

The d100 is the d100 adaptation of the d18 Ruleset, the default form of Frontiers. It resolves actions by rolling **2d10** as a percentile pair against a Target written on the sheet: meet or roll under it and the action succeeds. It trades the default form's curved reliability for width, with room for competence and circumstance to be exact, visible numbers.

WHAT IT IS

Attributes sit on a hundred-point scale, and each carries three pre-computed Targets. Difficulty picks the Target, Defense and circumstance adjust it, and the roll comes in at or under the final number. The units die is read a second time as the **Gradient**: how strongly the result landed.

WHY YOU'D CHOOSE IT

Your Score is your chance. A character with a ♦ Vigor of 70 succeeds at routine Vigor work 70% of the time, and can read that off the sheet before deciding to try. A hundred points also has room for circumstance to be numeric: cover, elevation, darkness, and range can each be a real number a player sees and weighs, without any one of them swamping the roll. And nothing becomes automatic. Every difficulty tier is clamped at both ends, so even a master never quite makes routine work certain, and the hardest tier stays a gamble for anyone.

Which Gradient you get, depending on whether you succeeded

Success	20%	60%	20%
Failure	20%	60%	20%

■ Positive ■ Standard ■ Negative

The units die decides the Gradient, and barely moves with the verdict

Flat, like the d20, but read as a true percentage. A Target of 65 means a 65% chance, with no arithmetic between the number on the sheet and the odds at the table.

HOW IT FEELS

Exacting, tactical, and readable:

- The odds are on the table. Every adjustment is visible before the dice commit, so choosing to act is itself a decision.
- Mastery buys pressure, not comfort. Pushing an Attribute past a ceiling stops improving easy work and keeps improving hard work.
- Failure is always live. A natural 100 always fails, and no Target reaches certainty.

Reach for the d100 when you want to see the odds, adjust them deliberately, and live with them.

EVERY VERSION AT A GLANCE

Before you continue, here are all three Rulesets side by side, for a quick sense of how this one differs from the others.

	d18	d20	d100 — this Ruleset
Dice	1d12 + 1d6 , summed	1d20 , plus an independent 1d10	2d10 as a percentile pair
Reading	Meet or beat the target	Meet or beat the target	Meet or roll under the Target
Difficulty	DT on a four-point ladder	DT on a five-point ladder	Picks one of three Targets on the sheet
Gradient	The 1d6 , welded: feeds the sum and colors the result	A separate 1d10 , independent of the verdict	The units die, reread: nearly independent
Circumstance	Favor and Disfavor; numbers are the exception	Favor and Disfavor; numbers are the exception	Numbers first; Favor for the larger swings
Passive play	Take your Score, or roll to reach past it	Take your Score, or roll to reach past it	No passive value; the Trivial tier does its work

For the reasoning behind the engine, and the variants, alternatives, and optional mechanics it supports, see the **Design Documentation**.

The Resolution System

The Resolution System answers any question the fiction cannot settle on its own. When the outcome of an action is uncertain and the result matters, you resolve it here.

A resolution uses the **Resolution Dice**: two ten-sided dice, one read as **tens** and one as **units**, together giving a result from 1 to 100. They answer two questions at once:

1. **Does it succeed?** Answered by the **pair read together** as a percentile result.
2. **How strongly does it land?** Answered by the **units die** alone, read as the **Gradient Die**.

Both questions come from a single throw.

The Resolution Dice

The **Resolution Dice** decide success or failure. In this Ruleset they are two d10 , one designated the tens die and one the units die, read together as a number from **1 to 100** (a result of 00 reads as 100).

Nothing is added to the dice. You roll against a **Target**: a number already written on your sheet, adjusted at the table by whatever the situation adds or takes away. The reading is **meet or roll under**:

- **Equal to or under** the Target: the action **succeeds**.
- **Over** the Target: the action **fails**.

Rolling low is good. A natural **100 always fails**, whatever the Target, so no action is ever truly certain.

Targets Live on the Sheet

A character's Attributes are recorded on the hundred-point scale, and each one carries its three **Targets** already worked out, so nothing needs multiplying at the table.

Attribute	Trivial	Routine	Challenging
❖ Vigor	99	70	35
☆ Deft	75	50	25
● Acuity	90	60	30
△ Resolve	60	40	20

The middle column is the Attribute itself, the number a player thinks of as their Score. The two flanking values are what that Score becomes when a task is easier or harder than ordinary. How Scores are generated, and what the scale means, is covered in Attributes & Derivatives; how they are chosen at creation is covered in Characters.

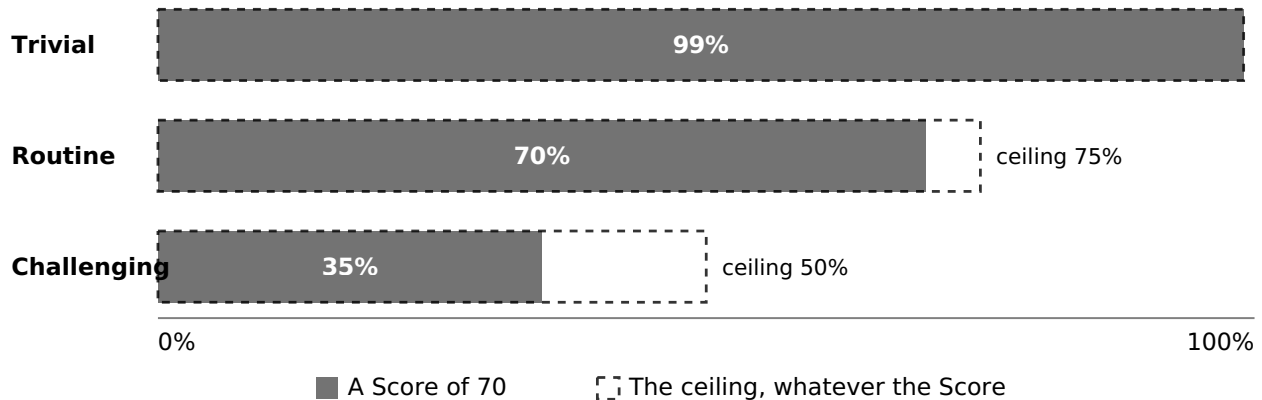
Difficulty

Difficulty is not a separate number to beat. It selects **which of a character's three Targets** is used, so the roll is always against that character's own competence, scaled to the task.

Difficulty	Target	Ceiling
Trivial	Score × 1½	99
Routine	Score	75
Challenging	Half Score	50

Round up. **Routine** is the default and the reason a Score reads as a percentage: at ordinary difficulty, a 70 Vigor character simply succeeds 70% of the time.

What each difficulty tier is worth, and where the ceilings cut it off



Trivial work is already at its ceiling; Challenging is where a higher Score still has room

The ceilings are the d100's temperament made numeric. A Trivial task tops out at **99**, so even the easiest work carries a single failing result on the dice. A Routine task can never exceed **75%**, so even an exceptional character fails ordinary work about one time in four. A Challenging task can never exceed **50%**, so the hardest tier remains a genuine gamble for anyone, however skilled.

Mastery buys pressure, not comfort

Because Routine and Challenging are both clamped, pushing an Attribute past the point where it meets a ceiling stops improving ordinary work and keeps improving hard work. A Score of 80 and a Score of 100 are identical on a Routine task and clearly different on a Challenging one. Mastery in the d100 form does not mean failing less at everyday things; it means holding up when things get difficult. That is deliberate, and it is the clearest expression of what separates this Ruleset from its siblings.

Adjusting the Target

Once the difficulty tier is chosen, the situation adjusts the Target further. This is where the hundred-point scale earns its place: adjustments are **numeric**, and a player sees them before committing the dice.

Two things adjust a Target:

DEFENSE

When an action is taken against another character, that character's relevant **Defense** is subtracted from the Target. Defense is $3 \times$ half the sum of the relevant pair's **Ranks**, rounded up, and it too is written on the sheet.

CIRCUMSTANCE

The GM may add or subtract for position, cover, lighting, range, improvised tools, and the like. As a guide, a **slight** edge or hindrance is worth about **5**, a **clear** one about **10**, and a **decisive** one about **20**. These are advisory; a system may set its own.

Example. A character with ♦ Vigor 70 attacks an opponent whose □ Physical Defense is 12. The GM calls it Routine, so the Target begins at 70, less 12 for the Defense, and less 5 more because the footing is bad: a final Target of **53**. The player rolls **d100** and needs 53 or under.

A Target is never reduced below **1**, and a character may always attempt an action, however poor the odds.

\\"Roll against a Defense\\" means subtract it

Abilities, weapons, and effects throughout Frontiers are written as a Resolution roll against a target's □ Physical Defense or □ Mental Defense. That phrasing is shared across every Ruleset so the same ability card works in all of them; each Ruleset defines how it resolves. In the d100 form it means: **take your Target for the action and subtract that Defense from it**. Nothing else about the ability changes.

The Gradient Die

The **Gradient Die** decides degree rather than direction. In this Ruleset it is the **units die** you already rolled: after the pair decides success or failure, the units die is read a second time, on its own, for how strongly the result lands.

Gradient Die (units)	Result
0 – 1	Positive
2 – 7	Standard
8 – 9	Negative

Low is strong, matching the Ruleset's roll-under grammar: the units die runs 0 low to 9 high, so the same face that is best for the result is also strongest for the Gradient.

- **Standard** results, six in ten, land exactly as the success or failure implies.
- **Positive** and **Negative** results are turning points: a breakthrough, a breakdown, a scene-changing consequence.

The units die contributes only a fraction of the percentile result, so the Gradient is nearly **independent** of success: texture laid over the verdict rather than a force inside it.

The Gradient never reverses the verdict

A Positive Gradient on a **failure** does not make it a success. It makes the failure productive: you still fail, but you gain ground, learn something, or come out better positioned. A Negative Gradient on a **success** does not make it a failure. It makes the success costly: you still succeed, but it bites. Direction comes from the percentile result alone, and the Gradient only colors it.

Configuring the Gradient

The default reads the units die in three tiers, wide in the middle and sharp at the edges. A designer who wants a finer read has a supported alternative, a change only to how the die is read:

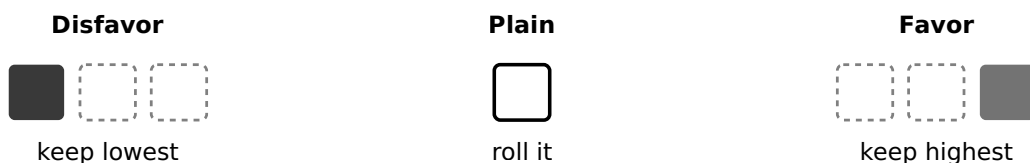
- **Three tiers (default).** 0–1 Positive, 2–7 Standard, 8–9 Negative. Deviations are uncommon and land hard.
- **Five tiers.** 0 Major Positive, 1–2 Minor Positive, 3–6 Standard, 7–8 Minor Negative, 9 Major Negative. A finer read that separates a wrinkle from a turning point, at the cost of a second band to remember.

These change only the reading of the die. Further options, including dropping the Gradient entirely or building a longer scale, live in the Design Documentation.

Favor and Disfavor

Favor and **Disfavor** are the engine's universal circumstance modifier. In this Ruleset they apply to **one physical die**, never to the pair.

Favor adds a die and keeps the best; Disfavor adds one and keeps the worst



Circumstance changes the dice you roll rather than the number you add

- **Favor:** roll one extra die of the chosen kind and keep the **better** result for that die.
- **Disfavor:** roll one extra die of the chosen kind and keep the **worse** result for that die.

The other die stands exactly as rolled. You never reroll the whole percentile result. Better and worse are judged on the **final percentile reading** instead of the face of the die alone: keep whichever result leaves the lower total once both dice are read together, remembering that a pair reading 00 counts as 100, the worst result of all.

Placed on	Effect
The tens die	Roll an extra tens die, keep the better. The strongest way to shift whether an action succeeds.
The units die	Roll an extra units die, keep the better. Shifts the result slightly, and shifts the Gradient substantially, since that die is read twice.

Favor is **placed before the die it affects is rolled**: the player chooses which die it goes on, then rolls that die. Favor is never added to a die already on the table. Disfavor is the opposite, with

whatever imposes it naming the die, or the GM doing so. When an effect grants Favor without naming a die, it applies to the tens die.

Favor and Disfavor each stack up to **3**: cap each side at 3 first, then cancel one against one, so the two never apply together. Each additional stack adds one more die of that kind, keeping the best result among them.

Numbers for the situation, Favor for the swing

A hundred-point range has room for small numeric adjustments, so this Ruleset uses both tools:

numeric adjustments for the ordinary texture of a scene, and **Favor** for the larger swings, for what training provides, and for effects a rule names directly. Reach for a number when the situation is a matter of degree, and for Favor when it is a matter of kind.

At a Routine Target of 50, Favor on the tens die raises your odds of success from 50% to 75%: a value equivalent of about +25. That is larger than any circumstance number the GM hands out, which is why Favor is kept for the swings.

Value equivalents

Favor and Disfavor change your odds of success by changing which dice you roll; a modifier changes the number you add. To help you compare the two, *Frontiers* sometimes gives a **value equivalent**: the size of modifier that would move your odds of success about as far.

A value equivalent is a comparison, not a number you add. It changes with the Ruleset, the difficulty, and the Rank, so every one says which roll it describes. Written out, it reads: a value equivalent of about +2.

Rolling the Dice

A full resolution runs in order:

1. **Declare intent.** The player states what they are trying to accomplish.
2. **Find the Attribute.** The GM identifies which Attribute governs the attempt.
3. **Choose the difficulty.** The GM calls the task Trivial, Routine, or Challenging, which selects one of the three Targets already written on the sheet. Attacks are Routine unless an ability states otherwise.
4. **Adjust the Target.** Subtract the relevant Defense if the action is against another character, then apply any circumstance the GM calls for. A Target is never reduced below 1.
5. **Check for a Skill.** If the approach relies on trained technique, the player gains **Favor** and places it on the tens die or the units die before rolling that die.
6. **Roll the Resolution Dice.** Roll both **d10** and read them as a percentile result from 1 to 100. Meet or roll under the adjusted Target to succeed.
7. **Read the Gradient Die** if the degree matters: look at the units die you just rolled.

The Resolution Dice are always rolled together. Reading the Gradient is free, since the units die is already on the table. Consult it when the texture of the result matters, and ignore it when it does not.

A Full Resolution, Worked

A character is scaling a rain-slick wall while guards close in below.

1. **Declare intent.** "I climb to the balcony before they reach me."
2. **Find the Attribute.** Climbing under pressure is ♦ Vigor. The character's ♦ Vigor is 70, so their sheet reads **Trivial 99 / Routine 70 / Challenging 35**.
3. **Choose the difficulty.** A rough wall is ordinary work for a climber: the GM calls it **Routine**, for a Target of 70.
4. **Adjust the Target.** No one is defending against the climb, but the rain is a clear hindrance, so the GM takes 10 off. The Target is **60**.
5. **Check for a Skill.** They have the Athletics Skill, granting Favor. The player places it on the tens die.
6. **Roll the Resolution Dice.** With Favor, the player rolls two tens dice and one units die. The tens come up 7 and 4, and the units die shows 1. Keeping the better tens die gives a reading of **41**, under the Target of 60, so the climb **succeeds**.
7. **Read the Gradient Die.** The units die shows a **1: Positive**. The GM narrates it: the character surges up the wall and vaults the rail in one motion, landing in cover before a guard thinks to look up.

The character succeeded, and the Gradient did not change that. It made the success decisive.

Contests

When two characters oppose one another directly, both **roll**, each against their own adjusted Target. A Contest is always rolled. Both sides roll at the same difficulty tier, **Routine** unless the system sets another, even when the two sides use different Attributes.

- **Both succeed, or both fail:** the one who rolled **further under** their Target wins, having managed it more cleanly.
- **One succeeds and one fails:** the one who succeeded wins.
- **Exact tie of margin:** the outcome is the GM's to settle. The recommended method is to favor the defender, with a reroll as an alternative.

Only the **winner's** Gradient Die is read, and only if the degree matters. A Contest has one result, so it takes one reading.

Flow of Play

Frontiers is played in three **modes**, each defined by how fast time moves. The mode sets the tempo, and the tempo decides what kind of effort a character can bring to bear.

Mode	Tempo	You mostly use
Encounter	Seconds	♦ Actions
Exploration	Minutes	♦ Actions and * Activities
Endeavor	Hours to days	* Activities

The table is guidance. The modes describe how play generally feels, and a system or table should lean on them however fits its game.

♦ Actions and * Activities

Every deliberate thing a character does is either an ♦ **Action** or an * **Activity**, and the line between them is intentionally soft. In the broadest sense every effort is an * Activity, a thing you do over time. An ♦ Action is the tight, immediate kind of effort, the sort that matters when seconds count.

The one firm rule is tempo:

- An ♦ **Action** is quick enough to happen inside an Encounter Round. Swinging a weapon, activating an ability, vaulting a railing.
- An * **Activity** takes longer than a single Round. Studying a wall, searching a library, resting, crafting, traveling. * Activities cannot be performed during an Encounter.

Everything past that is guidance. ♦ Actions can occur in any mode, but they are only metered, paid for with ♦ Action Points, inside an Encounter. In Exploration a character might take an ♦ Action and roll for it with no economy tracking it. In an Endeavor a character rarely reaches for an ♦ Action at all. Use the split the way that suits your table.

Encounter

Seconds matter. Combat, a chase, a leap from one cable car to the next. Encounter play is the only mode with a tactical economy: play breaks into structured **Rounds** and **Turns**, characters act through ♦ Actions paid for with ♦ Action Points, and turn order, ⇒ Movement, and Reactions all come into force. The rest of this section details how an Encounter runs.

Encounter Structure

An Encounter runs in repeating **Rounds**. In each Round, every participant takes one **Turn** in ► Initiative order. When all have acted, a new Round begins. There is no limit to the number of Rounds. A Round is roughly 10 seconds of in-world time.

When play shifts into an Encounter, all participants determine ► Initiative and a new Round begins; no one starts with ♦ Action Points or ⇒ Movement already spent. The Encounter ends when the conflict resolves or the GM judges that structured timing is no longer needed, and play returns to Exploration.

Turns

At the start of a participant's Turn:

- ♦ Action Points refresh to maximum.
- ⇒ Movement refreshes to maximum.
- Effects that expire "at the start of your Turn" resolve.

During their Turn, a participant may spend ♦ Action Points on ♦ Actions, spend ⇒ Movement, and use ◇ Free Actions.

At the end of their Turn:

- Unspent ⇒ Movement is lost.
- Unspent ♦ Action Points remain available for Reactions until the participant's next Turn.
- Effects that expire "at the end of your Turn" resolve.

A participant acts only on their own Turn unless a rule grants a Reaction.

► Initiative

► Initiative sets turn order. At the start of an Encounter, each participant chooses one of two methods, and that choice holds for the whole Encounter:

- **Flat.** Take a standing value of $10 + ☆ \text{ Deft Rank}$.
- **Rolled.** Roll $1d10$ (either die of the percentile pair) $+ 5 + ☆ \text{ Deft Rank}$.

(Attribute Ranks are defined in Attributes & Derivatives; a Rank is an Attribute's Score minus 50, divided by 5.)

Participants act in descending order, highest first. On a tie, the higher ☆ Deft Rank acts first; if still tied, those participants roll off.

Once set, the order holds for the Encounter unless an effect changes it. A change to ► Initiative takes effect at the start of the next Round and never reorders Turns already taken.

♦ Actions and Action Points

An ♦ **Action** is a discrete, mechanically significant effort taken during an Encounter. Unless stated otherwise, an ♦ Action costs **1 ♦AP**.

Each participant begins their Turn with **3 ♦AP**. ♦ Action Points refresh to maximum at the start of each Turn; they do not accumulate, so any left from the previous Round are overwritten. Between Turns, unspent ♦ Action Points remain available for Reactions.

To take an ♦ Action: declare it, pay its full ♦ Action Point cost, then resolve it completely before declaring another. A participant cannot begin an ♦ Action they cannot pay for.

Some ♦ Actions cost more than 1 ♦AP. The full cost must be available and paid before resolution begins; partial payment is not allowed.

◇ Free Actions

A ♦ **Free Action** is a minor effort that does not meaningfully tax a participant's time or focus. It costs no ♦ Action Points, and is used on the participant's own Turn unless a rule states otherwise.

Two limits apply:

- Each individual ◇ Free Action may be used **once per Turn**. Different ◇ Free Actions may be used in the same Turn, but the same one may not be repeated unless a rule allows it.
- A participant may use at most **3 ◇ Free Actions** in a single Turn, however many they have available.

◇ INTERACT

Handling equipment is a single ◇ Free Action called **Interact**: drawing, Stowing, Equipping, picking up, handing off, or otherwise manipulating an item (see Equipment). Because all item handling shares one ◇ Free Action, a participant may Interact only once per Turn, whichever item is involved. Drawing a blade and Stowing a shield in the same Turn is two Interacts, and therefore not possible without a rule that says otherwise.

Reactions

A **Reaction** is an effect used outside a participant's own Turn, in response to a trigger. Reactions are paid from the participant's current ♦ Action Point pool, the same pool used on their Turn. Because unspent ♦ Action Points persist through the Round, every Turn carries a choice: spend freely now, or hold points back to react. A participant with no ♦ Action Points available cannot react, and cannot react while incapacitated.

⇒ Movement

On their Turn, a participant may move a number of □ **Units** up to their ⇒ **Movement**, equal to 5 + ☆ Deft Rank, never less than 1. ⇒ Movement refreshes at the start of the Turn, and any unspent is lost at its end. ⇒ Movement may be split freely before, between, and after ♦ Actions.

One **Unit** is 5 feet, or one square or hex on a tactical map. Units measure movement, ranges, and areas of effect alike. Without a grid, Units stand in for relative positioning at the GM's discretion.

⇒ Movement forced by an external effect does not cost ⇒ Movement and occurs immediately when the effect resolves.

⚡ Energy Economy

Most ♦ Actions cost only ♦ Action Points. The more demanding ones, powerful abilities and strenuous feats, cost ⚡ **Energy** as well.

⚡ **Energy** is a character stat: a reserve of exertion a character draws on to reach past their own limits. Everything usable has a **Demand**, the level of power it operates at, and every character has a **Capacity**, the level they sustain without strain. Using something above your Capacity costs ⚡ Energy equal to the gap, and anything at or below it costs none. The pool grows slowly as a character advances, and is spent during play and regained through rest (see Vitality, Exertion & Death).

This section introduces ⚡ Energy only where it meets the action economy: some ♦ Actions cost ⚡ Energy on top of ♦ Action Points. How ⚡ Energy is sized, spent, overspent, and recovered is covered in full under Vitality, Exertion & Death.

Exploration

Exploration is the ordinary business of play between crises: moving through the world, dealing with the people in it, and working out what to do next. It is the default mode and the vast majority of a session. Where an Encounter resolves an immediate problem in quick succession, Exploration is the slower, open space around those problems, the free mixing of ♦ Actions and ✱ Activities with no economy tracking either.

Exploration runs on **choices**. The GM describes a situation, the players decide what their characters do, the fiction updates, and it repeats. Most decisions resolve through description alone; a roll enters only when the outcome is uncertain and the result matters (see The Resolution System). Most of the play happens in that decision, and the dice only settle what the description leaves open.

There is no ♦ Action Point tracking, no turn order, and no ⇒ Movement budget. Time is loose, measured in minutes at the pace the fiction needs. When a situation tightens so that order, position, and timing suddenly matter, play becomes an Encounter; when it stretches into a long, uneventful span, it becomes an Endeavor.

When a roll is called for, it is almost always a single check against one of a character's three Targets. The table below anchors the difficulty tiers (see Difficulty) to concrete tasks. Treat these as examples, not a fixed list.

Example Task	Difficulty
Recall common knowledge; haggle a friendly merchant; climb a knotted rope	Trivial

Example Task	Difficulty
Force a stuck door; pick a simple lock; scale a rough wall	Routine
Persuade a wary stranger; spot a hidden catch; leap a wide gap	Routine
Sway a hostile official; disarm a clever trap	Challenging
Talk down a sworn enemy; free-climb a sheer, wet cliff	Challenging , with Disfavor

Where a task sits between two tiers, the difference is made up with a circumstance adjustment or with Disfavor rather than a fourth tier.

Exploration commonly takes a few shapes. These are illustrations. Treat them as a starting point.

Interaction

Talking with the people of the world: negotiating, gathering rumors, buying and selling, reading a room. Most interaction is pure roleplay and needs no dice, and the best social play often never touches the sheet. When an outcome is genuinely uncertain, resolve it with whichever Attribute fits the approach, ♦ Vigor to lean on someone, △ Resolve to hold a line, ☆ Deft or ● Acuity to charm or misdirect, plus Favor from a Voice Skill (see Attributes & Derivatives). One roll usually settles a social beat; save repeated checks for a situation that meaningfully shifts.

Investigation

Searching a place, examining an object, piecing clues together, working a puzzle. What is plainly visible needs no roll at all: a GM simply tells the players what their characters see. What takes a moment's attention is a **Trivial** ● Acuity check, high enough that a character will almost always catch it. Finding what is deliberately hidden is **Routine** or **Challenging**, depending on how well it was hidden. Reserving the dice for what a scene is actually concealing keeps exploration moving.

Traversal

Moving through a space that is not itself a fight: a dungeon, a market, a stretch of wilderness. Outside an Encounter there is no ⇒ Movement budget; characters simply go where they choose, and a check happens only at a real obstacle. Traversal is handled one of two ways, whichever the table prefers:

Approach	How it works
On a grid	Position and distance are tracked in □ Units. Characters move freely between checks, with no ♦ Action Point or ⇒ Movement limit outside an Encounter.
In narrative	The GM tracks position loosely. Distances are described rather than measured, and a roll is called only when an obstacle demands one.

Endeavor

An Endeavor commits a long stretch of time, hours or days, to a single pursuit, handled as one compressed span instead of moment to moment. Where an Encounter is seconds of pressure and Exploration is the minute-to-minute business between, an Endeavor is the fast-forward: the GM narrates the span, pausing for a beat of roleplay or a single roll when something is worth zooming in on, then moving on.

Endeavors run almost entirely on ✱ Activities. No ♦ Actions are spent and no economy is tracked, since what the character is committing is time.

The kinds of Endeavor below are examples. Any activity that takes time can become one. Only **Rest and Recovery** carries firm rules; the rest are patterns a table can use, reshape, or ignore, and their numbers are defaults to tune.

Rest and Recovery

Rest is the most common Endeavor: a stretch of safe, uninterrupted time spent recovering. It is the one Endeavor with fixed rules.

Rest restores + HP, ⊗ Wounds, ⊖ Fatigue, and ♀ Energy on the schedule defined in Vitality, Exertion & Death → Recovery. In brief: ♀ Energy returns each hour, and every three uninterrupted hours restore half of maximum + HP and remove one ⊗ Wound or one ⊖ Fatigue. Interrupted rest loses that interval's progress.

Downtime

Downtime is time spent on a pursuit, not recovery, where characters do the things that never fit inside a scene. Common examples, with rough scopes a table can adjust:

Pursuit	What it is	Example scope
Training	Improve a capability, such as raising an Attribute or gaining a Skill	Roughly a week of focused effort per step; the improvement itself runs through Characters (Advancement)
Project	Craft, build, repair, or create something	A simple piece in a day; a complex one over weeks
Trade	Run a business or work a job	Earns resources across the span
Study	Research a question or learn a subject	The answer emerges with the time invested

The scopes above are illustrative. What a span of Downtime yields, and what it costs, is the table's to set, and anything that permanently changes a character's sheet runs through Characters.

Travel

Travel resolves a long journey as a span, not step by step. Instead of tracking every stretch of road, the GM moves the party across the distance and punctuates it with occasional events, a hazard, an encounter, a discovery, resolved as they come.

Measure a journey in **days of travel on foot**, then apply the conveyance's pace. A day of steady walking covers open ground at a rate the setting can pin down precisely (roughly 20 to 30 miles, or 30 to 50 kilometers); everything faster is a multiple of it.

Conveyance	Pace vs. on foot
On foot	Baseline
Mount	About 2×
Draft or wheeled	About 1.5×, and hauls cargo
Powered vehicle	About 5×
Fast vehicle or flight	10× or more

One event per leg of a journey, a day or a natural stage, is a reasonable default; a dangerous route warrants more, a safe one fewer. How events are chosen, and how far a day reaches, are the table's to set.

Attributes & Derivatives

Every character is defined by four core **Attributes**. An Attribute measures a fundamental capacity, what a character is innately capable of, before training, tools, or circumstance are taken into account. Almost every roll in *Frontiers* draws on one of the four, and most other values on a character sheet are calculated from them.

The Four Attributes

Icon	Attribute	Capacity
❖	Vigor	Bodily power and endurance: strength, stamina, toughness
☆	Deft	Bodily precision and speed: coordination, agility, reflexes
●	Acuity	Mental sharpness and reach: perception, reasoning, memory
△	Resolve	Mental fortitude and steadiness: willpower, composure, conviction

❖ **Vigor** is the body's power and durability. It governs how much force a character can exert, how much weight they can move, how long they can sustain effort, and how much physical punishment they can absorb before it tells.

Vigor drives: + Maximum HP, ⊗ Maximum Wounds. In play: lifting, forcing, enduring, and any attack or feat of raw strength.

☆ **Deft** is the body's precision and tempo. It governs accuracy, speed, balance, and reflexes, how cleanly a character executes a physical action and how early they react when timing matters.

Deft drives: □ Physical Defense, ⇒ Movement, ► Initiative. In play: dodging, aiming, sleight of hand, stealth, and acting first.

● **Acuity** is the mind turned outward. It governs perception, deduction, recall, and the grasp of detail: what a character notices about the world and what they work out from it, including seeing a lie or a manipulation coming. That sharpness is one of the two things that guard the mind, which is why ● Acuity feeds □ Mental Defense.

Acuity drives: □ Mental Defense (with △ Resolve), additional Skills. In play: noticing, investigating, recalling, and seeing through deception or coercion.

△ **Resolve** is the mind turned inward. It governs willpower, composure, and emotional endurance: how firmly a character holds under fear, pressure, and strain. Will is the other guard on the mind, so △ Resolve also feeds □ Mental Defense, and it alone sets the ⊖ Fatigue track that measures how much strain a character can absorb before it tells.

Resolve drives: ⊖ Maximum Fatigue, □ Mental Defense (with ● Acuity), and lends to + HP. In play: resisting fear and coercion, holding focus, and outlasting strain.

Telling Them Apart

The four Attributes sit on two axes. One axis is whether the capacity is **physical** or **mental**. The other is whether it is about **power and endurance** or **precision and reaction**.

	Power & Endurance	Precision & Reaction
Physical	❖ Vigor	☆ Deft
Mental	△ Resolve	● Acuity

This is the quickest way to resolve overlap:

- ❖ **Vigor** vs ☆ **Deft**. Both are physical. ❖ Vigor is raw force and the ability to endure; ☆ Deft is control, accuracy, and speed. Breaking a door is ❖ Vigor. Picking its lock is ☆ Deft.
- ● **Acuity** vs △ **Resolve**. Both are mental. ● Acuity is the sharp, processing mind that perceives and reasons; △ Resolve is the enduring, anchored mind that holds and resists. Spotting the lie is ● Acuity. Refusing to break under the interrogation is △ Resolve.

Reading an Attribute

Every Attribute is read two ways: as the **Score** that states a character's competence, and as the **Rank** that sizes everything derived from it.

- The **Score** is the number on the hundred-point scale. It is a character's chance, in percent, of succeeding at Routine work with that Attribute, and it is what a player rolls under.
- The **Rank** is the same Attribute compressed into steps, used to build derived values: **+** HP, the **⊗** Wound and **⊖** Fatigue tracks, **⇒** Movement, both Defenses, and bonus Skills. **The Rank is never added to a Resolution roll.** It is worked out once, when a character is made, and after that it only appears in the values it produced, the sole exception being rolled **➤** Initiative, which is not a Resolution roll.

Coming from the d18 or d20? Unlearn the Rank

In the roll-over Rulesets the Rank is on every roll a character makes. Here it is on none of them. Your **Score** is the number you roll against; the Rank exists only to size **+** HP, the tracks, **⇒** Movement, and the Defenses, and it is computed once at creation and then largely forgotten.

This is the single easiest thing to get wrong when moving between Rulesets, and getting it wrong means adding a bonus to a roll that should not have one.

Names are not fixed

This Ruleset uses **Score** and **Rank**. Other systems may call the same two readings whatever fits their tone. Only the structure is required: one reading a character rolls against, one reading that sizes what they are made of.

Score and Rank

An Attribute's **Score** defaults to **50**. Its **Rank** is the Score minus 50, divided by 5.

Score	Rank
30	−4
40	−2
50	+0
60	+2
70	+4
80	+6
90	+8

The table is representative. Extend it as your system needs. Five points of Score is one point of Rank, in both directions, for as far as a system allows.

The Rank is what keeps every derived value in its usual band: a character with a ♦ Vigor of 70 carries a +4 Rank into + HP, the ⊗ Wound track, and the Defenses, the same way any Score maps through the table above.

Ranks can be negative

A Score below 50 gives a negative Rank. This is the only place in the core engine where a value drops below zero, and a small number of derived values are floored so they cannot break (see Vitality, Exertion & Death).

The Attribute Scale

- The default Score is **50**, the baseline of an unremarkable adult, who succeeds at Routine work half the time.
- No effect can reduce a Score below **5**.
- The default ceiling is **100** (a Rank of +10). A system may set a different cap.
- In typical play most characters sit between **40 and 75**, with specialists reaching higher in a single Attribute.

As a rough reading of the scale: **30** is unpracticed, **50** is competent, **70** is skilled, and **90** is exceptional.

A system defines exactly where its characters begin and how high they may climb (see Characters for the default creation rules).

Using Attributes

An Attribute is used by rolling against one of its three **Targets**, chosen by how difficult the task is (see Difficulty). The Targets are written on the sheet, so nothing is calculated at the table:

- **Trivial** — Score $\times 1\frac{1}{2}$, to a ceiling of 99.
- **Routine** — the Score itself, to a ceiling of 75.
- **Challenging** — half the Score, to a ceiling of 50.

Roll **d100** and come in at or under the Target, after any Defense or circumstance is subtracted.

The d100 form has no passive value

There is no taking a Score without rolling in this Ruleset. In a roll-under system the Score is the roll, so there is no separate standing number to compare against. That work is done by the **Trivial** tier, which is what a GM calls for when a task is well within a character's competence. A Trivial Target is high enough that the outcome is rarely in doubt, without ever being certain.

A GM who does not want a roll at all should simply not call for one. That has always been true in every Ruleset, and it remains the right answer when nothing is at stake.

EXAMPLE

A character with ● Acuity 60 enters a room. Their sheet reads **Trivial 90 / Routine 60 / Challenging 30**. Glancing around for anything obvious, the GM calls it Trivial: they roll against 90 and will almost certainly notice it. Searching carefully for something well hidden, the GM calls it Challenging: they roll against 30, and the odds are against them.

Social Actions and the Four Attributes

Frontiers uses four Attributes and no dedicated social Attribute. A social action is governed by whichever Attribute fits the **approach**:

- ♦ **Vigor** to intimidate through physical menace.
- ☆ **Deft** or ● **Acuity** to charm, misdirect, or outmaneuver.
- △ **Resolve** to persuade or hold a room through conviction.

Trained social ability is applied on top as **Favor** from a Voice Skill (see Skills), so a naturally forceful character and a trained negotiator can reach the same goal by different routes. A system that wants an innate social Attribute can add one as a Variant (see the Design Documentation).

Derivatives

A **Derivative** is any value calculated from one or more Attributes. Together they make up most of a character sheet. Some are used only in a particular mode of play and are covered in full in a later section, but every one of them originates here, from the four Attributes.

Because Ranks can now be negative, any Derivative that could otherwise fall to zero or below is given a floor. Where a floor applies, it is listed.

The Derivatives Table

Icon	Derivative	Value	Minimum
+	Maximum HP	5 + ♦ Vigor Rank; +2 + half (♦ Vigor + △ Resolve Ranks) per Level	1
⊗	Maximum Wounds	3 +1 per 2 points of ♦ Vigor Rank (max 5)	1
⊖	Maximum Fatigue	3 +1 per 2 points of △ Resolve Rank (max 5)	1
⇒	Movement	5 + ☆ Deft Rank (units)	1
➤	Initiative	flat 10 + ☆ Deft Rank, or roll 1d10 + 5 + ☆ Deft Rank	—
□	Physical Defense	3 × half (♦ Vigor + ☆ Deft Ranks), round up	0
□	Mental Defense	3 × half (● Acuity + △ Resolve Ranks), round up	0
	Additional Skills	+1 per 2 points of ● Acuity Rank	see note

Health and Pressure Pools

+ **Maximum HP**, **⊗ Maximum Wounds**, and **⊖ Maximum Fatigue** set how much a character can endure before harm forces a consequence. **+** HP is the only one that grows with Level. It leans on **❖ Vigor**, the body's durability, with **△ Resolve** lending the will to keep standing. The **⊗ Wound** and **⊖ Fatigue** tracks stay short at every level of play, running from 1 to 5. How these pools fill, what each track does as it fills, and how they recover is covered in *Vitality, Exertion & Death*. All three are Anchored (see below).

⇒ Movement and ► Initiative

⇒ **Movement** is how far a character can travel under their own power in a single unit of Encounter time, measured in **□ Units** (see *Flow of Play*). It equals 5 plus the character's **☆ Deft Rank**.

► **Initiative** sets turn order in an Encounter. At the start of an Encounter, each participant chooses one of two methods, and that choice holds for the entire Encounter:

- **Flat.** Take a standing value of $10 + ☆ \text{ Deft Rank}$.
- **Rolled.** Roll **1d10** (either die of the percentile pair) $+ 5 + ☆ \text{ Deft Rank}$.

Higher ► Initiative acts first. On a tie, the higher **☆ Deft Rank** acts first; if still tied, those participants roll off.

The two methods trade certainty against range. Neither is the better choice in general. Rolling averages slightly lower than the flat value, 9.5 against 10, but can land anywhere from near-last to first; taking the flat value gives up that range for a result you can plan around. Choose the flat value when your position matters more than your odds.

Defense

Every character has two Defenses, each a standing value that a hostile action must overcome:

- **□ Physical Defense** equals $3 \times$ half the sum of the character's **❖ Vigor** and **☆ Deft Ranks**, rounding up. Never less than 0.
- **□ Mental Defense** equals $3 \times$ half the sum of the character's **● Acuity** and **△ Resolve Ranks**, rounding up. Never less than 0.

Each Defense draws on **both** Ranks of its pair, because there is more than one way to avoid harm: a blow is turned aside by toughness and speed together, an assault on the mind by sharpness and will. A character balanced across a pair defends better than one who leans on a single stat.

A Defense is never rolled against directly. It is **subtracted from an attacker's Target**, so it is stated as a plain quantity of percentage points: an untrained target subtracts nothing, a capable one subtracts 6 to 12, and an exceptional one as much as 30. A defender is a visible cost the attacker weighs before committing the dice. Each **⊗ Wound** a character is carrying lowers both Defenses, and so makes them easier to hit (see *Vitality, Exertion & Death*). Both Defenses are Live (see below).

Skills from ● Acuity

● Acuity is the one Attribute whose Rank grants a resource of its own. A character gains **one additional Skill for every 2 points of ● Acuity Rank**, rounded down, added on top of the Skills granted at character creation. A ● Acuity of 70 (Rank +4) grants 2 Skills; a ● Acuity of 75 (Rank +5) also grants 2.

A **negative** ● Acuity Rank removes Skills at the same rate. The floor of **1 Skill** applies to a character's total, not to this modifier: the modifier is free to go negative, and it is the total it produces that cannot fall below one. What Skills are, where they come from, and how many a character starts with is covered in Skills.

Live vs Anchored Derivatives

When an Attribute changes during an Encounter, some Derivatives update at once and some do not.

Live Derivatives update immediately:

- □ Physical Defense and □ Mental Defense
- ⇒ Movement
- ● Additional Skills from ● Acuity

Anchored Derivatives do not change mid-Encounter:

- + Maximum HP
- ⊗ Maximum Wounds
- ⊖ Maximum Fatigue
- ► Initiative (locked once the Encounter's order is set)

If an Attribute changes **permanently**, Anchored Derivatives recalculate at the next natural pause or at the end of the current Encounter.

Recalculation is **retroactive**. + Maximum HP accumulates an increment for every Level a character has gained, and each of those increments is recomputed at the new Rank, not only the ones gained afterward. A character who raises ♦ Vigor at Level 8 recalculates their whole + HP total as though they had held the higher Rank from the start.

Vitality, Exertion & Death

Frontiers divides what a character can withstand into two domains, each built the same way: a buffer in front of a short, serious track.

- **Vitality** is harm to the body. A buffer of + HP soaks ordinary blows, and behind it a small track of ⊗ Wounds marks lasting injury. A filled ⊗ Wound track is **death**.

- **Exertion** is the toll of pushing past your limits. A reserve of 4 Energy is spent to reach beyond yourself, and behind it a small track of ☹ Fatigue marks the strain. A filled ☹ Fatigue track is **collapse**.

Each track punishes a character more heavily as it fills, and takes them out of the fight when it maxes. The one difference is the ending, and it is the whole reason the two exist separately: ☹ **Wounds kill**, ☹ **Fatigue only downs you**. A character can be exhausted to the edge of blacking out yet in no real danger, or gravely wounded yet still sharp. Keeping the tracks apart is what makes that possible.

Vitality

Three separate tracks, mending at different speeds

+ HP	buffer, refills fastest
☹ Wounds	lasting injury, one per rest
☹ Fatigue	exhaustion, one per rest

A character can be upright and useless, or drop without ever being wounded

+ HP

+ HP is a character's buffer against harm: the grit, near-misses, and glancing blows that soak an ordinary hit before it does lasting damage. It is the only pool that grows as a character advances.

A character begins with a maximum **+ HP** of **5 + ♦ Vigor Rank**. At each new Level they gain **2 + half the sum of their ♦ Vigor and △ Resolve Ranks**, rounding up, to a minimum of 1 per Level. Durability leans on ♦ Vigor, with △ Resolve lending the will to keep standing, but neither runs away with the number.

Character	Level 1	Level 5	Level 10
♦ 50 / △ 50	5	13	23
♦ 60 / △ 60	7	23	43
♦ 70 / △ 70	9	33	63

Harm reduces **+ HP** first, and **+ HP** never drops below 0. Excess harm from a single blow is lost rather than carried over: a hit for 12 against 3 remaining **+ HP** empties the buffer and no more.

When **+ HP** reaches 0, the character takes **one ☹ Wound** and is **downed**, out of the fight and unable to act (see Unconscious). They take no further ☹ Wounds from that blow, however large it was. A downed character only gains more ☹ Wounds if something strikes them while they are down, one per instance of harm.

The amount of harm a given source deals is defined where that source lives, on the weapon, item, or ability itself; this section governs only how the pools behave.

✚ HP returns quickly with rest (see Recovery). It is meant to be spent and refilled often, so hoarding it only wastes the resource.

⊗ **Wounds**

A ⊗ **Wound** is serious injury: the harm that lands once a character has nothing left to absorb it with. Maximum ⊗ Wounds is **3**, plus 1 for every 2 points of ♦ Vigor Rank, to a maximum of **5** and never fewer than 1. The track is short at every level of play: even the toughest character is five injuries from death.

⊗ Wounds are inflicted by harm taken once ✚ HP is gone, by effects that deal them directly, and by overspending ♣ Energy (see Overspend).

Every ⊗ Wound a character is carrying degrades **both Defenses**, at −5 for each point held:

⊗ Wounds held	Penalty to □ and □ Defense
1	−5
2	−10
3	−15
4	−20
5	−25

Read the penalty at the current number of ⊗ Wounds. It does not accumulate across them, and it applies right up to a full track. A Defense reduced by ⊗ Wounds never falls below **0**: a badly wounded character stops contributing any protection, but never makes an attacker's Target better than it would be against no one at all. Because ⊗ Wounds are taken at the moment a character is downed, these penalties are usually felt in the next fight instead of the one that inflicted them: a character carries their injuries forward until they are treated.

The harm that would push a character **past** maximum ⊗ Wounds is **death**.

Exertion

⊖ **Fatigue**

⊖ **Fatigue** is depletion: the exhaustion of a body and mind pushed too hard. Maximum ⊖ Fatigue is **3**, plus 1 for every 2 points of △ Resolve Rank, to a maximum of **5** and never fewer than 1.

⊖ Fatigue is inflicted by overspending ♣ Energy (see Overspend), by strenuous exertion the GM judges taxing, and by effects that deal it directly.

Every point of $\ominus$ Fatigue applies a penalty to **every Target a character rolls against**, at -5 per point, on the same scale as $\otimes$ Wounds (-5 , -10 , -15 , and so on), read at the current number of points. Where $\otimes$ Wounds make a character easier to strike, $\ominus$ Fatigue makes them worse at everything they attempt, and it is felt by every character regardless of build.

A character at maximum $\ominus$ Fatigue is still conscious and can act, though heavily impaired. The point of $\ominus$ Fatigue that would push **past** maximum drops them **unconscious** (see Unconscious).

⚡ Energy

⚡ Energy is the reserve a character spends to reach beyond their normal limits.

Everything a character might use, an ability, a weapon, a suit of armor, an item, has a **Demand**: the level of power it operates at. Every character has a **Capacity**: the level of power they sustain without strain. **⚡ Energy** is what it costs to close the distance between the two.

$$\text{⚡ Energy cost} = \text{Demand} - \text{Capacity, minimum } 0.$$

- At or below your Capacity, a thing is **native**: using it costs no **⚡ Energy**. As a character grows, more of what they own becomes free.
- Above your Capacity, using it costs the difference. Reaching strains you.
- A thing whose Demand is beyond any Capacity a character can reach always costs **⚡ Energy**, since the gap can never close.

The rule is identical for abilities, weapons, armor, and items, so **⚡ Energy** is a single resource governing every kind of reach. An ability's cost is paid **each time it is used** (see Abilities), while equipment holds its cost **for as long as it is Equipped**, reducing available **⚡ Energy** until it is set aside (see Equipment). It is also a removable layer: strip the Demand from a thing and it simply works without a cost, leaving the rest of the engine intact.

COST EXAMPLES

A Demand 5 ability used by a Capacity 3 character costs 2 **⚡ Energy** each use; once that character reaches Capacity 5, it is native and free. A Demand 4 weapon is free in the hands of a Capacity 6 fighter, but locks 2 **⚡ Energy** away from a Capacity 2 fighter for as long as it is Equipped.

Capacity and Level

Capacity is the Level a character references when using anything with Demand. By default it is simply their total character Level. Systems with multiple progression tracks, a total level alongside class or discipline levels, define which Level a character's Capacity references. Character Level determines the size of the **⚡ Energy** pool; Capacity determines what is native.

THE POOL

A character's maximum **⚡ Energy** starts at 3 and rises by 1 at every even Level:

Character Level	Maximum ⚡ Energy
1	3

Character Level	Maximum ⚡ Energy
2 – 3	4
4 – 5	5
6 – 7	6
8 – 9	7
10	8

The reserve grows slowly on purpose. Native use is free, so ⚡ Energy is only ever spent on reaching, and a lean pool keeps that reaching a real decision rather than a habit.

THE BASE ENVELOPE

The base engine assumes a maximum character Level of **10**, a maximum ability or item Demand of **15**, so at the Level cap the widest reach is **5**. Below the cap it is wider: a Level 1 character facing a Demand 15 item has a gap of 14. Even at the ceiling, a single maximal reach (a gap of 5) costs most of an 8-point pool and cannot be repeated without overspending. A system that scales past these bounds inherits the same cost rule and sizes its own pool to its own ceiling; these numbers are the base default, and a system may raise them.

Overspend

A character may spend more ⚡ Energy than they have. There is no ceiling on how far past 0 they reach, but the shortfall is paid immediately in harm: **each point of ⚡ Energy spent below 0 becomes 1 ⊗ Wound or 1 ⊖ Fatigue, chosen point by point as it is paid.**

This is the bridge between the two domains, and the choice is a genuine tactical read. Paying in ⊖ Fatigue means you will be worse at everything but recover it with rest; paying in ⊗ Wounds means you keep your edge but move closer to death.

EXAMPLE

A Level 2 character with an empty pool activates a Demand 5 ability, costing 3. All 3 are overspent, so they take 3 points split however they like across ⊗ Wounds and ⊖ Fatigue as they pay.

Overspend needs no cap of its own: the tracks run to 5 at most, so reaching a few points past empty is often enough to drop a character on its own. Pushing that far is a gamble, and landing it is the kind of moment a table remembers.

Death

Unconscious

A character is put out of the fight in one of two ways:

- **Downed by harm.** Their **+** HP reaches 0. They take one ⊗ Wound and drop.

- **Collapsed by exhaustion.** Their $\ominus$ Fatigue would be pushed past its maximum. They are held at maximum $\ominus$ Fatigue and drop.

Either way, the character cannot take $\blacklozenge$ Actions or Reactions, cannot move under their own power, and is left at the mercy of the scene. They remain vulnerable: harm inflicted on a character who is already down adds a $\otimes$ Wound for each instance, and enough of those will kill them.

A character downed by harm returns as soon as any $\blackplus$ HP is restored. A character collapsed by exhaustion returns the moment their $\ominus$ Fatigue falls below maximum, which happens naturally as they rest (see Recovery). In the middle of an Encounter, where no meaningful time passes, both usually mean out for the fight unless something restores them directly.

Mid-Encounter revival is left to the system, by design

The base engine defines no default way to bring an ally back mid-fight, because how forgiving a game is at that moment is one of its most defining choices. A system should supply the means it wants, commonly one of two shapes: a **Stabilize action**, where a character spends $\blacklozenge$ Action Points to restore a downed ally's $\blackplus$ HP, or a **recovery ability**, an authored heal that returns $\blackplus$ HP or lifts $\ominus$ Fatigue. With neither in play, a downed character stays down until the Encounter ends, which is a valid, lethal default a system may keep.

Death

A character **dies** when $\otimes$ Wounds would be pushed past their maximum. There is no dying timer and no bleeding out: the maximum $\otimes$ Wound track is itself the edge, the point where a character fights on gravely exposed, and the wound beyond it is the end.

Recovery

Every pool refills, but by different means and at different speeds. Recovery happens during rest, an $\clubsuit$ Activity taken in an Endeavor (see Flow of Play), in a safe place free of pressure.

For each uninterrupted **hour** of rest:

- Regain 1 $\blacklozenge$ Energy.

For each **3 uninterrupted hours** of rest:

- Regain half of maximum $\blackplus$ HP. This stacks at rolling intervals, so 3 hours restores 50% and 6 hours restores 100%. $\blackplus$ HP never exceeds its maximum.
- Remove one $\otimes$ Wound **or** one $\ominus$ Fatigue. Not both in the same interval.

If rest is broken by combat, immediate danger, or significant exertion, only the current unfinished interval's progress is lost. Anything already recovered in completed intervals stays, and the count toward the next interval restarts from zero.

The three pools mend at different speeds, which is what keeps them distinct. $\blacklozenge$ Energy returns steadily, hour by hour. $\blackplus$ HP is fully restored after six hours. $\otimes$ Wounds and $\ominus$ Fatigue come off one point per three hours, and only one or the other in that span, so a character carrying a full $\otimes$ Wound

track needs most of a day to clear it and longer still if they are exhausted as well. Injuries follow you out of the fight; the buffer does not.

A character collapsed by exhaustion wakes at the first interval that removes a $\ominus$ Fatigue, three hours in. A character downed by harm returns as soon as any $\oplus$ HP is restored.

Beyond rest, remedies and restorative effects can return any pool directly; what each one restores is defined where it lives, in Equipment or in an ability's own entry.

Skills

A **Skill** is trained expertise in a specific field. Where an Attribute measures what a character is innately capable of, a Skill measures what they have learned to do well.

Skills are **binary**. A character either has a Skill or does not; there are no ranks, degrees, or partial training. A character without a relevant Skill may still attempt anything a trained character can. Training never decides whether an action is possible, only how well it is likely to go.

What a Skill Does

When a Skill applies to a roll, it grants **Favor**, which is the whole of its mechanical effect.

The player chooses which die that Favor applies to:

- **On the tens die**, improving the odds the action succeeds at all.
- **On the Gradient Die**, pushing the result toward a stronger outcome.

The choice is declared before that die is rolled, and it matters. A character attempting something difficult usually wants the better odds; a character who cares more about how the result lands may put their training toward the Gradient instead. Expertise can buy either confidence or impact, but it must be chosen before the dice answer.

Only **one Skill** applies to a given roll. If several could plausibly fit, the player picks which one. Favor from other sources, circumstance, abilities, equipment, still stacks with it, up to the usual maximum of 3 (see The Resolution System).

Skills never gate an action

Lacking a Skill is never a reason a character cannot try something. A character with no training in medicine can still attempt to treat a wound; they simply do so without Favor. Systems that want hard requirements, where an untrained character genuinely cannot pilot the ship, should impose them through the fiction or through their own rules rather than through Skills.

Skill Categories

Every Skill belongs to one of three categories:

Category	Covers
Body	Physical training: athletics, stealth, craft, endurance, handling tools and vehicles
Mind	Mental training: knowledge, investigation, medicine, technical and academic fields
Voice	Social training: persuasion, deception, intimidation, performance, negotiation

Categories carry **no modifier of their own**. They are how Skills are organized, and more importantly how the rest of the engine addresses them.

Because every Skill sits in a category, abilities, equipment, and effects can target a whole group at once instead of naming Skills one by one:

- Favor on all Voice rolls for the scene.
- While this is worn, ignore Disfavor on Body rolls.
- Choose a Mind Skill; you have Favor on it permanently.

This is what the categories are for. They give designers three stable, genre-neutral handles to hook effects onto, in a system where the Skill list itself is defined per system.

Gaining Skills

A character's Skills come from two places:

1. **Character creation.** Origin, Archetype, and any other identity layer a system defines can grant Skills (see Characters).
2. **Their ● Acuity Rank.** A character gains one additional Skill for every **2 points** of ● Acuity Rank, rounded down. A **negative ● Acuity Rank** removes Skills at the same rate, reflecting a character who never absorbed their training.

A character always has at least **1 Skill**, no matter how far a negative ● Acuity Rank reaches. When ● Acuity would reduce a character below one Skill, the player chooses which single Skill remains.

Skills may also be gained during play through Advancement (see Characters).

Setting a Starting Count

This Ruleset does not fix how many Skills a character begins with. That number sets how broadly capable characters feel, and it belongs to the system being built and not to the engine.

As a reference point, a starting total of around **3 Skills before ● Acuity** suits most systems. At that baseline:

● Acuity Score	Rank	Skills from ● Acuity	Starting Skills
30	−4	−2	1
40	−2	−1	2
50	+0	+0	3
70	+4	+2	5
90	+8	+4	7
100	+10	+5	8

Adjust the base up for competent, broadly trained casts, or down for a grittier game where expertise is rare.

The recommended method is to distribute Skills through the identity layers rather than as a flat pool, so a character might receive one Skill from their Origin and two from their Archetype rather than picking three freely. This ties training to who the character is. Other approaches work equally well: a flat pool the player spends freely, a fixed list attached to each Archetype, or Skills earned entirely through play. How Skills are handed out is the designer's call.

When a Skill Applies

A Skill applies when the character's approach relies on that training: using a practiced technique, drawing on specialized knowledge, or doing something they have rehearsed.

A Skill does not apply merely because it is topically related. A character with the Medicine Skill has Favor when treating an injury, not when shoving a wounded ally out of the way.

The GM decides whether a Skill applies. When it is arguable, the usual answer is to allow it: Favor is a modest benefit, and rewarding a player for engaging with their character's training is generally worth more than a strict ruling.

Example Skill Lists

Skill lists are **defined by the system**. The engine supplies none of its own. A list should be short enough to hold in mind and specific enough that having a Skill means something. Somewhere between 12 and 24 Skills suits most systems.

The lists below are examples of how a list might look across different genres, and none of them are canonical.

Fantasy

Category	Example Skills
Body	Athletics, Stealth, Survival, Riding, Smithing, Sleight of Hand
Mind	Arcana, Medicine, History, Nature, Investigation, Religion
Voice	Persuasion, Deception, Intimidation, Performance, Command

Science Fiction

Category	Example Skills
Body	Athletics, Stealth, Zero-G Maneuvering, Piloting, Engineering, Demolitions
Mind	Computers, Xenology, Medicine, Astrogation, Sensors, Cryptography
Voice	Negotiation, Deception, Intimidation, Bureaucracy, Broadcast

Horror

Category	Example Skills
Body	Athletics, Stealth, Drive, Firearms, First Aid, Scavenging
Mind	Occult, Research, Medicine, Forensics, Psychology, Notice
Voice	Persuasion, Deception, Intimidation, Comfort, Interrogation

Western

Category	Example Skills
Body	Riding, Shooting, Brawling, Survival, Ranching, Tracking
Mind	Frontier Lore, Medicine, Law, Prospecting, Gambling
Voice	Persuasion, Deception, Intimidation, Bartering, Oratory

Building a List

When assembling a Skill list for a system:

- **Spread it across all three categories.** A list weighted heavily toward Body will make ● Acuity and ▲ Resolve builds feel unsupported.
- **Name Skills for what characters do**, not for what they know. Investigation earns its place in play more readily than Anthropology.
- **Keep each Skill broad enough to come up regularly.** If a Skill applies less than once a session, it is too narrow.

- **Let genre drive the list.** The presence of Astrogration or Prospecting tells a player what kind of game they are in before they read a single rule.
-

Abilities

An **Ability** is any discrete, repeatable thing a character can do that the base rules do not already cover: a trained maneuver, a spell, a piece of technology, a supernatural talent. Abilities are where a system says what makes its characters special.

The engine defines how Abilities are used and priced. What Abilities exist, what they do, and how characters get them is defined by the system built on top of it.

Demand and Capacity

Every Ability has a **Demand**: the level of power it operates at.

Every character has a **Capacity**: the level of power they can sustain without strain.

Capacity is the Level a character references when using anything with Demand. **By default it is simply their total character Level.** It carries its own name because systems with multiple progression tracks, a total level alongside class or discipline levels, need to define which Level a character operates at. Character Level sets the size of the ♣ Energy pool; Capacity sets what is native. By default these are the same number.

The relationship between the two is the entire cost rule:

♣ Energy cost = Demand – Capacity, minimum 0.

- An Ability whose Demand is at or below a character's Capacity is **native**. Using it costs no ♣ Energy.
- An Ability whose Demand exceeds a character's Capacity costs ♣ Energy equal to the difference, **every time it is used**.
- An Ability may have a Demand beyond any Capacity a character can reach. Such an Ability always costs ♣ Energy, no matter how far its user advances.

A character may spend more ♣ Energy than they have. Each point spent below 0 becomes 1 ⊗ Wound or 1 ⊖ Fatigue, chosen as it is paid (see Vitality, Exertion & Death).

EXAMPLES

A Demand 5 Ability used by a Capacity 3 character costs 2 ♣ Energy per use. When that character reaches Capacity 5, it becomes native and free. A Demand 12 Ability, in a system where Capacity caps at 10, costs at least 2 ♣ Energy forever.

Equipment uses the same rule, with one difference: because gear is worn or wielded continuously and not used once, its cost is an ongoing drain rather than a per-use payment (see Equipment).

Demand is a removable layer

Demand is a cost layer sitting on top of an Ability. Strip it away and the Ability still works exactly as written, simply without an ⚡ Energy cost, so a system that does not want an ⚡ Energy economy can remove Demand entirely and lose nothing else.

What an Ability Costs to Use

An Ability may carry up to three costs, and most carry only the first:

Cost	What it is
✦ Action Points	What the Ability costs on a character's Turn in an Encounter. Most cost 1 ✦AP.
⚡ Energy	Demand minus Capacity, as above. Native Abilities cost none.
Anything else	A cost the Ability itself defines: an item consumed, a ☹ Fatigue taken, a condition accepted.

An Ability may instead be a ✧ Free Action, a Reaction, or a ✱ Activity usable only outside an Encounter (see Flow of Play). Abilities with no action cost at all are **passive**: they apply continuously and are never activated.

What an Ability Targets

Every Ability that resolves against something should say what it resolves against. Abilities in Frontiers commonly target:

- **An Attribute.** The user rolls, or the target resists, using ♦ Vigor, ☆ Deft, ● Acuity, or △ Resolve.
- **A Defense.** The Ability is opposed by □ Physical Defense or □ Mental Defense.
- **A Skill category.** The Ability grants, hinders, or applies to Body, Mind, or Voice rolls as a group (see Skills).
- **Nothing.** The Ability simply takes effect, with no roll involved.

Targeting a Skill category is what lets an Ability read "gain Favor on all Voice rolls this scene" rather than naming a dozen individual Skills. Because categories are genre-neutral, an Ability written this way works in any system regardless of its Skill list.

A Note on Damage Output

The engine defines no attacks and no damage values. What a weapon or ability deals belongs to the system built on top, and combat specifics belong to a Module. This section does not describe an attack. It describes only the **output a character should be capable of**, so a designer knows what their own numbers need to add up to.

The anchor is the average character and not any single ability:

A competent, average character, using **1 ♦AP** and a **native** ability (one that costs no ♣ Energy), should be able to output roughly **10-15% of an even target's + HP** per successful use, landing **50-75%** of the time against that target.

The hit band emerges from a character's Target against typical □ Defenses rather than being fixed by rule. An attacker rolling at a **Routine** Target of 70, less a defender's Defense of 6 to 12, lands between 58% and 64%, which sits inside the band. A ♦ Vigor 50 attacker lands below it against any Defense worth having, and that is this Ruleset's temperament rather than a fault: here, reaching the band is part of what competence buys. A designer who wants attacks to connect more or less often adjusts either side, and the band moves with it.

Everything else follows from that one expectation:

- **The rhythm it produces.** With 3 ♦AP a turn, a character makes about three such attempts, of which one to two land, for roughly a quarter to a third of a target's + HP each round. An even fight resolves in **about three to four rounds at the top of the band, and closer to six or seven at the bottom**. That spread is the band doing its job: the anchor is a range, not a single setting, and where a designer sits inside it decides how long a fight runs. If you want the shorter, punchier pace, build toward the upper end of both accuracy and damage; the lower end produces a longer, more attritional fight. A designer who wants grittier or faster play than either simply moves the target.

The band describes output, not balance

This anchor covers what a single native ability should output. It does not tell you how many abilities a character should have, how often they should be able to lean on the strong ones, or what a full loadout adds up to across a fight. That question, how much access a character has to their best options, is the one this engine deliberately leaves to the system built on it, and it is the one most likely to break a game if it goes unanswered.

The precision of the numbers above can make it look as though more of the work is done than it is. Hitting the band exactly and then handing every character unrestricted access to a dozen abilities will still produce something unbalanced. See the Starter Sets, which settle that question for a specific genre and are the clearest worked examples of what a complete answer looks like.

- **Free output is the baseline; ♣ Energy buys above it.** The band describes what a character can do at no ♣ Energy cost. Spending ♣ Energy, or reaching for a higher-Demand option, is how a character exceeds it. Native output is the floor a designer balances around; reaching is the spike on top.
- **Scale output to the pool, not to a fixed die.** Because + HP grows with Level, a fixed number falls behind. Hold the percentage, and let whatever produces the damage grow to match it as characters advance.
- **The band flexes by target.** The 10-15% figure assumes an even target. A fragile foe takes a larger share and drops faster; a bulky one takes less and is meant to. Building a boss is deliberately pushing + HP and □ Physical Defense up so the same character output takes longer to matter.

- **The "even target" is a median. No character at the table will match it exactly.** A real party brackets that median widely, since **+** HP can differ severalfold between a fragile build and a durable one. A single hit tuned to the median is a large share of a fragile character's pool and a small share of a tough one's, so when statting adversaries, treat spike damage as something the party's most fragile member feels first.
- **Not all output is damage.** Control, movement, and utility spend the same **♦**AP and **⚡** Energy without touching **+** HP. The band is a reference for what damage should feel like, never a floor every ability must clear.

How any of this is actually produced, weapons, dice, attack abilities, is defined by the system and its Modules.

Ability Anatomy

An Ability should read like a card: everything needed to use it, visible at a glance, in the same order every time.

(♦♦) Ability Name How it is used · **⚡** Demand X · Category (opt.)

Trigger or requirement, if any: what the Ability actually does.
Any conditional clause that changes the effect.

The elements, in order:

Element	What it says
(♦)	The ♦ Action Point cost, one icon per point. (♦) is 1, (♦♦) is 2, (♦♦♦) is 3. (◇) is a ◇ Free Action. (—) means it costs no ♦AP at all.
Name	What the Ability is called.
How it is used	♦ Action, ◇ Free Action, Reaction, ✱ Activity, or Passive. If it can only be used in certain modes, say so here.
⚡ Demand	The Ability's Demand, omitted entirely if it has none.
Category	Body, Mind, or Voice, when the Ability interacts with Skills.
Trigger	The condition that must be true, for Reactions and conditional Abilities.
Effect	What happens, in plain language.

Marking Optional and Required

A Category tag alone does not say whether a Skill is needed. Two notations make it explicit:

- **(opt.)** — the Ability works for anyone. Having a Skill in that Category improves it.
- **(req.)** — the Ability cannot be used without a Skill in that Category.

Most Abilities should be (opt.). Requiring a Skill narrows who can use an Ability, which is occasionally the point but should be a deliberate decision instead of a default.

Examples

A direct Ability, used on a character's own Turn:

(♠) Driving Blow ♠ Action · ½ Demand 2 · Body (opt.)

Make a Resolution roll against a target's □ Physical Defense within 1 □ Unit. On a success, the target is pushed 1 □ Unit directly away from you.

If you have a Body Skill covering close combat, the push is 2 □ Units instead.

A Reaction, used outside your own Turn:

(♠♠) Shatter Stance Reaction · ½ Demand 4 · Body (opt.)

When an adjacent creature misses you with an attack: the attacker is knocked prone and loses 2 □ Units of ⇒ Movement on their next Turn.

If you have the Athletics Body Skill, they also take a ⊖ Fatigue.

An Ability used outside an Encounter entirely:

(—) Read the Room * Activity · Exploration only · ½ Demand 3 · Mind (req.)

Spend a few minutes observing a space and the people in it. The GM answers one question truthfully about what happened here recently, or about what someone present is concealing.

Once the question is answered, this Ability cannot be used again in the same location until the situation meaningfully changes.

An Ability performed across a long span of time:

(—) Field Repair * Activity · Endeavor only · ½ Demand 1 · Body (opt.)

Over an hour of work, restore a damaged piece of equipment to working order, or remove one ⊗ Wound from a construct or vehicle.

If you have a Body Skill covering craft or repair, you may work on two items in the same hour.

Keep effects short and declarative. An Ability a player cannot read aloud in one breath is too complicated for the table.

Balancing Abilities

The Demand rule has a consequence designers must plan around.

Because cost is Demand minus Capacity, **Abilities become free as characters grow into them.** A character who reaches Capacity 5 uses every Demand 5 Ability they own at no cost, forever, as often as they can pay the ♦ Action Points. That is deliberate: it is what makes advancement feel like genuine mastery instead of a treadmill of rising prices.

It also means **⚡ Energy alone will not balance your Abilities.** ⚡ Energy prices reaching beyond yourself. It does nothing to limit what a character does within their own tier.

This Ruleset does not resolve that for you, and that is on purpose. How many Abilities a character can hold, how they acquire them, and whether their use is limited are the decisions that most define how a system actually plays. They belong to the designer, not to the engine.

What follows are three approaches that work well in Frontiers. Use one, combine them, or build something else.

Approach One: Loadout

Limit how many Abilities are available at once.

A character may know any number of Abilities but can only keep a set number **prepared**. Only prepared Abilities can be used. Swapping the set takes time, typically during an Endeavor or a rest.

A character with four prepared slots and a dozen known Abilities decides before every venture: what do I expect to need today? Free native Abilities stay balanced under this approach, because a player may use a prepared Ability as often as they like but has to commit to the set before setting out, which limits how much any one Ability can be leaned on.

- **Good for:** tactical play, preparation as a gameplay loop, characters who feel distinct session to session.
- **Tune with:** the number of slots, how often swapping is allowed, and whether high-Demand Abilities take more than one slot.

Approach Two: Resource

Limit how often Abilities are used, with a spendable pool.

Every Ability carries a cost paid from a pool that refreshes on a schedule: per Encounter, per rest, or per day. This is the familiar shape of mana, focus, or spell points. It layers cleanly on top of ⚡ Energy, which continues to price reaching beyond Capacity.

- **Good for:** attrition-driven play, resource management, games where pacing across a day matters.
- **Tune with:** pool size, refresh rate, and whether costs scale with Demand.

Approach Three: Conditions

Limit when Abilities can be used, with triggers instead of counters.

Abilities are always available but each requires a situation to be true: only while $\ominus$ Fatigued, only after an ally has already struck the target, only once per target, only while a piece of equipment is Activated. There is no pool and no slot count. Availability is driven entirely by the state of the fiction.

- **Good for:** teamwork and combo play, cinematic escalation, systems that want no bookkeeping at all.
- **Tune with:** how narrow the conditions are, and how much a character can influence whether they trigger.

What I Recommend

LOADOUT

It fits Frontiers better than the alternatives, for three reasons:

- It does not add a second currency. $\hookleftarrow$ Energy already governs cost, and a loadout governs breadth, so the two never overlap or compete.
- It preserves the free-native rule that makes advancement feel like mastery, while still keeping power in check.
- It creates decisions before play as well as during it, which pairs naturally with the Endeavor mode.

That said, Frontiers is modular by design. If a resource pool or a condition-driven system suits the game being built, use it. Nothing elsewhere in the engine depends on this choice.

Equipment

Equipment is everything a character carries, wears, or wields. The engine defines how items are held, priced, and used. What items exist in a system, and what they do, belongs to the system built on top of it.

Equipment in Frontiers follows one guiding idea: **an item should widen what a character can do before it adds anything to track.** A weapon does not simply sit in an inventory waiting for a generic "attack" rule to reference it. The weapon carries its own Actions. Picking up a different sword changes what a character can do.

Item States

Every item is in exactly one of four states.

State	What it means
Stowed	Packed, slung, or carried. The item costs Slots and cannot be used until it is Equipped.

State	What it means
Equipped	Worn or held, ready to use. Equipped items cost no Slots. A character can only Equip what their body allows: what fits in their hands, and what can be worn at once.
Active	The item is currently doing its work: a lantern burning, a shield powered on, a dose being swallowed. Most items return to Equipped when finished. A consumable is used up and leaves the character's possession entirely.
Removed	The item is out of reach for now: dropped, disarmed, knocked away, confiscated, or left behind. It costs no Slots and grants nothing while Removed, but it still exists and can be recovered. An item that is destroyed outright is simply gone, and leaves the sheet.

Moving between states is quick. Equipping a Stowed item, or Stowing an Equipped one, is the **✧ Interact** Free Action in an Encounter, and takes no meaningful time outside one. Because all item handling shares that one **✧ Free Action**, only a single Interact is possible per Turn (see Flow of Play). Putting on or removing armor is an **✱ Activity**.

Going Active is an ✧ Interact too, unless the item says otherwise. Lighting a lantern, powering a shield, uncapping a dose: an item usually states on its own card what activating it costs, and where it does, that text wins. Where it says nothing, treat entering or leaving the Active state as the same **✧ Interact** that Equipping uses. It therefore competes with drawing and stowing for the one handling action a Turn allows, which is the point: a character cannot draw a blade and light a torch in the same breath.

Slots

Carrying capacity is measured in **Slots**. Slots are not weight, and are not derived from any Attribute. They measure how much a character can practically distribute about their person, and every Attribute already governs two things without carrying a third job.

Every character has a flat number of Slots. **10 is a reasonable default**, adjusted by the system for the kind of game being run.

Each item costs a number of Slots while **Stowed**:

Item	Typical Slots
Small item, tool, or dose	1
Weapon	1 – 2
Armor	3 – 4
Large or awkward gear	4+
Negligible items	0

EQUIPPED ITEMS COST NO SLOTS

A sword in hand and armor on the body are being worn, and worn gear costs nothing to carry, so Slots track the pack rather than the person.

Negligible Items

Anything too minor to matter costs **0 Slots and is not tracked individually**. Rations for a day, a coil of cord, a handful of coins, a torch: these are assumed unless a system says otherwise.

Ammunition is handled the same way. A quiver of arrows or a magazine of rounds is a single item. Individual rounds are never counted. Rather than tracking individual shots, a system may simply rule that ammunition is **Consumed after an Encounter in which it saw heavy use**. Counting arrows one at a time adds bookkeeping without adding decisions.

⚡ Demand on Equipment

Items carry **Demand** exactly as Abilities do (see Abilities). A character's **Capacity** is the level of power they sustain without strain, and an item pitched above it costs ⚡ Energy.

The difference is that gear is not used once and released. It is worn or wielded continuously, so its cost is **an ongoing drain rather than a per-use payment**:

While an item with Demand above your Capacity is Equipped, your ⚡ Energy maximum is reduced by the difference.

The reduction applies to the **maximum**, not to points already spent: the pool is simply smaller, and spending and recovery work normally against the smaller cap. A character's ⚡ Energy maximum can never be reduced below **1** this way, which is why gear whose combined drain would take them below 1 cannot be Equipped at all.

Unlike an ability, a held drain cannot push a character into Overspend. A character can only Equip gear whose combined drain leaves them at **1 ⚡ Energy or more**. Gear that would drop available ⚡ Energy below 1 simply cannot be brought online until something else is set aside or the character's Capacity rises to afford it.

Rather than being spent repeatedly, the cost is held. A Capacity 1 character wearing Demand 3 armor has 2 ⚡ Energy locked away for as long as they wear it: with a pool of 3, they operate on 1 until they take the armor off, at which point it returns immediately.

EXAMPLES

A Demand 2 blade is free to a Capacity 4 fighter, and locks 1 ⚡ Energy away from a Capacity 1 novice. A Demand 5 powered suit worn by a Capacity 2 character reduces their available ⚡ Energy by 3, which for most characters at that level is nearly everything they have.

This is what makes over-gearing feel right. A character can take up equipment far beyond them. They simply operate diminished while they do, and gear that outclasses its wielder is a burden as much as a prize.

Demand is optional here too

As with Abilities, Demand is a removable layer. An item stripped of its Demand works exactly as written, at no ⚡ Energy cost. A system that wants gear to be freely usable can simply not assign Demand to it.

Items Carry Their Own Actions

There is no generic "attack" Action in Frontiers. **Weapons carry their attacks, and items carry their uses.** An Action granted by an item is written in the same format as an Ability (see Abilities) and is available only while the item is Equipped.

This does three things:

- **Weapons become distinct.** Two short swords can differ meaningfully, because each carries its own Actions instead of pointing at one shared rule.
- **Character sheets get simpler.** A player does not track a list of generic combat Actions. They read what is in their hands.
- **The system stays modular.** Because attacks live on items, a designer who replaces the Equipment rules replaces the combat Actions along with them. Nothing else in the engine breaks.

Unarmed and improvised options are assumed rather than printed. A character can always strike with what they have; a system defines how well.

Item Anatomy

Items are written like Abilities, as a card that can be read at a glance. Any Actions the item grants are listed beneath it in the standard Ability format.

(Slots) Item Name Type · ♣ Demand X · any standing properties

Any passive effect the item has while Equipped.

Followed by its Actions, if it has any.

Examples

A weapon carrying two different attacks:

(2) Broad Blade Weapon · ♣ Demand 2 · Reach 1 □

(♠) Cleave ♠ Action · Body (opt.) Resolution roll against a target's □ Physical Defense within reach. On a success, deal the weapon's damage.

(♠♠) Sweeping Cut ♠ Action · Body (opt.) Resolution roll against the □ Physical Defense of every adjacent target. On a success, deal the weapon's damage. On a Negative Gradient, you are left off balance and lose your Reaction until your next Turn.

Armor, which is worn instead of used:

(3) Reinforced Plate Armor · 4 Demand 3

While Equipped, gain +2 to □ Physical Defense and Disfavor on Body rolls requiring stealth or agility.

Donning or removing this armor is an ✱ Activity.

A consumable:

(1) Restorative Draught Consumable · 4 Demand 0

(✦) **Drink** ✦ Action Regain half of your maximum + HP, then remove one ⊖ Fatigue. The draught is Consumed.

A tool that grants an out-of-Encounter Action:

(1) Surveyor's Kit Tool · 4 Demand 1

(—) **Chart the Ground** ✱ Activity · Exploration or Endeavor · Mind (opt.) Spend an hour mapping the surrounding area. The GM reveals one significant feature, route, or hazard within it. If you have a Mind Skill covering navigation or survey work, you learn two instead.

Building an Equipment List

Equipment lists are defined by the system. The engine supplies none. A few principles hold across genres:

- **Give items Actions as well as numbers.** An item that only modifies a statistic is inert. An item that adds a new option to a character's turn is a choice.
- **Use Demand to mark aspiration.** Gear a character cannot yet use freely is one of the clearest ways to show them what growth looks like.
- **Keep Slot costs coarse.** 1, 2, 3, 4. Fractional or finely-tuned encumbrance is exactly the complexity this system exists to avoid.
- **Do not track what does not matter.** If a resource never runs out at a dramatic moment, it should not be counted at all.
- **Let categories emerge from the genre.** Weapons, armor, and consumables suit most systems, but a game may want Vehicles, Implements, Relics, or nothing but a single flat list.

Characters

A character in Frontiers is built from three things: **who they are** (the identity layers), **what they can do** (the numbers every other section defines), and **what they want** (their Pursuit, which is also how they grow).

This section assembles the whole engine into a person. It defines the identity layers, walks through creation step by step, and covers Advancement.

The Identity Layers

Every character is defined by two identity layers, each answering one question:

Layer	Question	Examples of what a system might make it
Origin	What are you, and where do you come from?	Species and upbringing; a district and a chassis; a homeland; a bloodline
Archetype	What do you do?	A class; a trade; a profession; a calling

Both layers are **umbrellas**: name whatever pieces your setting needs under each. A fantasy system might split Origin into a Species and an Upbringing; a cyberpunk one into a Chassis and a District. A system might make Archetype a rigid class or a loose profession. The engine only requires that the two questions get answered.

Name the pieces rather than the umbrella

A system that splits a layer into parts should name the parts. If Origin divides into Species and Background, character sheets and rules should say Species and Background, reserving Origin for the category itself. Using one word for both the umbrella and a piece of it is how identity systems become confusing.

Mechanically, the identity layers are where a system attaches starting benefits: Skills, Abilities, Attribute adjustments, equipment, and anything else a designer wants identity to carry. The recommended method is for **Origin to grant Skills and standing traits** (what your nature and history taught you) and **Archetype to grant Abilities and its Level features** (what your practice lets you do), but other splits work equally well, and how much weight each layer carries is the designer's call.

Starting Values

A new character begins with the values the rest of this Ruleset has already defined, gathered here in one place:

Value	Starting point	Defined in
Attribute Scores	50 each, adjusted by creation choices	Attributes & Derivatives
+ Maximum HP	5 + ♦ Vigor Rank (grows each Level)	Attributes & Derivatives
⊗ Maximum Wounds	3, +1 per 2 points of ♦ Vigor Rank (1-5)	Attributes & Derivatives
⊖ Maximum Fatigue	3, +1 per 2 points of △ Resolve Rank (1-5)	Attributes & Derivatives

Value	Starting point	Defined in
⇒ Movement	5 + ☆ Deft Rank	Attributes & Derivatives
□ / □ Defenses	3 × half the sum of the pair's Ranks, round up	Attributes & Derivatives
⚡ Energy	3 (Level 1)	Vitality, Exertion & Death
Capacity	1 (equal to Level, by default)	Abilities
Skills	From identity layers, ±1 per 2 points of ● Acuity Rank (min 1)	Skills
Slots	10	Equipment

How players adjust Attribute Scores at creation, point spending, arrays, or fixed by identity, is a system decision. The recommended method for a baseline game: **start every Score at 50, then distribute +30 points and take –10 points**, no Score above 70 or below 40 at creation. That spread produces capable, uneven characters without extremes, and gives a starting character a spread of thirty percentage points between their strongest and weakest Attribute.

Creating a Character

The recommended order of creation. Each step leans only on the ones before it:

1. **Concept.** Decide who the character is in a sentence.
2. **Origin.** Choose it, and take what it grants.
3. **Archetype.** Choose it, and take what it grants.
4. **Attributes.** Adjust Scores from 50 using the system's method. Note each Rank $((\text{Score} - 50) \div 5)$ and write the three Targets for each Attribute.
5. **Derivatives.** Calculate + HP, ⊗ Wounds, ⊖ Fatigue, ⇒ Movement, and both Defenses.
6. **Skills.** Total the Skills from your identity layers, then add one Skill per 2 points of your ● Acuity Rank, rounded down (minimum 1 Skill).
7. **Abilities and Equipment.** Take what your layers grant and what the system offers. Note the ⚡ Energy drain of anything Equipped above your Capacity.
8. **Pursuit.** Define what your character wants (see Advancement). Write it on the sheet.

The last step matters as much as the first. A character without a Pursuit is a stat block; the Pursuit is what tells the GM which stories are yours, and it is how you grow.

Advancement

Frontiers advances characters through **Pursuits**: the goals a character and a party are actively chasing. Progress in the story is progress on the sheet.

There are two kinds, and they drive two different rewards.

Pursuit	Whose	Drives
Individual Pursuit	One character's personal goal	Marks , immediate stat growth
Party Pursuit	The group's shared goal	Levels , the larger beat

Individual Pursuits and Marks

At creation, each player defines an **Individual Pursuit** for their character: a personal goal specific enough to act on and open enough to survive the story. Earn the guild's respect. Find out who burned the archive. Master the old forms.

When a character meaningfully advances their Pursuit, the GM grants a **Mark**. A Mark is spent immediately on one of the following:

Mark	Effect
Attribute	+5 to two different Attribute Scores
Skill	Gain one Skill
Ability	Gain one Ability
Resilience	+1 maximum ⊗ Wounds or +1 maximum ⊖ Fatigue
System	A Mark type the system defines

Because ⊗ Wounds, ⊖ Fatigue, and Skills each rise every **2 points of Rank**, and a Mark raises a Score by 5 (one Rank), a single Mark does not always cross one of those thresholds. Raising a Score always improves the values built on it, and the track or Skill follows on the next Rank.

Note where the ceilings bite: once a Score passes the point where its **Trivial** and **Routine** Targets are capped, further Marks buy **Challenging**-tier work, Defense, and derived values instead of everyday accuracy. That is the intended shape of mastery in this Ruleset. The two rates are intentional, growth in what a character rolls against is continuous, growth in their capacities comes in steps.

A character may earn at most **3 Marks per Level**. A character already at 3 simply earns no further Marks until the party's next Level. There is no bank and no carryover: capped means capped.

A player may modify their Pursuit as the story reshapes it, and may rewrite it entirely at each Level. A fulfilled Pursuit should be replaced with a new one.

The Party Pursuit and Levels

The group shares a **Party Pursuit**: the goal the campaign is currently bending toward. Reach the capital before the war does. Break the siege. Get the crew paid and out alive.

When the Party Pursuit resolves, **every character gains a Level together**, and the group defines its next Pursuit. On gaining a Level, a character:

1. Increases their **Capacity** by 1 (and their Level).
2. Gains **2 + half the sum of ♦ Vigor and △ Resolve Ranks**, rounding up, maximum **+** HP (minimum 1).
3. Gains **+1 maximum ♣ Energy**, if the new Level is even.
4. Gains their **Archetype's features** for the new Level, as the system defines them.
5. May rewrite their Individual Pursuit.
6. Begins the new Level at 0 Marks.

Levels are not earned by Mark count. The party rises together when the shared story turns, so no character outpaces the table, and the group always has a reason to care about the same goal.

Pacing

The recommended tempo: an Individual Pursuit should produce roughly **one Mark per session**, and a Party Pursuit should resolve every **three sessions or so**, yielding a Level. A campaign run at that pace reaches the base envelope's Level 10 in about thirty sessions. Faster or slower is a tone decision that the rules leave entirely open.

The Sheet Shows the Table

Pursuit advancement makes the GM and the players responsible to each other, and it does so visibly.

If one character sits at 3 Marks while another sits at 0, that is not a rules failure. It means one player's story is getting table time and another's is not, and the sheet is showing it. The fix is narrative: the GM steers toward the neglected Pursuit, or the player and GM reshape a Pursuit that has drifted out of reach. A GM running *Frontiers* should treat every Individual Pursuit as a promise to spend time there, and players should chase what they wrote down.

This is a deliberate design choice. Advancement in *Frontiers* is not a private meter that fills in the background; it is a record of whose stories are being told.

Other Methods

Pursuits are the recommended method because they tie growth to the stories the table is actually telling. They are not the only method, and nothing in the engine depends on them.

- **Plain milestone.** The GM grants Marks and Levels directly at story beats, with no Pursuits defined. Identical chassis, no lens.
- **Experience points.** Characters accrue a currency from play, spending it on Marks at fixed prices, with a Level at every third. Reliable and familiar; loses the narrative tie.

Both are fully supported: everything downstream, the Mark menu, Capacity, Archetype features, works unchanged under either. Systems and tables should use whichever fits how they actually play.

Icon Reference

Every recurring value in this Ruleset carries an icon. They appear wherever the value is mentioned.

Icon	Meaning	Defined in
❖	Vigor	Attributes & Derivatives
☆	Deft	Attributes & Derivatives
●	Acuity	Attributes & Derivatives
△	Resolve	Attributes & Derivatives
+	HP	Attributes & Derivatives
⊗	Wounds	Vitality, Exertion & Death
⊖	Fatigue	Vitality, Exertion & Death
⚡	Energy	Vitality, Exertion & Death
□	Physical Defense	Attributes & Derivatives
□	Mental Defense	Attributes & Derivatives
⇒	Movement	Attributes & Derivatives
►	Initiative	Attributes & Derivatives
✦	Action / Action Point	Flow of Play
✧	Free Action	Flow of Play
✱	Activity	Flow of Play
□	Unit	Flow of Play