

スクリプト相関図・機能一覧 (2025-10-27)

対象：今日付の主要スクリプト（一覧提供分）。 「何を担当し、誰とやり取りするか」を一望できるように整理。

1. ディレクトリ構成（対象）

```
res://scripts/
    world.gd
    csv_loader.gd
    map_layer.gd

res://ui/
    debug_panel.gd
    Dialog.gd
    dialog_player.gd
    dice_overlay.gd
    game_hud.gd
    inventory_window.gd    (※実ファイルは inventory_window.gd の表記。要リネーム整理)
    menu_panel.gd
    move_window.gd
    rank_table.gd
    trade_window.gd
```

2. 中核（Core）

res://scripts/world.gd（中核ロジック）

- ・役割：世界状態と日次進行・価格・供給・イベント適用の“心臓部”。
- ・主要シグナル：
 - `day_advanced(day)` : 日付が進んだことを通知。
 - `world_updated()` : 日次処理完了後の状態更新通知。
 - `supply_event(city_id, product_id, qty, mode, flavor)` : 供給やイベントの発生通知 (`mode="supply" | "event"`)。
 - `stall_sold(...)` / `player_decay_event(...)` : 屋台売上・疲労関連。
- ・主要API：
 - `step_one_day()` : 日付+1 → `use_event_dice=true` なら シグナルのみ発火 (HUDが可視ロール → `finalize_day()`)。 `false` なら従来どおり内部で完結。
 - `finalize_day()` : 生産/消費→供給→価格→市/屋台→NPC→到着→疲労→`world_updated()`。
 - `resolve_daily_with_roll(roll)` / `resolve_travel_with_roll(roll)` : 出目→tier→テーブル抽選 (`events_daily.csv` / `events_travel.csv`)。
 - ・データ読み込み：`csv_loader.gd` で読んだ CSV (cities/products/routes/events/...) を保持。

res://scripts/csv_loader.gd (データ読込)

- ・役割：CSV群のロード・行辞書化・キャッシュ提供。
- ・関係：`world.gd` が起動時に利用。レイアウト変更（列追加：`tier` / `Layer` など）にも追従。

res://scripts/map_layer.gd (マップ表示)

- ・役割：地図・都市・ルートの表示／ハイライト。
- ・関係：`world.gd` の都市・ルート情報を参照。`move_window.gd` の選択結果を視覚反映。

3. UI (HUD / 演出 / 操作)

res://ui/game_hud.gd (HUD・イベントUIハブ)

- ・役割：HUD総元締め。ログ領域、トースト/ダイアログ表示、ダイス演出のオーケストレーション。
- ・受信：
 - `world.day_advanced` → `_on_day()` : 可視ダイス (`dice_overlay.gd`) を起動→`resolve_*` を順に実行→最後に `world.finalize_day()`。
 - `world.supply_event(..., mode, flavor)` → `_on_supply_event()` : `mode="event"` はフレーバーダイアログ (ON/OFF可) 、`mode="supply"` は入荷通知。
 - `world.world_updated` → `_on_world_updated()` : 各パネルの再描画。
- ・設定：
 - `show_event_log_panel` (ログ表示) 、`event_log_rows` (行数) 、`show_supply_dialog` (入荷ダイアログ) 、
 - `show_event_flavor_dialog` (イベントのフレーバーダイアログON/OFF)。
- ・関係：`dice_overlay.gd` / `Dialog.gd` / `dialog_player.gd` と連携。`menu_panel.gd` の操作結果を受けて描画更新。

res://ui/dice_overlay.gd (ダイス演出)

- ・役割：可視ロール。`roll_started(kind)` / `roll_revealed(value)` / `roll_done(kind,value)` を発火。
- ・関係：`game_hud.gd` が生成・購読し、結果を `world.resolve_*_with_roll` へ受け渡し。

res://ui/menu_panel.gd (メニューバー)

- ・役割：一日進める/ポーズ等の操作。
- ・動作：`world.step_one_day()` / `world.pause()` を直接呼び出し。HUD経由の演出と整合。

res://ui/debug_panel.gd (デバッグ)

- ・役割：価格・供給イベント数などの内部値を可視化。
- ・関係：`world.world_updated` を購読、必要に応じて `world` に直接問い合わせ。

res://ui/Dialog.gd / res://ui/dialog_player.gd (会話・告知)

- ・役割：テキスト表示、スキップ/進行制御（モーダル）。
- ・関係：`game_hud.gd` のイベントフレーバー表示や各種システムダイアログの基盤。

res://ui/move_window.gd (移動/ルート選択)

- ・役割：出発地→目的地、護衛や危険度の確認と決定。
- ・関係：決定後 `world` のプレイヤー状態 (`enroute/dest/arrival_day`) を更新。`map_layer.gd` と見た目連携。

res://ui/trade_window.gd (売買UI)

- ・役割：都市ごとの在庫・相場を見ながら売買操作。
- ・関係：`world` の在庫・価格に読み書き。売買後に `world.world_updated` でHUD再描画。

res://ui/inventory_window.gd (所持品UI)

- ・役割：プレイヤーの持ち物表示（※ファイル名は `inventory_window.gd` が実体。要リネーム整理）
- ・関係：`world` のプレイヤー在庫を表示・更新。

res://ui/rank_table.gd (ランキング/成績)

- ・役割：利潤・交易スコアなどの集計表示。
- ・関係：`world` の累積値を参照。

4. 相関 (ざっくり矢印)

- ・`menu_panel → world` : `step_one_day()` / `pause()` を発火
- ・`world → HUD` : `day_advanced` / `world_updated` / `supply_event`
- ・`HUD → dice_overlay` : ロール開始 (`daily` → `travel`) / 結果購読
- ・`dice_overlay → HUD` : `roll_started` / `revealed` / `done`
- ・`HUD → world` : `resolve_daily_with_roll()` → (移動中なら) `resolve_travel_with_roll()` → `finalize_day()`
- ・`move_window → world` : 移動先・到着日・護衛などを設定
- ・`trade_window → world` : 売買反映 → `world_updated`
- ・`map_layer ← world` : 都市・ルート・危険度を参照
- ・`Dialog / dialog_player ← HUD` : イベントのフレーバー文を表示

5. 掃除 (削除・統合の目安)

- ・重複UIの統合：告知系 (AcceptDialogと自作Dialogの二系統) が並立する場合、`Dialog.gd` に一本化。
- ・命名整理：`inventry_window.gd` → `inventory_window.gd` (スクリプト名・シーン名とも統一)。
- ・責務の境界：
 - ・日次パイプラインは `finalize_day()` に集約 (他関数末尾の重複処理を削除済)。
 - ・イベントのダイアログON/OFFはHUDの `show_event_flavor_dialog` に統一。

6. 参考 (シグナル一覧 抜粋)

- ・`world.gd` : `day_advanced` / `world_updated` / `supply_event` / `stall_sold` / `player_decay_event`
- ・`dice_overlay.gd` : `roll_started(kind)` / `roll_revealed(value)` / `roll_done(kind,value)`

- HUDは上記を購読し、`_on_day()` / `_on_supply_event()` / `_on_world_updated()` 等で処理。
-

この一覧で「残すべきメイン」と「統合候補」が判断しやすくなります。削除計画に入る前に、依存元（接続先）を確認してから実施してください。