Gruesøme’s Arcade — Game Integration Guide (v1.2)

Last updated: 2026-01-05

built by gruesøme

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

What this is

This guide is for building *any* web game that plugs into Gruesøme’s Arcade economy and leaderboards.

Core rule:

- The Arcade (parent) is authoritative for Credits, membership, and payouts.
- The Game (iframe) is authoritative for gameplay + metrics *inside the run*.
- A run MUST be granted by the parent before play starts in embedded mode.

1) Embed model (iframe + postMessage)

Your game runs at:

- /public/games//index.html (or similar)

The Arcade loads it in an iframe and uses postMessage to coordinate.

Your game must:

- Detect embedded mode: window.parent !== window
- In embedded mode:
- Send ARCADE:READY on load
- Wait for ARCADE:RUN_GRANTED before starting gameplay
- Send ARCADE:RUN_RESULT once per run, on game over

2) Message protocol (ArcadeBridge)

All messages are:

- type: string
- channel: shared constant (prevents random cross-postMessage spam)
- payload: object

Required messages (game -> parent)

- ARCADE:READY { gameId }
- ARCADE:REQUEST_RUN { gameId, runType? }
- ARCADE:RUN_RESULT {

gameId,

runId,

score,

durationMs,

metrics: { [metricId]: number },

spentInRunAC?: number,   // if the game consumes Credits inside the run
}

Optional messages (game -> parent)

- ARCADE:OPEN_WALLET { reason?: string }
- ARCADE:LOG { level: "info"|"warn"|"error", message, data? }

Required messages (parent -> game)

- ARCADE:SYNC {

address,

paidCredits,

promoCredits,

membership,

displayName,   // nickname if active

avatarPng,   // if PRO avatar exists

pohVerified,
}

- ARCADE:RUN_GRANTED { runId, runType, costAC, eligible, rules? }
- ARCADE:RUN_DENIED { reason, message }

3) Start gating (best practice)

When the user presses the first input (tap/click/space):

- If NOT embedded: start immediately (standalone demo still works)
- If embedded:
- do NOT start immediately
- send ARCADE:REQUEST_RUN

- show a lightweight “Starting…” overlay
- if ARCADE:RUN_GRANTED arrives: start, apply the queued “start input” immediately
- if ARCADE:RUN_DENIED arrives: show the message and optionally send ARCADE:OPEN_WALLET

Timeout UX:

- If no response in ~2500ms, show:
- “Connect wallet to play” OR “Not enough credits — open Wallet”

4) Metrics (what to submit)

Metric IDs are standardized in:

- public/arcade-metrics-library.json
- docs/gruesome-arcade-metrics-library-v1.0

Every run should submit:

- score (even if score is not the default metric; the arcade can still show it)
- durationMs (performance.now delta)
- metrics object with the game’s declared metrics

Example metrics for common types:

- Defense (Storm the House 2 style): waves, kills, accuracyBp, durationMs, inRunSpendAC, efficiency
- Speedrun: timeMs, deaths, retries, completionBp
- Puzzle: moves, mistakes, timeMs, completionBp

5) Economy-in-run games (Credits inside the game)

If the game consumes Credits during gameplay (buying upgrades, ammo, etc.):

- Set in arcade-games.json:
- usesCreditsInRun: true
- rankedSpendCapAC: (fairness guardrail)
- In the run result:
- send spentInRunAC (integer)

Important:

- The parent should still be authoritative for the balance.
- The game should track *in-run* spend for fairness metrics.

- The leaderboard defaultMetric should avoid “raw spend wins”.

Prefer efficiency-style metrics (output per spend) or clamp spend impact.

6) Security / hardening

- Always verify postMessage:
- channel key must match
- origin check when possible (document.referrer origin)
- Ignore unexpected message types
- Never trust balances coming from the game
- Never let the game mint payouts or transfer value directly

7) Minimal run result payload (example)

ARCADE:RUN_RESULT payload example:

```
{
  "gameId": "moonshot",
  "runId": "0xRUNID",
  "score": 1234,
  "durationMs": 48210,
  "metrics": {
    "score": 1234,
    "durationMs": 48210,
    "accuracyBp": 8750,
    "nearMissCount": 12
  }
}
```

8) QA checklist

- Standalone: game starts and plays without Arcade
- Embedded:
- READY fires exactly once on load
- Start input triggers REQUEST_RUN and does not start without GRANTED
- RUN_DENIED shows a clear message
- RUN_RESULT posts exactly once per run (no duplicates)
- Metrics:
- values look sane
- durationMs uses performance.now()
- currency / AC fields are integers