational systems, adjustments may be required.

Combat in Frontiers is not a separate engine.
It is a deliberate application of the existing one under pressure.

Combat in Frontiers always occurs within **Encounter Play**.

This means combat inherits all timing and structure from the Flow of Play system:

- Time is divided into Rounds.
- Participants act in Turns.
- Actions cost ♦Action Points.
- Movement uses »Movement Points.
- Momentum determines order.

This module does not redefine any of those mechanics. Combat is simply a context in which those mechanics are applied under heightened stakes.

When an encounter escalates into violence or structured forceful opposition, the game shifts into Encounter Play. From that point forward, all standard Encounter rules apply as written in the core engine. Combat does not introduce a new timing layer; it intensifies the existing one.

Combat in this module rests on four principles.

First, attacks are simply ♦Actions. They follow the same declaration and resolution structure as any other mechanically significant effort in Encounter Play.

Second, Defense values function as Difficulty Ratings. An attack compares its Resolution result against either □ Physical Defense or □ Mental Defense, depending on the nature of the harm.

Third, success and harm are distinct. The Resolution Roll determines whether an attack connects. Damage determines how much harm is inflicted. These are related, but they are not the same question.

Fourth, Gradient expresses texture rather than polarity. A positive or negative Gradient does not reverse success or failure. Instead, it shapes the tone, consequence, and momentum of the exchange.

This approach keeps combat fully integrated with the core design philosophy of the Frontiers Engine rather than introducing a separate mechanical identity.

An attack is a standard ♦Action performed during Encounter Play.

When a participant chooses to attack, they first declare their intent. This includes naming the target, describing the method of attack, and clarifying the approach. As with all Resolution Rolls, the task determines the governing Attribute. A heavy overhead strike might use ♦Vigor. A precise ranged shot might use □Finesse. A psychic intrusion might rely on □Acuity or □Resolve. The Game Master confirms the final Attribute selection.

Once the Attribute is determined, the attacker identifies which Defense the attack targets. Physical harm compares against □ Physical Defense. Psychological, emotional, or cognitive harm compares against □ Mental Defense. These Defense values function as Difficulty Ratings, as established in the Attributes & Derivatives rules.

The attacker then rolls the Resolution Die, adds the relevant Attribute Modifier, and applies any Skill Benefit or situational modifiers. If the Modified Result is equal to or greater than the target's Defense, the attack succeeds. If it is lower, the attack fails.

There are no automatic successes or failures unless a system built on Frontiers explicitly introduces them. The final modified result compared to Defense determines the outcome.

If the attack succeeds, damage is rolled.

Damage represents reduction of a target's ☯Health Pool.

Weapons, abilities, and harmful effects define their own damage expressions. This module does not prescribe specific dice sizes, damage tables, or mandatory scaling assumptions. Instead, it establishes that a successful attack inflicts harm according to the damage expression defined by the system.

A weapon or damaging effect should clearly state:

- Its damage dice.
- Whether any flat modifiers apply.
- Its range in □Units.
- Any relevant traits or tags.

Some systems may choose to add the governing Attribute Modifier to damage. Others may prefer damage dice to stand alone. Both approaches are structurally valid within this module. The decision will directly shape lethality and pacing, and should therefore be made intentionally at the system level.

After rolling damage, the total is subtracted from the target's ☯Health Pool. If this reduction brings the target to 0 ☯HP, the default rules for gaining a □Wound and falling Unconscious apply. This module does not alter the Wound system, the death threshold, or recovery mechanics.

Combat is one of the most common contexts in which Gradient meaningfully applies.

After determining whether an attack succeeds or fails, the table may roll the Gradient Die when deviation would meaningfully enrich the outcome. The Gradient result does not reverse the binary result of the Resolution Roll. Instead, it modifies how that success or failure manifests.

On a successful attack, a positive Gradient might represent increased damage, superior positioning, tactical leverage, or sustained pressure. A negative Gradient might represent partial damage, exposure to retaliation, overextension, or resource strain. The attack still succeeds; the Gradient shapes its impact.

On a failed attack, a positive Gradient might represent a glancing blow, maintained positioning, or preparation for a follow-up. A negative Gradient might represent vulnerability, imbalance, or lost tempo. The attack still fails; the Gradient shapes its consequence.

The exact mechanical interpretation of each Gradient band is intentionally left to the system designer or table. What matters is consistency. Once a combat system defines how Gradient modifies damage or positioning, that interpretation should remain stable within that system.

Not all combat interactions are resolved against static Defense values.

When two participants directly oppose one another in a struggle—such as grappling, disarming, overpowering, or resisting mental domination—the interaction may use Contested Resolution. In such cases, both participants roll a Resolution Die and compare results. The higher modified result prevails.

If degree matters, one or both participants may roll Gradient to determine how decisively the exchange resolves.

Contested combat follows the same structure as other contested interactions in the engine. No additional combat-specific framework is required.

Abilities operate exactly as defined in the Abilities system.

A combat-oriented Ability may require a Resolution Roll, may apply damage directly, may compare a defined value against Defense, or may impose narrative or mechanical effects without rolling to hit. The presence of combat does not alter the universal Ability structure.

If an Ability compares against Defense without using a standard Resolution Roll, it should clearly state how its effect scales when the target's Defense exceeds or falls below its defined threshold. This scaling remains consistent with the principle that Defense functions as a passive Difficulty Rating.

All Action Point costs, Energy costs, Overspending rules, and timing constraints remain unchanged in combat.

A participant is removed from active combat when they are rendered unable to act.

This most commonly occurs when:

- Their ☒Health Pool is reduced to 0 and they gain a ☐Wound.
- They exceed their maximum ☐Wounds.
- They exceed their maximum ☐Fatigue.
- A specific effect renders them incapacitated.

This module does not introduce additional defeat states such as death saves, bleed-out timers, or stabilization subsystems. Designers who wish to include such mechanics may create supplemental modules that build on this one.

Recovery from harm follows the standard $\mathbb{Z}^2$ Downtime rules. Combat does not alter recovery pacing unless another module specifies otherwise.

Equipment interacts with combat through damage expressions, Defense bonuses, range definitions, and potential $\mathbb{Z}$ Energy strain if used above character level.

This module does not define weapon lists, armor categories, or mandatory equipment scaling. Instead, it establishes expectations for clarity.

A weapon should clearly communicate what Attribute it commonly uses, what Competency it may require, how it expresses damage, and whether it carries any special traits. Armor may increase Defense, reduce damage, or introduce other defensive mechanics, depending on system design.

All such implementations attach to the combat framework presented here without modifying core engine rules.

The lethality and pacing of combat are not determined by this module alone.

They emerge from:

- The size and scaling of $\mathbb{H}$ Health Pools.
- The average damage per successful attack.
- Whether Attribute Modifiers add to damage.
- The frequency and interpretation of Gradient.
- The availability of healing.
- The availability and strain of $\mathbb{Z}$ Energy.

A heroic fantasy system may allow characters to endure multiple Rounds of sustained combat. A survival horror system may make each hit catastrophic. A science-fiction system may emphasize equipment strain and Energy management over raw durability.

This module provides structure.

The system designer determines tone.

To adopt this Combat Module within a Frontiers-based system:

1. Confirm use of the default Gradient Resolution structure.
2. Confirm use of the default Encounter Play timing and Action economy.
3. Define how weapons express damage.
4. Define whether and how Attribute Modifiers apply to damage.
5. Define how Gradient modifies combat outcomes.
6. Define Abilities and Equipment consistent with this framework.

Attribution Line: "Includes material from the **Basic Combat Module** by Jayke Paver"

This module may stand alone as the default combat structure for a system, or it may serve as a foundation for additional genre-specific expansions.