# UGS Infinite Runner — Prompt Library

Generated: 2025-10-16 15:20:26 UTC

---

UGS Infinite Runner — Master Context + Prompt Library

Master Context Pack (prepend to every prompt)
---
PROJECT: Infinite Runner using Unity Gaming Services (UGS).
GOALS:
- Cloud Save: profile/progression, personal leaderboards (lifetime/weekly/monthly), 12-week history.
- Leaderboards: Weekly tiered (Bronze/Silver/Gold/Platinum, threshold-based), Monthly tiered (phase 2), Lifetime global.
- Economy: server-authoritative coins/gems, daily/weekly rewards via Cloud Code only; Cloud Save holds non-authoritative mirrors for UI.
- Anti-cheat: all writes go through Cloud Code; distance/duration sanity; per-run coin caps; one submission per run; metadata requirements.
- Resets: Weekly Mon 00:00 UTC; archive personal weekly to 12 entries; demote/promote by thresholds.
- UX: personal bests, tier badge, countdown; Kanban tasks tracked in this chat's canvas "UGS Infinite Runner — Kanban".
DECISIONS (LOCKED):
- 4 tiers; threshold promotion; personal history depth 12 weeks; leaderboard entries carry metadata (distance, duration, characterId, runVersion).
ACCEPTANCE PRINCIPLES:
- Cross-device persistence; no client-authoritative economy; leaderboards reflect correct tier; reset accuracy; <1% false positives on anti-cheat.
REFERENCES:
- Plan PDF in chat ("UGS Infinite Runner Plan (Decisions & Execution)").
- Canvas Kanban in this chat: "UGS Infinite Runner — Kanban".

Prompt Library
---
1) User Story Kickoff
- As a… I want… so that…
- Acceptance Criteria (bullet, testable, no code)
- Non-Goals
- Dependencies
- Risks + Mitigations (max 3)
- Estimation by owner (Backend/Client/Data/QA/Ops)

2) Task Breakdown from a Story
- Work Items table: ID, Title, Owner, Priority, Status=To Do, Exit Criteria
- Sequencing with parallelizable notes
- Test Cases (QA) bullets
- Telemetry events/fields

3) Cloud Code Change Request (spec-only, no code)
- Endpoint name, purpose
- Inputs (fields/types/constraints)
- Server Validations (rules, bounds, timestamps, idempotency keys)
- Side Effects (Cloud Save keys, Leaderboards, Economy)
- Failure Modes (errors)
- Rate Limits
- Observability (logs/metrics)
- Security (trust/callers)
- Acceptance Tests

4) Anti-Cheat Rule Update (spec-only)

- Signals used
- Detection rules
- Actions (reject/shadowban/log)
- False positive safeguards
- QA tests (legit vs cheat edges)

5) Leaderboard/Tier Change Ticket
- Change summary
- Affected boards (IDs), tiers, thresholds
- Rollover (Mon 00:00 UTC) & migration
- Backfill personal history (12 cap)
- UI comms
- Rollback plan

6) Economy/Reward Flow Ticket (spec-only)
- Reward table (amounts, caps, cooldowns)
- Call path (client → Cloud Code → Economy/Cloud Save)
- Idempotency & replay safety
- Abuse prevention (daily/weekly limits)
- Acceptance Criteria (server timestamps, no client grants)
- Analytics emitted

7) UX Copy + Telemetry Checklist
- Microcopy (buttons, errors, toasts)
- Empty/edge states
- Countdown handling (Mon 00:00 UTC)
- Accessibility notes
- Telemetry fields + sample payload names

8) Test Plan Slice (per feature)
- Scope in/out
- Environments (time travel for reset, offline)
- Boundary matrix (e.g., 19,999 ↔ 20,000)
- Negative tests (invalid metadata, stale version)
- Performance checks (latency targets)
- Sign-off criteria

9) Risk Review (pre-release)
- Top 5 risks (title, trigger, impact, owner)
- Runbooks (reset failure, LB post outage, economy drift)
- Mitigation readiness (Y/N)
- Go/No-Go checklist

10) Kanban Sync Helper
- For each item: ID → new Status/Owner/Priority + one-line rationale
- Flag blockers with dependency IDs
- List missing tasks