

Class Conflict: Millionaire

大富豪 × 人狼 × Mistral AI

Daifugo × Werewolf × AI-Generated Characters

Mistral AI Hackathon 2025



大富豪 / Card Game



人狼 / Werewolf



AI生成 / AI-Generated



陽動 / Decoy Bluff

コンセプト / Concept

3つの要素の融合 — Three Mechanics as One



大富豪（カードゲーム）

Daifugo – card game

×



人狼（議論・投票・正体隠匿）

Werewolf – social deduction

×



Mistral AI（リアルタイム生成）

Real-time AI character generation

「カードゲームの"事実"が、人狼の"議論"を動かす」

"Last night's card plays become today's evidence in debate."

ゲームフロー / Game Loop

☀ 昼 / Day – 会議 / Council

チャットで議論 Debate via chat



ヘイト投票 Hate vote



処刑 Execute suspect



イカサマフェーズ / Bluff Phase

貧民が陽動+本命でカードを奪う

Hinmin steals cards via decoy bluff

🌙 夜 / Night – 大富豪 / Daifugo

全員でカードを出し合う Play cards in turn



場が3回流れたら翌日へ 3 table clears → next day

3つの階級と勝利条件

Three Classes & Win Conditions

階級 / Class	人数	目的 / Objective
富豪 🏰 Fugo / Rich	1名	誰よりも早く手札を0枚に First to empty hand
貧民 🗡️ Hinmin / Poor	1名	革命 / 富豪より先に上がる / 富豪を吊るす Revolution / beat Fugo / hang Fugo
平民 👤 Heimin / Common	3名	富豪と貧民を両方追放 Eliminate both Fugo & Hinmin

隠れ勝利条件 (True Win) Secret Personal Missions

目玉機能：陽動イカサマ

Core Feature: The Decoy Bluff System

貧民だけが使える**非対称な騙し合い** — Asymmetric deception for Hinmin only

イカサマ側（貧民） Attacker — takes their time:

- 【陽動 / Decoy】 天井を指さして「うわっ、虫！」と叫ぶ
Points at ceiling and yells "A bug!"
- 【本命 / Real】 相手が上を向いた隙に右端のカードを奪う
Steals the rightmost card while target looks up

防御側（ターゲット） Defender — 5 seconds / 15 chars only!

反応 / Reaction	結果 / Result
	大成功 — カード盗奪 バレない / Card stolen

Mistral AI の活用 / How We Use Mistral AI

用途 / Purpose	モデル / Model	内容 / Content
キャラクター生成 Char gen	mistral-large	職業・性格・関係を生成 Job, personality, connections
イカサマ審判 Bluff judge	mistral-large	陽動vs対策を3段階判定 3-tier decoy judgment
NPC昼発言 NPC debate	mistral-small	伏線・示唆・扇動を含む発言 Hints, implications, agitation
NPC投票判断 NPC vote	mistral-small	カード証拠+発言から投票先決定 Evidence-based voting

NPCの思考ロジック / NPC AI Logic

Trust × Affinity の2軸パラメータ — Two-axis parameter system

NPCは **Trust**（論理的信頼度） × **Affinity**（感情的好感度） で動く

Each NPC maintains Trust (logic) and Affinity (emotion) scores toward every other player

性格 / Personality	行動原理 / Behavior
論理型 Logical	カードの強さ・矛盾を根拠に告発 Accuses based on card evidence
感情型 Emotional	好きな相手は庇う。「探偵が嘘つきだ！」 Defends favorites, ignores logic
中立型 Neutral	感情と論理のバランスで動く Balances emotion and logic

技術スタック / Tech Stack

Browser

Vanilla JS + HTML + CSS (no build tools)

Dark gothic theme – black × gold

fetch (same origin)

FastAPI

main.py – 29 endpoints, in-memory state

game_engine.py – Daifugo logic (pure functions)

ai_service.py – Mistral API (async)

models.py – Pydantic v2 schemas

Mistral API

mistral-large-latest (gen + judge)

mistral-small-latest (NPC actions)

プレイ体験の流れ / Player Experience

1. ゲーム開始 Game Start

Mistralが5人のキャラクターと関係性を即時生成

Mistral instantly generates 5 characters with backstories and connections

2. ☀️ 昼の会議 Day Council

NPCが昨夜のカードを根拠に疑いをかける

"You played a Joker last night — you're obviously the Fugo!"

3. 🎭 イカサマ対決 Bluff Duel

5秒の心理戦。陽動で惑わせてカードを強奪

5-second psychological battle. Misdirect and steal.

4. 🌙 夜の大富豪 Night Card Game

奪ったカードで有利に。強いカードは翌日の証拠になる

Stolen cards help you — but playing them makes you suspicious tomorrow

まとめ / Summary

Class Conflict: Millionaire が実現したこと

What We Built

- **プロンプトが武器になる** イカサマ対決
Natural language as a weapon — free-form bluff vs. reaction
- **毎回違うドラマ** を生み出すAI生成キャラクター
AI-generated characters ensure every game tells a different story
- **人狼 × 大富豪** の情報を連動させた推理
Card game "facts" feed directly into social deduction "debate"

「カードゲームの"事実"が、人狼の"議論"を動かす」

Q & A

ご質問をどうぞ / Questions Welcome

 ゲーム時間 / Play time?

 AI速度 / AI latency?

 セットアップ / Setup?

 今後 / Roadmap?

Q: 1ゲームの時間は？ / How long is one game?

A: ライトモードで 約**15～20分** — ~15-20 min in Lite mode

フェーズ / Phase	時間 / Time
ゲーム開始・キャラ生成 Setup & char gen	~5秒 / sec
昼の会議 Day council	~3～5分 / min
イカサマフェーズ Bluff phase	~30秒 / sec
夜の大富豪 Night card game	~5～8分 / min

Liteモードは「職業・性格・顔見知り1名」の軽量設定で高速化
Lite mode uses minimal character profiles for faster generation

Q: AIの応答速度は？ / AI Response Latency?

シーン / Scene	モデル / Model	速度 / Speed
キャラ生成 Char gen	large	~3~5秒 / sec
NPC発言 NPC speech	small	~1~2秒 / sec
イカサマ審判 Bluff judge	large	~2~3秒 / sec
NPCカードプレイ Card play	small	<1秒 / sec

NPC発言は現在**直列処理**。並列化で2~3倍の高速化が可能（既知の改善点）

NPC speeches currently run serially — parallelizing is a known optimization

Q: セットアップ方法は？ / How to Run?

```
# 1. Clone the repo
git clone <repo-url> && cd Mistral_hack2

# 2. Install dependencies
cd backend
pip install -r requirements.txt

# 3. Set API key
cp ../.env.example .env
# Add MISTRAL_API_KEY=<your-key> to .env

# 4. Start server
uvicorn main:app --reload
```

ブラウザで `http://localhost:8000` を開くだけ！

Open `http://localhost:8000` that's it!

Q: 今後の展開は？ / Roadmap?

短期 Short-term

- [] NPC発言の並列処理化（応答速度2〜3倍） Parallel NPC speech generation
- [] 探偵役職のフルUI実装 Full Detective role UI
- [] ゴーストモード（処刑後も観戦） Ghost spectator mode

中期 Mid-term

- [] Hardモード完全実装（記憶喪失・1000字ワールド） Hard mode: amnesia + 1000-char world
- [] SQLite永続化 SQLite persistence (no data loss on restart)

長期 Long-term

- [] マルチプレイヤー対応 Multiplayer support