🔍 Jury Deliberation Simulator

Ever watched 12 Angry Men and wondered what really happens in that jury room? 🤔

How do strangers wrestle with facts, biases, and pressure to reach a unanimous verdict?

🎯 Our Goal

We built an interactive jury deliberation simulator using Python and Google Gemini AI

Not just another chatbot—but an exploration into whether Generative AI can realistically capture the messy dynamics of human jury deliberation.

⚖️ The Case Setup

You're Juror #4 in a murder trial:

```
•   Defendant: Jack Miller, 18, accused of killing Leo Briggs
•   Murder weapon: Jack's knife, but only victim's prints found
•   Shoe prints at scene match Kai (Jack's rival), who also had motive
•   Jack's digital alibi exists—but no eyewitness
```

The evidence is ambiguous, packed with reasonable doubt.

🧠 Simulation Modes

You can choose how your AI jury is assembled:

```
•   Scripted Mode
```

Clearly defined jurors with fixed personas (stored in SCRIPTED_JURORS_DATA)

Perfect for controlled tests or repeatable experiences

```
•   Generated Mode
```

Jurors dynamically generated by AI (generate_jurors()), creating new personalities every run

Ideal for replayability and exploring unpredictable group dynamics

All jurors are guided by a shared behavioral framework (JUROR_TEMPLATE) to ensure consistency.

🛠️ Technical Overview

💬 LLM-Powered AI Agents

Agents created using LangChain + Google Gemini 2.0 (Flash)

Each juror is semi-autonomous, making decisions based on:

```
•   Personal biases and persona traits
•   Complete deliberation history (stored transcript)
•   Context-aware LLM prompting to guide responses
```

🔄 Core Gameplay Loop (run_game())

The gameplay alternates between:

```
•   AI jurors taking turns speaking or responding
•   Player's actions, including:
•   Speaking to influence deliberation
•   Skipping a turn
•   Initiating voting to reach a verdict
•   Exiting deliberation (available after 2 voting rounds)
```

Gameplay ends when:

```
•   A unanimous verdict is reached (Guilty or Not Guilty)
•   Player chooses to exit
•   A hung jury occurs (no unanimous decision after votes)
```

🧠 Designing Juror Behavior

Each AI juror's actions are shaped by:

```
•   Detailed persona descriptions (static or AI-generated)
•   Memory of previous arguments and interactions
•   Structured AI prompts to ensure logical, consistent dialogue
```

📈 Player Performance Evaluation

After deliberation ends:

```
•   The entire conversation and rubric criteria are analyzed by Gemini AI
•   The player receives:
•   Five performance insights highlighting strengths and areas for improvement
•   An overall score out of 100
•   Optional recommendations for better performance in future simulations
```

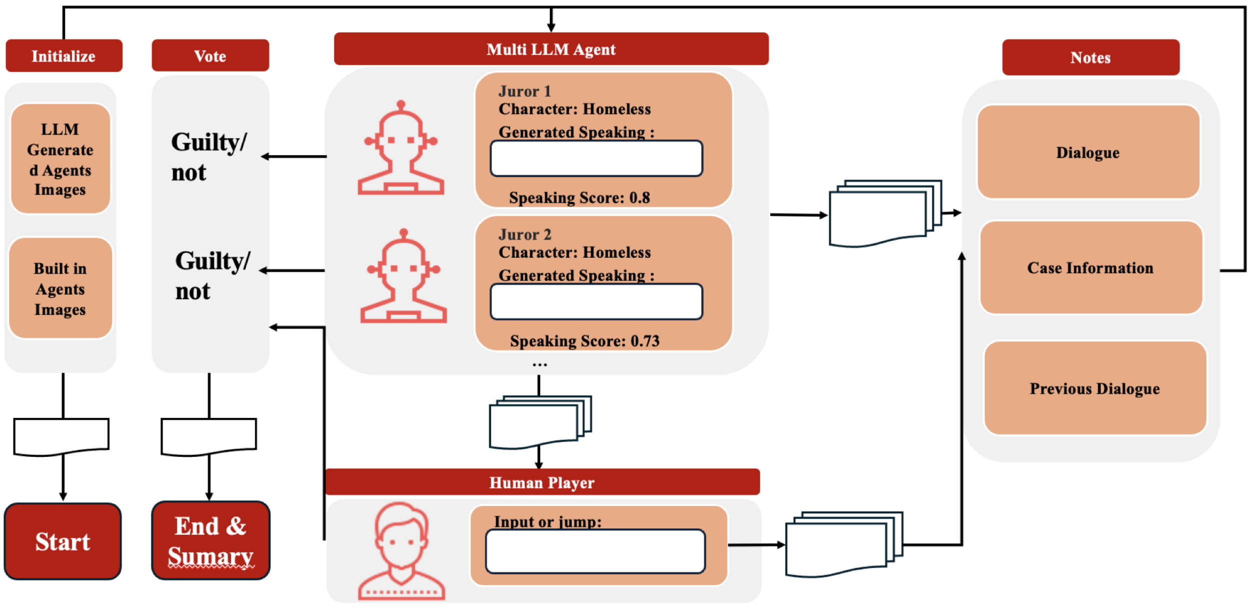
🚀 Future Directions

Exciting expansions under consideration:

```
•   Adding visual evidence like crime-scene photos
•   Allowing players to take the role of lawyers, presenting evidence to AI jurors
•   Creating smarter, emotionally intelligent AI jurors who form alliances or rivalries
•   Developing a web-based UI or open API for educational and research purposes
```

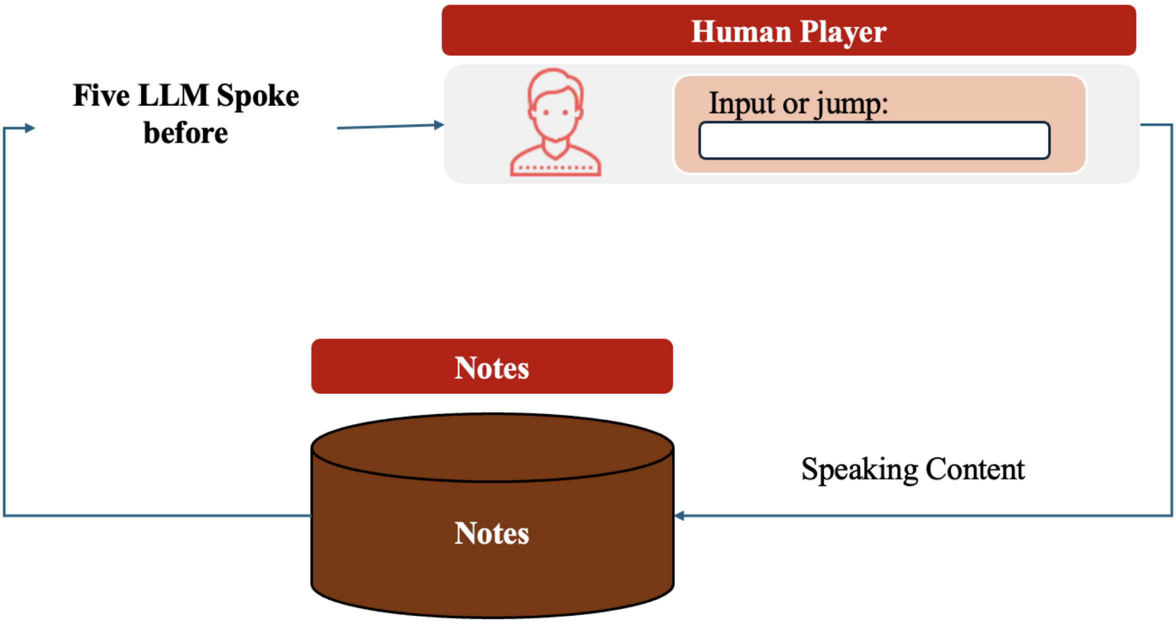
LLM Jury Simulation Flowchart

This diagram illustrates a courtroom simulation system powered by multiple LLM agents as jurors. Each juror votes and speaks based on case notes, while a human player interacts with them. The system initializes with agent personalities, facilitates dynamic dialogues, and concludes with a summary based on collective verdicts and reasoning scores.

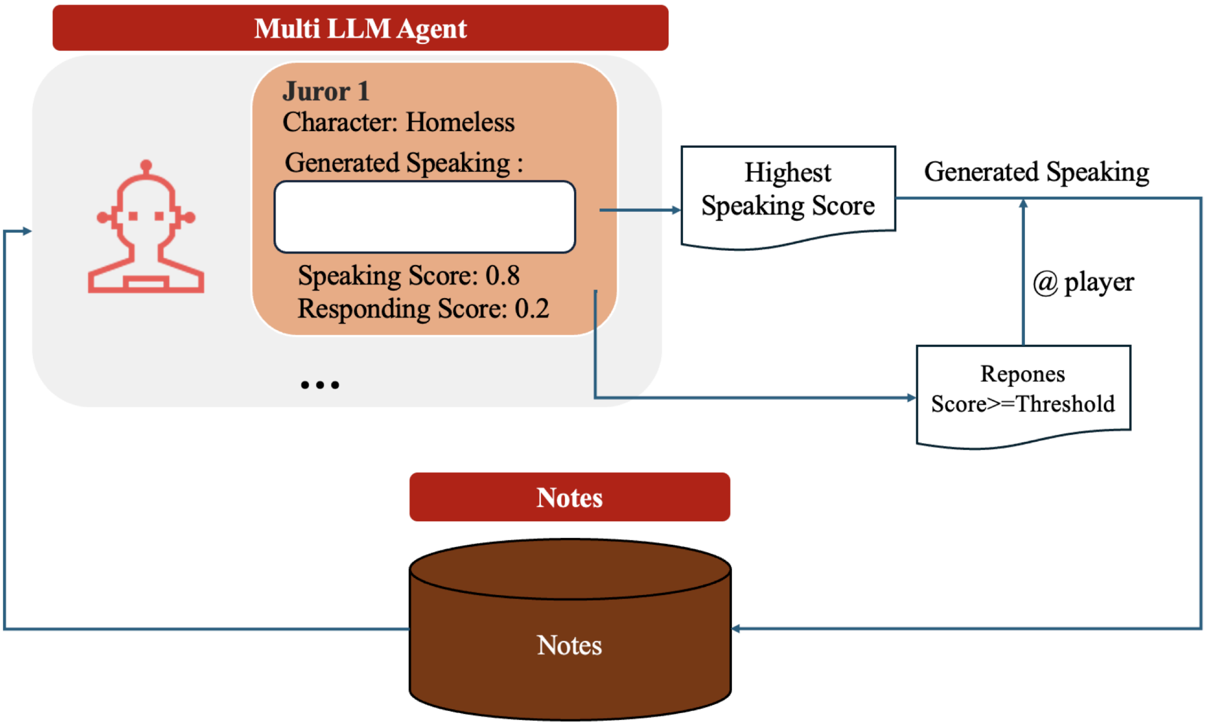


![Flow_chat 1]

![



](Flow_chat2)



Flow_chat3

Step 1: Enviroment and API set up

```
# Remove conflicting packages from the Kaggle base environment.
!pip uninstall -qqy kfp jupyterlab libpysal thinc spacy fastai ydata-profiling google-cloud-bigquery google-gene
# Install langgraph and the packages used in this lab.
!pip install -qU 'langgraph==0.3.21' 'langchain-google-genai==2.1.2' 'langgraph-prebuilt==0.1.7'
```

```
WARNING: Skipping kfp as it is not installed.
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.5/43.5 kB 1.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 138.0/138.0 kB 5.2 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.0/42.0 kB 2.5 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.4/1.4 MB 27.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 433.9/433.9 kB 24.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.0/42.0 kB 2.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 47.2/47.2 kB 3.2 MB/s eta 0:00:00
```