

Gruesøme's Arcade — Leaderboard Metrics Library (Reference)

Version: 3.1

Last updated: 2026-01-05

built by gruesøme

sig(enc,xor:0x5A,hex): 382f33362e7a38237a3d282f3f2999e2373f

Purpose

- This is the canonical list of “metric IDs” your games can submit in ARCADE:RUN_RESULT.
- Each game chooses:
 - defaultMetric (what the Skill leaderboard ranks by for that game)
 - metrics[] (what the UI can switch between)
 - direction per metric (asc/desc), and clamp (anti-cheat & sanity)
- Activity rewards are NOT based on these metrics. Activity uses paid Credits spent + active days.

(Skill = leaderboard; Activity = participation.)

Metric schema (public/arcade-metrics-library.json)

- id: stable identifier used in payloads and URLs
- label: human-readable display label
- kind:
- counter: integer “more is better” (usually)
- duration: time values (ms)
- ratio: a normalized value (often basis points)
- currency: arcade Credits (AC) style integer counters
- composite: derived (server or client computed)
- direction:
- desc: higher ranks better (score, waves, accuracy)
- asc: lower ranks better (timeMs, damageTaken)
- format:
- int: integer count
- ms: milliseconds
- bp: basis points (0–10000 = 0–100.00%)

- ac: arcade credits (integer)

Important constraints

- Clamp: The server may clamp metric values to a min/max range for safety.
- One run = one payload: submit all metrics you can in ARCADE:RUN_RESULT once, on game over.
- Game-type aware: A “defaultMetric” must match the game’s core skill expression.

Example: speedrun → timeMs (asc), defense → waves (desc), shooter → score or accuracyBp.

Metric list

ID	Label	Kind	Dir	Format	Clamp
--- --- ---: ---: ---: ---					
`score`	Score	counter	desc	int	0-1000000000
`durationMs`	Survival Time	duration	desc	ms	0-86400000
`timeMs`	Completion Time	duration	asc	ms	0-86400000
`objectiveCount`	Objectives Completed	counter	desc	int	0-1000000
`attempts`	Attempts	counter	asc	int	0-10000
`mistakes`	Mistakes	counter	asc	int	0-10000
`streak`	Max Streak	counter	desc	int	0-1000000
`accuracyBp`	Accuracy	ratio	desc	bp	0-10000
`completionBp`	Completion	ratio	desc	bp	0-10000
`difficultyTier`	Difficulty Tier	counter	desc	int	0-20
`kills`	Kills	counter	desc	int	0-10000000
`headshotBp`	Headshot Rate	ratio	desc	bp	0-10000
`damageDealt`	Damage Dealt	counter	desc	int	0-10000000000
`damageTaken`	Damage Taken	counter	asc	int	0-10000000000
`deaths`	Deaths	counter	asc	int	0-100000
`nearMissCount`	Near Misses	counter	desc	int	0-1000000
`shotsFired`	Shots Fired	counter	desc	int	0-1000000000
`shotsHit`	Shots Hit	counter	desc	int	0-1000000000
`waves`	Waves Survived	counter	desc	int	0-1000000
`enemiesLeaked`	Enemies Leaked	counter	asc	int	0-1000000
`baseHpLeft`	Base HP Left	counter	desc	int	0-1000000
`towersBuilt`	Towers Built	counter	desc	int	0-1000000
`towersSold`	Towers Sold	counter	asc	int	0-1000000

`laps`	Laps	counter	desc	int	0-10000
`bestLapMs`	Best Lap	duration	asc	ms	0-86400000
`trackResets`	Resets	counter	asc	int	0-10000
`levelsCleared`	Levels Cleared	counter	desc	int	0-1000000
`hintsUsed`	Hints Used	counter	asc	int	0-10000
`moves`	Moves	counter	asc	int	0-10000000
`perfectClears`	Perfect Clears	counter	desc	int	0-1000000
`notesHit`	Notes Hit	counter	desc	int	0-100000000
`notesMissed`	Notes Missed	counter	asc	int	0-100000000
`maxCombo`	Max Combo	counter	desc	int	0-100000000
`timingStdDevMs`	Timing Std Dev	duration	asc	ms	0-10000
`spentInRunAC`	Credits Spent In-Run	currency	asc	ac	0-1000000
`efficiency`	Efficiency	composite	desc	int	0-1000000000
`noSpendRun`	No-Spend Run	counter	desc	int	0-1
`distance`	Distance	counter	desc	int	0-1000000000
`coinsCollected`	Coins Collected	counter	desc	int	0-1000000000
`purchasesInRun`	Purchases In Run	counter	asc	int	0-1000000
`upgradesBought`	Upgrades Bought	counter	desc	int	0-1000000
`resourcesCollected`	Resources Collected	counter	desc	int	0-1000000000
`resourcesSpent`	Resources Spent	counter	asc	int	0-1000000000
`apm`	Actions Per Minute	ratio	desc	int	0-2000
`win`	Win	counter	desc	int	0-1
`placement`	Placement	counter	asc	int	1-1000000
`assists`	Assists	counter	desc	int	0-1000000000
`revives`	Revives	counter	desc	int	0-1000000000
`healingDone`	Healing Done	counter	desc	int	0-1000000000
`buildVariety`	Build Variety	counter	desc	int	0-1000000
`secretsFound`	Secrets Found	counter	desc	int	0-1000000
`bossKills`	Boss Kills	counter	desc	int	0-1000000
`checkpointsCleared`	Checkpoints Cleared	counter	desc	int	0-1000000000
`spendPerObjectiveAC`	Spend Per Objective	currency	asc	ac	0-1000000000
`precisionScore`	Precision Score	composite	desc	int	0-1000000000
`cleanRun`	Clean Run	composite	desc	ratio	0-1

Recommended metric patterns by game type (generic)

- Endless survival / defense: durationMs (desc), waves (desc), score (desc), efficiency (desc), damageTaken (asc)
- Speedrun / time trial: timeMs (asc), deaths (asc), retries (asc), accuracyBp (desc)
- Accuracy / precision: accuracyBp (desc), shotsHit (desc), misses (asc), combo (desc)
- Puzzle / strategy: moves (asc), mistakes (asc), completionBp (desc), timeMs (asc)
- Economy-in-run games (Credits affect outcome):
- Include spentInRunAC and purchasesInRun.
- Use rankedSpendCapAC in arcade-games.json (prevents whale domination).
- Prefer “efficiency” style defaultMetric (outcome per spend) when appropriate.

Notes

- Your public/arcade-games.json should only reference metric IDs that exist in this library.
- Tools:
 - tools/validate-arcade-catalog-metrics.mjs (schema + references)
 - tools/apply-game-profile.mjs (apply genre profiles to games)