Idle RPG — GDD & MVP (Live)

Repo: https://github.com/AlfredGoldfish/IdleRPG

---

1) Game Snapshot

Pitch: Tap-style, side-view idle battler. Enemies spawn, player causes damage (tap/auto), enemies drop visible coins that vacuum to the player and persist via simple JSON saves.

Core Loop:

1. Enemy spawns ➔ player damages ➔ enemy dies.
2. Coins burst into the world ➔ magnet tugs coins ➔ player collects.
3. Wallet updates ➔ numbers persist across sessions ➔ repeat.

Design Pillars: clarity of feedback (popups, coins), inspector-first iteration, tiny-file workflow.

---

2) Current Systems (Shipped)

Combat

- **Health2D**: Max/Current with `OnDamaged/OnHealed/OnDied` events.
- **Enemy**: delegates to Health2D; click-to-damage; notifies spawner on death.
- **Damage Popups**: world-space TMP mesh; anchor/jitter/z-offset knobs.

Loot & Currency

- **CoinPickup2D**: world pickup (trigger + optional click); `Initialize(metal, value)`.
- **CoinDropper2D**: burst spawn on `OnDied` (min/max count & value, radius, impulse, torque, lifetime override).
- **MagnetCollector2D**: vacuum using `Physics2D.OverlapCircle(ContactFilter2D, …)`; no center override; smooth velocity steer.

Economy & Save

- **Wallet (POCO)**: `Get/Add/Set/ClearAll` + `OnChanged(metal, total)`.
- **PlayerEconomy**: singleton host of the one Wallet instance.
- **WalletSave**: JSON with idempotent **set** semantics on load.
- **SystemsBootstrap**: robust hookup in `Start()` (+ retry on scene load); autosave on change; save on pause/quit/destroy; **ResetSave()** context menu.

HUD / Debug

- **HUDCurrencyCounterSession**: session-only, anchor-based counters; Reset starts at 0 so next coin shows 1.
- **HUDDebugTools**: `ResetHudCounters()` (display-only) and `ResetPersistentAndHud()` (clears file + wallet + re-anchors HUD).

Folder Layout

```
Assets/
  Scripts/
    Core/    (Wallet, PlayerEconomy, WalletSave, SystemsBootstrap)
    Combat/  (Health2D, Enemy, …)
    Loot/    (CoinPickup2D, CoinDropper2D, MagnetCollector2D)
    UI/      (HUDCurrencyCounterSession, HUDDebugTools)
```

3) MVP Roadmap (Tiny Steps)

-

Later

-

4) Acceptance Checks

Magnet: coins glide (no teleport/jitter), no heavy GC on \~50+ coins.

Formula: tweaking one curve affects all enemies; no per-prefab edits.

Metals: visual matches metal; per-metal tallies update.

Pooling: behavior identical; frame spikes on mass kills disappear.

Inventory: players see totals in Bag; Wallet event drives UI.

Persistence: restart loads exact totals; ResetSave zeros file + wallet; no double-load.

5) Tiny Packs — How We Work

Rhythm: 1–3 scripts per pack, with README + wiring checklist. Import → 30s acceptance test → iterate.

Applying a pack

1. Unzip into project root so `Assets/` merges.
2. Follow the README wiring steps (what to attach, suggested inspector values).
3. Run the acceptance checks.

Code Style

- C#9-safe (no file-scoped namespaces), correct `using`s.
- Inspector-first configuration; minimal allocations; pooling later.
- Model (POCO) vs Scene (MonoBehaviours) separation.

Packs shipped so far

- **01 – Save & Loot Stabilization**: Wallet, Bootstrap (Awake version), CoinPickup/Dropper, Magnet.
- **02 – HUD Session Counters + Reset + Magnet cleanup**.
- **03 – Anchor-based HUD + Bootstrap Start()**.
- **03b – Fix Double-Load**: set-semantics load + centralized load + session guard.
- **03c – Robust Hookup + ResetPersistentAndHud**.

---

6) Wiring Quick Reference

- **_Systems** (scene root): add `SystemsBootstrap`.
- **Player**: has `PlayerEconomy`; child trigger tagged **Collector**.
- **Enemies**: `Health2D` + `CoinDropper2D` (assign `Coin_*` prefab).
- **HUD**: per-metal `HUDCurrencyCounterSession` on text labels; add `HUDDebugTools` to a HUD object; Button → `ResetPersistentAndHud()`.

---

7) Notes & Known Good Settings

- Coins: RB2D gravity = 0, Interpolate = On, Drag ≈ 1–2, Collider isTrigger.
- Magnet: buffer size 64 (raise if >64 coins on screen regularly).
- Layers: coins collide with Collector, not with enemies/level.
- Editor: if Enter Play Mode Options disable Domain Reload, ensure loaders are idempotent (done), or re-enable reload.

---

8) Parking Lot / Open Questions

- Do we want coin-merge rules under heavy drops?
- Any analytics/telemetry needed (enemyId) before we cull that field?
- When to introduce pooling based on real perf numbers?