

# 評判システム設計メモ（商人ゲーム）

## 1. 評判とは何か

- ・「取引相手としてどれくらい信頼されているか／嫌われているか」をスコア化したもの。
- ・行政・領主、商人ギルド、地元商人・職人、市民などの総合評価。
- ・主な役割：
  - ・各地域の商業認可証の条件
  - ・メインクエストの進行条件
  - ・契約達成／失敗の評価
  - ・市場を荒らす行為（悪質な買い占めなど）へのペナルティ

---

## 2. 評判のデータ構造（グローバル＋リージョン別）

World.gd 側に以下のイメージで保持する：

- ・`rep_global: float`
- ・世界共通の評判（例：0～100）
- ・`rep_by_province: Dictionary`
- ・`{ "Pilton": 20.0, "XXX": 5.0, ... }` のようにリージョン（province）ごとに保持
- ・（将来用）`rep_by_city: Dictionary`
- ・MVP段階では空、必要になった時に拡張

### 有効評判値の考え方

- ・都市ごとに「有効評判値」を計算：
- ・`effective_rep(city_id) = global` と、その都市が属する province の評判を合成
- ・例：`effective_rep = rep_global * 0.3 + rep_by_province[prov] * 0.7`
- ・実際の判定（認可証、契約、税率など）は基本的にこの「有効評判値」を参照する。

---

## 3. 数値レンジと段階ラベル

- ・内部値のレンジは 0～100 を想定。
- ・UI上は段階・ラベルとして表示：

値	ラベル案	イメージ
0～9	無名	誰も知らない新人
10～24	顔見知り	そこそこ仕事をしている商人
25～49	評判良し	契約も回ってくる、軽く優遇
50～74	信頼厚い	高難度契約・認可証の話が来る

値	ラベル案	イメージ
75~100	名うて	特別案件・VIPルートが解放される

- World.gd に `func reputation_tier(value: float) -> String` のようなヘルパー関数を用意して、UIから使いやすくする。
- 

## 4. 評判が上がる行為 (MVP)

### 4-1. 契約達成

- 契約を期限内にクリアすると、その契約に紐づく province の評判と global が上昇。
- rarity別の加点例：

  - common : +2
  - uncommon : +3
  - rare : +4
  - epic : +6

### 4-2. メインクエスト進行

- 章や大きな節目をクリアしたとき：
- 関係する province : +5~+10
- global : +3 程度 (世界規模の噂)

### 4-3. 将来候補

- 定期市での「貢献度」が一定以上
- 商館システムで「期待値以上の利益」を継続して出す

MVPでは「契約」と「メインクエスト」から始め、あとで定期市や商館に接続する。

---

## 5. 評判が下がる行為 (MVP)

### 5-1. 契約失敗・期限切れ

- 契約の未達成／期限切れで、その契約に紐づく province の評判が低下。
- rarity別の減点例：

  - common : -3
  - uncommon : -5
  - rare : -8
  - epic : -12

### 5-2. 悪質な買い占め (生活必需品)

#### 基本方針

- すべての買い占めが悪ではない。
- 評判が下がるのは「生活必需品を枯らして、他人の生活や商売に実害を与えたとき」。

## 判定条件の例

- 対象商品が「生活必需度が高いカテゴリ」である。
- 取引後の在庫が、目標在庫に対して極端に低い：  
• `stock_qty / target_stock < 0.2` など
- その日のプレイヤー購入量が、その都市での基準在庫に対して大きい：  
• `today_player_buy_qty > target_stock * 0.5` など

これらの条件を満たしたときに、初めて「悪質な買い占め」と認定し、評判ペナルティを与える。

## 5-3. 他の市場／リージョンへの影響

- 直接のペナルティ対象は、その都市が属する province の評判。
- 将来的には「噂の伝播」として、隣接リージョンや global に小さいマイナスを飛ばす拡張も可能。

## 6. 商品カテゴリと評判用係数

### 6-1. 既存のカテゴリと消費係数

World.gd より：

```
@export var category_base_consume: Dictionary = { # category→日次ベース (RANK5時の基準)
  "staple": 0.80, # 主食 (小麦/豆/パン等)
  "food": 0.60, # 食料 (魚/チーズ等)
  "drink": 0.50, # 飲料 (エール/ワイン等)
  "textile": 0.22, # 繊維 (羊毛/麻等)
  "material": 0.15, # 材料 (木材/革/石鹼等)
  "luxury": 0.08, # 嗜好品
  "general": 0.20 # その他
}
```

これは「どれくらい日常的に消費されるか」を表す係数であり、評判で見たい「枯らしたときにどれだけ強く悪評が出るか」とは軸が異なる。

### 6-2. 評判用のカテゴリ係数を別で持つ

評判では別軸として「カテゴリごとの評判ダメージ係数」を用意する。

```
@export var category_rep_weight: Dictionary = { # 買い占め時の評判ダメージ係数
  "staple": 1.0, # 主食：ガチで怒られる
  "food": 0.8, # 食料：かなりマズい
  "drink": 0.6, # 飲料：生活には直結しにくいが恨まれる
  "textile": 0.4, # 繊維：生活必需だが短期インパクトはやや小さい
  "material": 0.3, # 材料：職人筋に影響、大衆には伝わりにくい
  "luxury": 0.0, # 嗜好品：原則ノーペナルティ (別ロジックでプラスも検討)
  "general": 0.2 # その他：一応少しだけ
}
```

買い占めペナルティ例：

```
var base_penalty := -3.0 # 「悪質な買い占め」の基本ダメージ
var cat: String = product_category # "staple" など
var w := float(category_rep_weight.get(cat, 0.0))
var rep_delta := base_penalty * w
change_rep(province_id, rep_delta, "buyout")
```

- `category_base_consume` と独立しているので、経済バランス調整が評判システムに波及しない。
- 初期値は `category_base_consume` を参考にしつつ、人間の感覚で微調整する。

## 7. 評判の利用先

### 7-1. 商業認可証（リージョンごとの許可）

- provinceごとに「商業認可Lv1～3」などを設定。
- 取得条件：
  - `rep_by_province[prov] >= 閾値 + 手数料 (費用)`
- 認可レベルに応じて：
  - その地方の契約ランク解放 (common～epic)
  - 商館設立権
  - 税率軽減、定期市の事前情報など

World.gd 側の例：

- `player_permits: Dictionary = {"Pilton": 1, ...}`
- 契約生成、ルート解放、商館関連などの条件チェックで `effective_rep` + `player_permits` を使用。

### 7-2. 契約の質と量

- 評判が高い province では：
  - 契約オファー数の上限増加
- rarity 重み補正 (rare/epic が出やすくなる)
- 評判が低いと：
  - 契約がそもそも来ない
  - common 中心、期限がシビア、報酬が低いなどのデメリット

### 7-3. 税・手数料・護衛などの価格

- 既存の `get_trade_tax_rate()` の rank 判定に評判も組み込む：
  - 評判が高い province では `tax_rate *= 0.9`
  - 惡名レベルだと `tax_rate *= 1.1`
- 护衛費・通行料などにも将来連動させる余地あり。

---

## 8. 実装フェーズ案（MVPロードマップ）

### フェーズ1：フィールドとAPIだけ追加

- World.gd に以下を追加：
  - `rep_global`
  - `rep_by_province`
  - `func change_rep(province: String, delta: float, reason: String = "")`
  - `func get_effective_rep_for_city(cid: String) -> float`
  - `func get_rep_tier_label(value: float) -> String`
- UI (MenuPanel / HUDなど) に「Pilton評判：評判良し」程度の簡易表示を追加。

### フェーズ2：契約システムに紐付け

- 契約達成／失敗処理から `change_rep()` を呼び出し、実際に数値が上下するようにする。
- この時点ではまだ認可証や税率への影響は入れない。

### フェーズ3：認可証・メインクエスト条件へ適用

- メインクエストの進行条件に `rep_by_province` / `effective_rep` を組み込む。
- 新リージョン解放や商業認可証の取得条件にも評判を使用。

### フェーズ4：買い占めペナルティ・経済システムとの連動

- プレイヤーの1日ごとの取引ログ (city, pid, buy/sell, qty 等) を記録。
- 日次処理で「悪質な買い占め判定」をし、条件を満たした場合にだけ `change_rep()` でペナルティ。
- 将来的には「世界ニュース」やイベントと連携させ、global評判への影響も検討。

---

## 9. プレイヤーへの見せ方（UXメモ）

- 商品説明やチュートリアルで：
- 生活必需品カテゴリには「買い占めると地元で悪評が立つことがある」等の注意書きを入れる。
- 取引確認時：
  - 買い占めラインを超えそうなときに「この取引は〇〇地方での評判-2の原因となる可能性があります」と警告を表示。
- 嗜好品・贅沢品：
  - 買い占めによる評判マイナスは原則なし。
  - 逆に「高級品を大量に扱う名うての商人」として別のフラグやイベントに繋げる余地あり。

以上をベースに、まずは「契約とメインクエストに紐づく評判システム」としてMVP実装し、その後に買い占め・定期市・商館などへ連動を拡張していく。