

Gruesøme's Arcade — Embedding Guide (Game Library + iframe + Bridge)

Version: v1.3

This guide covers embedding ANY game into the Arcade Game Library with:

- safe iframe + postMessage bridge
- run gating (Web3 locks)
- multi-metric leaderboards

1) Game Library Packaging (public/)

Minimum recommended layout per game: public/games// index.html game.js (or main module) assets/... manifest.json (optional but recommended)

The Arcade should only need:

- the gameId
- the URL to load (usually /games//index.html)
- the game catalog entry in public/arcade-games.json

2) iframe Best Practices (security + UX)

2.1 Use a restrictive sandbox

Example:

2.2 postMessage security hardening

- Use a per-game channel key.
- Parent should validate:
 - message origin matches the iframe origin
 - channel matches expected
 - gameId matches expected

2.3 Fallback UX

If player is not connected or has no funds:

- The game should show a centered message and not start.
- The Arcade should show actionable CTA in Wallet.

3) Bridge Wiring (Parent Side)

Parent responsibilities:

- Maintain the authoritative session:

```
address, paid/promo credits, membership, PoH, avatar
- Reply to ARCADE:READY with ARCADE:SYNC
- On ARCADE:REQUEST_RUN:
  - verify eligibility (connected, PoH if required, credits)
  - assign runId
  - return ARCADE:RUN_GRANTED or ARCADE:RUN_DENIED
- On ARCADE:RUN_RESULT:
  - verify runId belongs to address + game
  - verify metrics within clamps / caps
  - write run result to Redis leaderboards
  - update activity score (paid spend)
  - update UI + claimables
```

4) Catalog Requirements (public/arcade-games.json)

Each game entry should include:

Required:

- id: "moonshot"
- title: "Moonshot"
- type: "skill|defense|puzzle|racing|idle|strategy|rhythm|sports|shooter|platformer"
- url: "/games/moonshot/index.html"
- defaultMetric: "score" (or "waves", "timeMs", etc)
- metrics: [{ id, label, direction, unit, clamp? }]

Recommended:

- usesCreditsInRun: boolean
- rankedSpendCapAC: number

- runCostAC: number (if you charge per run)
- payoutEligible: boolean (if in skill pool)
- description, tags, thumbnail

5) Multi-metric Boards (UI + API)

Leaderboards should be queryable by:

- board: skill | activity | pro | lottery
- period: daily | weekly | all
- metric: score | timeMs | waves | efficiency | ...

Important: direction matters

- score: desc
- timeMs: asc
- durationMs: desc for survival, asc for speedruns

6) 3D Metrics Map (Admin overlay)

Admin data spine endpoint:

- GET /api/admin/snapshot (admin-only)

Client adapter (provided):

- public/gruesome-arcade-3d-map-adapter-v1.0/ga3d-admin-adapter.js

Exports:

- getAdminSnapshot()
- snapshotToGraph()

This lets the 3D map visualize:

- epochs + pools
- catalog coverage
- guardrail alerts
- exclusions

7) Quick Smoke Tests

- Load a game standalone → it starts normally.
- Load same game embedded:
- ARCADE:READY sent
- ARCADE:SYNC received
- Start input triggers ARCADE:REQUEST_RUN
- RUN_GRANTED starts game with queued input
- RUN_RESULT posted on game over

END