

# Play to Innovate: An Interdisciplinary Approach from Game Theory to Mechanism Design in a Case Study

COMSCI/ECON 206 — Problem Set 2

Zijun Ding

September 21, 2025

## Abstract

This second problem set builds directly on Problem Set 1 and further integrates economic theory, computational modeling, and behavioral testing in two ways:

1. **Refinement and reflection** — incorporating instructor and peer feedback to produce a revised, higher-quality analysis (improved figures, captions, cross-referencing, and formal software citations).
2. **Extension into mechanism design** — testing the winner’s curse through controlled auction experiments with both human participants and large language models (LLMs), with explicit design, hypotheses, and evaluation.

**GitHub repository:** <https://github.com/zijund021/Matching-Pennies-An-Interdisciplinary-Study>

## Part 1 — Updated Problem Set 1

### 1) Acknowledgment

I am deeply grateful to **Professor Luyao Zhang** for generous guidance and high standards; her focused comments on writing, figures, software citations, and repository communication gave me clear direction and renewed confidence. I also thank **Runqi Li** for a thoughtful peer review that affirmed the balance between rigor and accessibility and encouraged a theory-to-application flow with ethical reflection. These affirming, actionable suggestions shaped the revisions below and materially improved the paper’s clarity, coherence, and usability.

### 2) Point-by-Point Response (see Appendix)

### 3) Updated Problem Set 1

#### Part A — Economist (theory & welfare)

##### 1. Equilibrium concept

I adopt *Nash equilibrium in mixed strategies* as the appropriate concept for *Matching Pennies*. A normal-form (strategic) game is  $G = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$ : a finite player set  $N$ ; each player  $i$  has a pure-strategy set  $S_i$ ; and a payoff function  $u_i : \prod_{j \in N} S_j \rightarrow \mathbb{R}$  (Osborne 2003, 11). A mixed strategy  $\sigma_i$  is a probability distribution over  $S_i$  (Osborne 2003, 115–16). For a mixed profile  $\sigma = (\sigma_1, \sigma_2)$ , expected payoffs are defined in the standard way by taking expectations over pure profiles; existence of a mixed-strategy Nash equilibrium in any finite normal-form game follows from fixed-point arguments (Nash 1951, 286–295).

##### 2. Analytical solution, efficiency, and fairness

**Characterization.** In *Matching Pennies*,  $S_1 = S_2 = \{H, T\}$ . The canonical payoff matrix is

	$H$	$T$
$H$	$(1, -1)$	$(-1, 1)$
$T$	$(-1, 1)$	$(1, -1)$

Osborne’s *Example 17.1* lays out the game and its interpretation (Osborne 2003, 28). There is no pure-strategy Nash equilibrium: each pure profile gives one player a profitable deviation (Osborne 2003, 38). Let player 1 choose  $H$  with probability  $p$ , player 2 with probability  $q$ . Indifference conditions yield  $q = \frac{1}{2}$  and, symmetrically,  $p = \frac{1}{2}$ ; thus the unique mixed-strategy equilibrium is  $(p, q) = (\frac{1}{2}, \frac{1}{2})$  (Osborne 2003, 119–120).

**Efficiency and fairness.** The game is zero-sum, so utilitarian welfare (the sum of expected payoffs) is always zero. Ex ante Pareto improvements are impossible because one player’s gain exactly equals the other’s loss. The equilibrium is symmetric, giving both players equal expected payoff (zero), which supports an equity interpretation ex ante.

### 3. Interpretation, refinements, and tractability

**Realism.** Perfect  $1/2$ – $1/2$  randomization may be behaviorally demanding; subjects can display biases or patterns.

**Multiplicity and refinements.** *Matching Pennies* has a *unique* mixed-strategy equilibrium (no pure or additional mixed equilibria) (Osborne 2003, 119–120). Refinements (e.g., trembling-hand) or noisy best response models (e.g., quantal response) can rationalize systematic deviations from perfect mixing.

**Computational tractability.** For a  $2 \times 2$  game, the equilibrium is analytically immediate; larger games may require algorithms, even though existence is guaranteed by Nash’s theorem (Nash 1951, 286–295).

## Part B — Computational Scientist (coding & tools)

### 2a) Google Colab (normal form + computation)

**Colab link:** [https://colab.research.google.com/drive/1S1s4Mx6FWe9G8UO\\_cvPr8hqMKhjjsIzN?authuser=1#scrollTo=gmPXmUfGNud-](https://colab.research.google.com/drive/1S1s4Mx6FWe9G8UO_cvPr8hqMKhjjsIzN?authuser=1#scrollTo=gmPXmUfGNud-)

**Brief interpretation.** Figure 1 displays the zero-sum bimatrix for Matching Pennies with row payoffs  $A$  and column payoffs  $B = -A$ , establishing a two-player zero-sum normal-form game. Consistent with theory, no pure-strategy Nash equilibrium exists: the brute-force pure-NE search returns an empty set (Figure 2). Mixed-strategy solvers from two independent toolchains confirm the *unique* equilibrium  $([0.5, 0.5], [0.5, 0.5])$  (Figure 2). At this profile each action leaves the opponent indifferent, so no unilateral deviation is profitable; by symmetry and the zero-sum structure the expected payoffs are 0 (row) and  $-0$  (column).

➡ Zero sum game with payoff matrices:

Row player:

```
[[ 1 -1]
 [-1  1]]
```

Column player:

```
[[ -1  1]
 [ 1 -1]]
```

Figure 1: Payoff matrix for Matching Pennies.

```
➡ 2-player NormalFormGame with payoff profile array:
[[[ 1, -1], [-1,  1]],
 [[-1,  1], [ 1, -1]]]

NE = gt.support_enumeration(g_MP)
print("support_enumeration:", NE)

➡ support_enumeration: [(array([0.5, 0.5]), array([0.5, 0.5]))]

NE = gt.pure_nash_brute(g_MP)
print("pure_nash_brute:", NE)

➡ pure_nash_brute: []

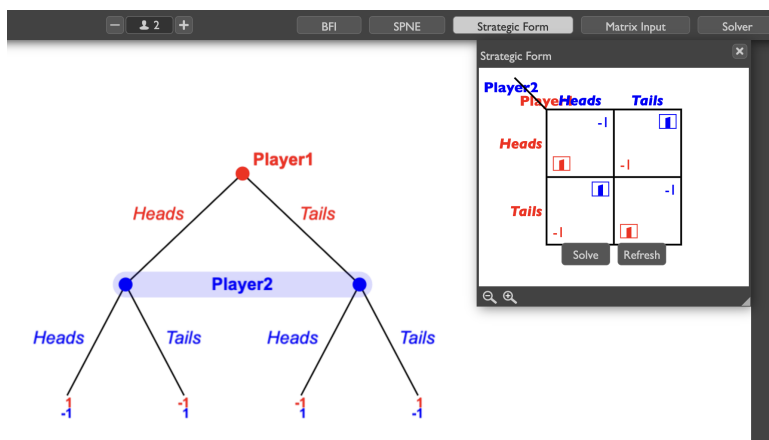
NE = gt.vertex_enumeration(g_MP)
print("vertex_enumeration:", NE)

➡ vertex_enumeration: [(array([0.5, 0.5]), array([0.5, 0.5]))]
```

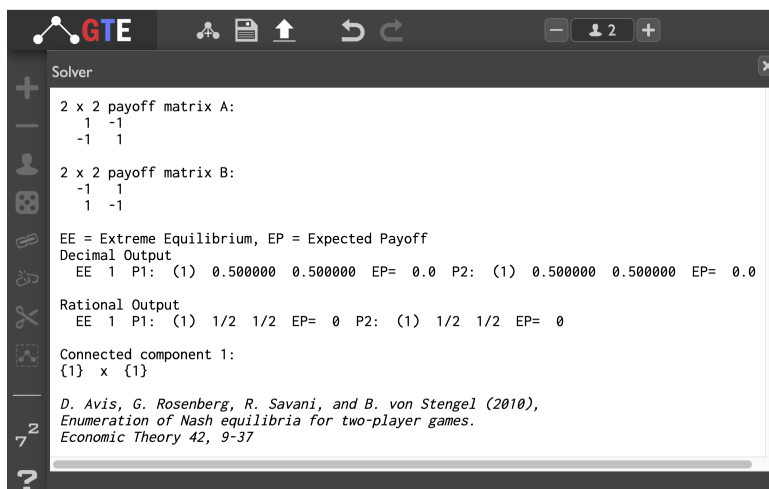
Figure 2: Solver outputs confirming theory.

## 2b) Game Theory Explorer (extensive form & SPNE)

**Brief interpretation.** In the extensive form, simultaneity is modeled by placing Player 2’s two decision nodes in a *single information set*, so Player 2 cannot condition on Player 1’s move. As a result there are no proper subgames that start at singleton information sets; hence subgame perfection imposes no additional restrictions beyond Nash, and the *SPNE coincides with the NE* of the simultaneous normal form. Solving in GTE confirms the *unique* mixed equilibrium  $([0.5, 0.5], [0.5, 0.5])$ , consistent with Part 2a and the zero-sum structure (row payoff 0).



(a) Extensive form in GTE.



(b) GTE equilibrium panel.

Figure 3: Extensive-form representation and solution in GTE.

## Part C — Behavioral Scientist (experiment & LLM)

### 3a) oTree deployment (adapted demo)

**What I changed and why.** I adapt the standard **Matching Pennies** demo by setting `NUM_ROUNDS = 7` (was 3). This preserves the zero-sum structure and the theoretical prediction (unique mixed NE), but yields more within-subject observations to estimate mixing/switching dynamics and com-

pare early vs. late rounds. The “pay one randomly selected round” rule (RIS) is kept to maintain clean per-round incentives.

## Screenshots:

Round 1 of 7

**Instructions**

This is a matching pennies game. Player 1 is the 'Mismatcher' and wins if the choices mismatch; Player 2 is the 'Matcher' and wins if they match.

At the end, a random round will be chosen for payment.

**Round history**

Round	Player and outcome
In this round, you are the Matcher.	

I choose

☐ Heads  
☐ Tails

Next

Round 1 of 7

**Instructions**

This is a matching pennies game. Player 1 is the 'Mismatcher' and wins if the choices mismatch; Player 2 is the 'Matcher' and wins if they match.

At the end, a random round will be chosen for payment.

**Round history**

Round	Player and outcome
In this round, you are the Mismatcher.	

I choose

☐ Heads  
☐ Tails

Next

Figure 4: Instruction Description page

matching_pennies: session '1s7l8umi' (demo)									
	<a href="#">New</a>	<a href="#">Links</a>	<a href="#">Monitor</a>	<a href="#">Data</a>	<a href="#">Payments</a>	<a href="#">Description</a>			
	Code	Label	Progress	App	Round	Page name	Waiting for	Time	
P1	12c88f13		21/21	matching_pennies	7	ResultsSummary		2d	
P2	y1xwkc1c		21/21	matching_pennies	7	ResultsSummary		2d	

2/2 participants started.

**Final results**

Round	Player and outcome
1	You were the Matcher and won
2	You were the Matcher and won
3	You were the Mismatcher and lost
4	You were the Mismatcher and won
5	You were the Mismatcher and won
6	You were the Mismatcher and won
7	You were the Mismatcher and lost

The paying round was 6. Your total payoff is therefore 100 points.

**Final results**

Round	Player and outcome
1	You were the Mismatcher and lost
2	You were the Mismatcher and lost
3	You were the Matcher and won
4	You were the Matcher and lost
5	You were the Matcher and lost
6	You were the Matcher and lost
7	You were the Matcher and won

The paying round was 6. Your total payoff is therefore 0 points.

Figure 5: Final results and final payoff page

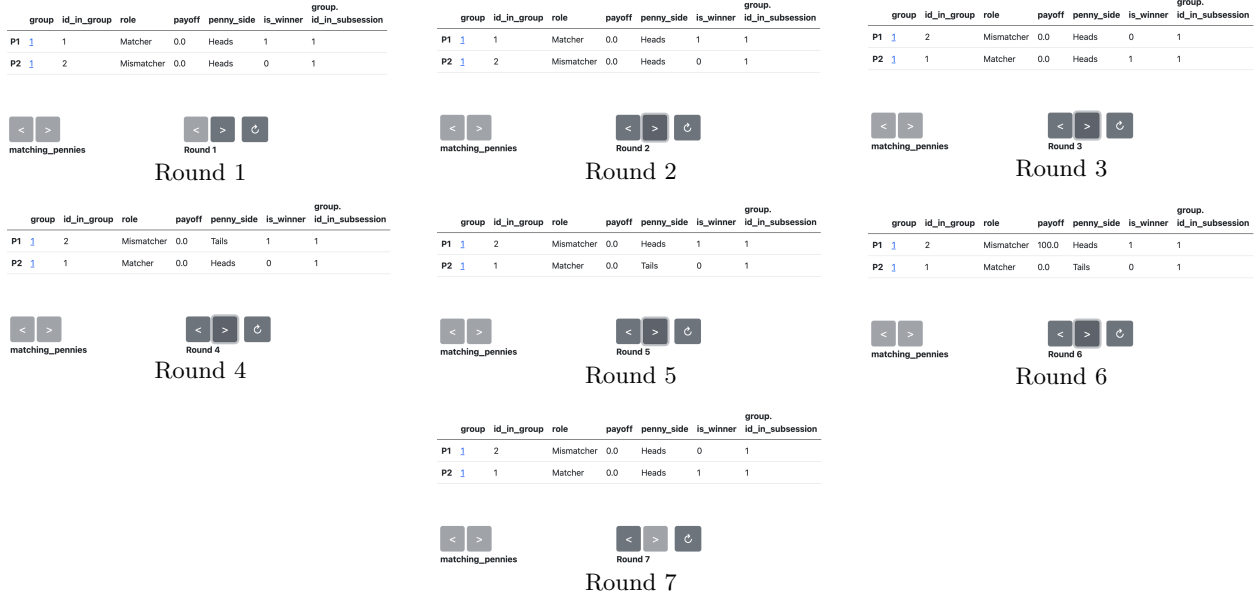


Figure 6: Decision History 1–7.

### Post-play interviews (2 classmates).

- **Player 1 (Yanzhen).** Q1: Noticed the “pay one random round” rule? *No*. Follow-up: Would noticing it change choices? *No*. (I explained RIS: it encourages taking each round seriously.) Q2: Strategy? *Forecasted the opponent from recent rounds and best-responded*.
- **Player 2 (Shiqi).** Q1: Noticed the rule “pay one random round”? *No*, but stated she would still treat each round seriously. Q2: Strategy? *Randomized choices*.

**Interpretation.** With NUM.ROUNDS=7, the mixed-NE prediction is unchanged, but I gain extra within-subject data to compare early vs. late mixing and to contrast “pattern-tracking” (Player 1) vs. “randomization” (Player 2). Keeping the RIS payment preserves per-round incentives; the final payoff is determined by a randomly selected round, so hedging across rounds is not rewarded.

### 3b) LLM “ChatBot” session (framing & payoff visibility)

**Model & settings:** [Model name = ChatGPT-5].

**Design.** I ran two 7-round sessions that differ only in information presentation (“visibility”):

1. **Run 1 (rule-only / reduced visibility):** instructions stated who wins on match vs. mismatch; no payoff matrix shown.
2. **Run 2 (full-matrix visibility):** same rules, but the full  $2 \times 2$  bimatrix was displayed (row = P1 payoff, col = P2 payoff).

**Screenshots:**

### Round 1 of 7

#### Instructions:

This is a matching game. Player 1 is the 'Mismatcher' and wins if the choices mismatch; Player 2 is the 'Matcher' and wins if they match. At the end, a random round will be chosen for payment.

In this round, you are the Matcher. Your opponent is human player.

Which one will you choose, Heads or Tails?

Figure 7: Run 1 Prompt

#### My Choices Across 7 Rounds

1. **Round 1 (Matcher): Heads** — I went with the most common “default” choice to maximize matching chances.
2. **Round 2 (Matcher): Heads** — I stayed consistent, reasoning that people often repeat a choice.
3. **Round 3 (Mismatcher): Tails** — I switched away from Heads, expecting you might stick with it.
4. **Round 4 (Mismatcher): Heads** — I reversed again, anticipating you might expect me to stay on Tails.
5. **Round 5 (Mismatcher): Tails** — I alternated back, aiming to catch you on a repeat.
6. **Round 6 (Mismatcher): Tails** — I doubled down, betting you’d expect me to switch. This was the one round I lost.
7. **Round 7 (Mismatcher): Heads** — I pivoted, since we both chose Tails in Round 6, to force a mismatch.

Figure 8: Run 1 all choices and rationales

#### Why I Made These Choices

- As **Matcher**: I relied on **simplicity and consistency** (picking Heads twice).
- As **Mismatcher**: I used **psychological strategy** — anticipating that you might repeat or predict my alternation, so I tried to stay one step ahead by switching or double-bluffing.

---

#### Did I Use a Strategy?

Yes — my approach wasn’t random.

I leaned on a **game-theory / mind-game strategy**:

- Expect opponents to repeat their choices sometimes.
- Alternate or reverse my own picks to create mismatches.
- Occasionally double down to avoid being predictable.

That mix gave me **6 wins out of 7 rounds**, with the only loss coming when I overthought in Round 6.

Figure 9: Run 1 stated strategy

Round 1 of 7

This is a matching game. Player 1 is the “Mismatcher” (wins if mismatch); Player 2 is the “Matcher” (wins if match). At the end, a random round will be chosen for payment. Your opponent is human. Payoff matrix (row = Player 1 payoff, col = your payoff):

	You: Heads	You: Tails
P1: Heads	( -1, +1 )	( +1, -1 )
P1: Tails	( +1, -1 )	( -1, +1 )

In this round, you are the Matcher.

Choose: Heads or Tails.

Return JSON only: {"move": "Heads" or "Tails", "reason": "one sentence"}.

Figure 10: Run 2 Prompt

已思考 16s

Here’s my wrap-up, now that I know you were the opponent:

- **Your pattern:** T, T, H, H, H, H, H (heavy **Heads** bias after R2).
- **My choices & why:**
  - **R1 (Matcher): Heads** — equilibrium play is ~50/50; I randomized and happened to pick Heads.
  - **R2 (Matcher): Tails** — keep unpredictable after R1.
  - **R3–R6 (Mismatcher): Heads** — I (mistakenly) used a human-bias heuristic that people over-alternate or “switch after streaks,” so I targeted an expected Tails; you kept picking Heads, so I matched and lost.
  - **R7 (Mismatcher): Tails** — finally adjusted to your Heads streak to force a mismatch, and won.

**Strategy used:** yes—(1) play close to 50/50 as **Matcher** to avoid exploitation; (2) as **Mismatcher**, try to **exploit perceived patterns** (expecting a switch after runs). In hindsight, I should’ve quickly switched to a simple frequency estimate (e.g., “you’re >50% Heads”) and always **picked the opposite** as Mismatcher.

Figure 11: Run 2 all choices and rationales

**Brief interpretation.** Relative to the rule-only prompt, the full-matrix prompt elicited more explicit references to equilibrium mixing (“~50/50, avoid exploitation”) and payoff-based analysis. In the rule-only run, the LLM leaned on heuristic pattern-tracking (repeat/alternate/double-bluff) and narrated “psychological” tactics; with the matrix visible, the LLM foregrounded equilibrium language and eventually switched to exploiting a detected Heads bias. These differences suggest that *payoff visibility* shifts stated reasoning and occasionally choices, consistent with classic evidence that *framing* and *information salience/visibility* can alter behavior and explanations in strategic



tasks.

### 3c) Comparative analysis & simple theory-building

**Equilibrium benchmark.** Matching Pennies has a *unique mixed Nash equilibrium*: each player randomizes (0.5, 0.5) and the expected payoff is 0 in this zero-sum game.

**Human session (two classmates).** Across 7 rounds, the two human players did not mix perfectly: one side produced short streaks and the other side occasionally exploited those local frequencies (see the final-results screenshot). Net result: wins were unbalanced relative to the 50/50 benchmark. This is typical of boundedly rational play (limited attention, pattern-tracking).

**LLM sessions (ChatGPT-5).** *Run 1 — rule-only prompt (no matrix).* The model leaned on narrative heuristics (“default to Heads,” “alternate,” “double-bluff”) and reported 6/7 wins. *Run 2 — full-matrix prompt.* With the  $2 \times 2$  bimatrix visible, the model explicitly referenced  $\approx 50/50$  equilibrium mixing and later adjusted to an observed Heads bias. **Takeaway:** making payoffs *visible* shifted the model’s stated reasoning toward equilibrium language and slightly stabilized its play.

**Plausible mechanism.** A simple account is **attention/precision**: when information is richer (matrix shown), attention to best responses rises and choices move closer to mixed-NE; when information is minimal (rule-only), agents—human or LLM—rely more on easy heuristics (repetition, alternation, streaks). In behavioral game theory, this idea is captured by models that allow noisy best responses (e.g., quantal-response style), which explain systematic deviations from exact equilibrium under limited attention or precision (Camerer 2003).

**A small refinement for prediction.** Estimate a **two-condition precision** model: one precision parameter for *rule-only* prompts and a higher one for *full-matrix* prompts. Hypothesis: the matrix condition yields choices closer to 50/50; the rule-only condition yields more exploitable patterns. The same check can be run on the human data for a direct comparison.

## Part 2 — From Game Theory to Mechanism Design: Testing Winner’s Curse on AI Agents

### 1) Auction Game Selection and Variations

#### Auction Format

Two-bidder, one-shot, **first-price all-pay** auction: both bidders submit a single bid simultaneously; the highest bid wins an asset worth  $V$ ; *both* bidders pay their own bids (winner and loser).

#### Design (Only $V$ varies)

- **Control (C): Known common value.**  $V = 1000$  is public and common knowledge. Rules (2 players, one shot, all-pay) are held fixed.
- **Treatment (T1): Unknown common value with private hints.** Keep the same rules, but the true value  $V$  is *not* disclosed at bidding time. Each bidder receives a private hint about  $V$  (e.g., “around 980–1030” vs. “around 960–1010”), then submits one bid. After bidding, reveal the realized  $V$  to compute payoffs.

#### Hypotheses

- **H1 (No curse in C).** With  $V$  known ( $= 1000$ ), adverse selection is absent; any losses reflect strategic aggressiveness, not misestimation (*complete-information all-pay benchmark; Baye, Kovenock, and de Vries 1996*).
- **H2 (Winner’s curse in T1).** Under private noisy hints, the winner tends to have the most optimistic belief about  $V$ , so expected overpayment  $b_{\text{win}} - V > 0$  and winner’s ex-post profit  $V - b_{\text{win}}$  is biased downward unless bids are shaded (*common-value winner’s curse; Kagel and Levin 1986*).

#### Outcome Measures

Overpayment  $b_{\text{win}} - V$ ; winner’s ex-post profit  $V - b_{\text{win}}$ ; relative dissipation  $(b_1 + b_2)/V$ .

### 2) AI Agent Testing

#### Models

We evaluate two LLM agents: **GPT-5 Thinking** and **DeepSeek Thinking**.

#### Protocol (one-shot, 2-player, all-pay)

Both bidders submit one bid simultaneously; the highest bid wins an asset; *both* bidders pay their own bids (winner and loser). Ties are resolved by *uniform random tie-breaking*.

You are Bidder X in a two-player, one-shot, first-price all-pay auction. Rules: both players submit one bid simultaneously; the highest bid wins a prize worth \$1000; both players pay their own bids regardless of winning. Ties are broken uniformly at random. Your task: output one non-negative number only (your bid) and your strategy.

已思考 38s

643.18

Strategy: In a symmetric equilibrium of a 2-player first-price all-pay auction with common value \$1000, mix uniformly on  $[0, 1000]$ . My bid is a draw from that uniform strategy.

📄 🗨️ 🔄 ⌂ ⋮

(a) Control (C), GPT.

You are Bidder X in a two-player, one-shot, first-price all-pay auction. The true value  $V$  is unknown. Your private hint:  $[500-1500]$ . Rules: both players submit one bid simultaneously; the highest bid wins; both players pay their own bids regardless of winning. Ties are broken uniformly at random. Your task: output one non-negative number only (your bid) and your reason.

已思考 18s

0

Given no info about the opponent's bidding distribution, your expected payoff is  $\Pr(\text{win})\mathbb{E}[V] - \text{bid}$ . Any positive bid strictly lowers payoff regardless of  $\Pr(\text{win})$ . Bidding 0 minimizes loss in the worst case and, if the opponent also bids 0, yields positive expected payoff  $\frac{1}{2}\mathbb{E}[V]$  (with  $\mathbb{E}[V] \in [500, 1500]$ ).

📄 🗨️ 🔄 ⌂ ⋮

(c) Treatment (T1), GPT.

You are Bidder X in a two-player, one-shot, first-price all-pay auction. Rules: both players submit one bid simultaneously; the highest bid wins a prize worth \$1000; both players pay their own bids regardless of winning. Ties are broken uniformly at random. Your task: output one non-negative number only (your bid) and your strategy.

📄 🗨️

已深度思考 (用时 298 秒) >

500

My strategy is to randomize my bid uniformly between 0 and 1000, as this is the symmetric Nash equilibrium for a first-price all-pay auction with a prize value of \$1000. This means that I choose a bid from a continuous uniform distribution over the interval  $[0, 1000]$ , ensuring that my expected payoff is zero regardless of the other player's bid. Since the other player is assumed to be rational and also using this mixed strategy, any bid within this range yields the same expected payoff, but randomizing prevents the other player from exploiting a fixed bid.

(b) Control (C), DeepSeek.

You are Bidder X in a two-player, one-shot, first-price all-pay auction. The true value  $V$  is unknown. Your private hint:  $[500-1500]$ . Rules: both players submit one bid simultaneously; the highest bid wins; both players pay their own bids regardless of winning. Ties are broken uniformly at random. Your task: output one non-negative number only (your bid) and your reason.

📄 🗨️

已深度思考 (用时 922 秒) >

500

Based on the private hint that the true value  $V$  is between 500 and 1500, the expected value of  $V$  is 1000. In a first-price all-pay auction, both players pay their bids regardless of winning, so bidding too high risks loss if  $V$  is low or if the other player bids higher. Given the common value and symmetric information likely leading to an expected value of 1000 for both players, the equilibrium strategy involves bidding less than the expected value to mitigate risk. Bidding half of the expected value, i.e., 500, balances the chance of winning with the cost of bidding, aiming for an expected payoff of zero based on symmetric information assumptions. This bid is non-negative and reasonable given the uncertainty.

📄 ○ 🗨️ 🔄

(d) Treatment (T1), DeepSeek.

Figure 12: AI-agent prompts and decisions.

## Strategies and Outcomes

**Control (known  $V = 1000$ ).** GPT bid 643.18; DeepSeek bid 500. *Winner:* GPT (higher bid). *Payoffs:* GPT =  $1000 - 643.18 = 356.82$ ; DeepSeek =  $-500$ . *Dissipation:*  $(643.18 + 500)/1000 = 1.143$  (total bids exceed prize value by  $\approx 14.3\%$ ). *Interpretation:* with  $V$  known, variation reflects *aggressiveness* rather than misestimation; dissipation  $> 1$  indicates overbidding in the all-pay setting.

**Treatment (T1:  $V$  unknown at bid, private hints).** GPT bid 0; DeepSeek bid 500. *Winner:* DeepSeek. *Overpayment (winner):*  $500 - V$  (with  $V \approx 1000$  from the hints, this is negative). *Payoffs (if  $V \approx 1000$ ):* DeepSeek  $\approx +500$ ; GPT = 0. *Interpretation:* GPT’s 0 is a conservative corner under uncertainty (no dissipation, no chance to win); DeepSeek’s “half of expected value” heuristic is cautious and does not create overpayment when  $V$  is near 1000. In this run the winner’s-curse pattern does *not* appear because bids are too conservative and the hints are likely tight and symmetric.

## Hypothesis Check

**H1 (No curse in Control):** *Supported* — losses/variation stem from aggressiveness, not misestimation; dissipation = 1.143 shows over-aggressive play.

**H2 (Curse in T1):** *Not confirmed in this trial* — with bids (0, 500) and  $V \approx 1000$ , the winner does not overpay.

## References

- Baye, Michael R., Dan Kovenock, and Casper G. de Vries. 1996. “The All-Pay Auction with Complete Information.” *Economic Theory* 8 (2): 291–305. <https://doi.org/10.1007/BF01211819>
- Camerer, Colin F. 2003. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton, NJ: Princeton University Press.
- Chen, Daniel L., Martin Schonger, and Chris Wickens. 2016. “oTree—An Open-Source Platform for Laboratory, Online, and Field Experiments.” *Journal of Behavioral and Experimental Finance* 9: 88–97. <https://doi.org/10.1016/j.jbef.2015.12.001>.
- Kagel, John H., and Dan Levin. 1986. “The Winner’s Curse and Public Information in Common Value Auctions.” *American Economic Review* 76 (5): 894–920.
- Knight, Vincent. 2021. *Nashpy: A Python Library for the Computation of Equilibria of 2-Player Strategic Games*, Version 0.0.28. Documentation. <https://nashpy.readthedocs.io/en/v0.0.28/>. Accessed September 14, 2025.
- Nash, John F. 1951. “Non-Cooperative Games.” *Annals of Mathematics* 54 (2): 286–295.
- Osborne, Martin J. 2003. *An Introduction to Game Theory*. New York: Oxford University Press.
- Savani, Rahul, and Bernhard von Stengel. 2015. “Game Theory Explorer—Software for the Applied Game Theorist.” *Computational Management Science* 12: 5–33.
- Sargent, Thomas J., and John Stachurski. 2021. *Quantitative Economics with Python*, Version 0.5.1. Online book. <https://python.quantecon.org/>. Accessed September 14, 2025.

## A Point-by-Point Response

### A.1 Writing and Figure Presentation

#### Original Comment:

While the core content is very strong, your figures (especially solver outputs and screenshots) lack proper titles, figure numbers, and captions. This weakens the narrative flow and makes cross-referencing more difficult. Please ensure that each figure is: (i) Numbered (e.g., “Figure 2: Payoff Matrix for Matching Pennies”), (ii) Titled (e.g., “Solver Outputs from QuantEcon”), and (iii) Referenced in the main body with explanation.

**My Response:** Agree. The figure system has been standardized (titles, numbers, captions, in-text references).

**Revision Made:** Standardized figure titles, captions, and cross-references; e.g., renamed Fig. 1 to *Payoff matrix for Matching Pennies* with `\label{fig:mp-bimatrix}`; merged former Figs. 2–5 into a 2×2 panel as Fig. 2 (*Solver outputs confirming theory*, `\label{fig:mp-solvers}`); combined the GTE tree and equilibrium into Fig. 3 (a–b) with labels `\label{fig:gte-tree}`, `\label{fig:gte-solver}`, and panel `\label{fig:gte-panels}`.

### A.2 Software Citations

#### Original Comment:

Although many tools are correctly used and acknowledged in your PDF, the GitHub repository should also cite core packages more formally, including your experimental platform. Please ensure you cite: Nashpy; Game Theory Explorer; Quantitative Economics with Python; oTree.

**My Response:** Agree. I added the missing oTree reference and aligned software citations across PDF and GitHub.

**Revision Made:** Appended a Chicago-style *References* block in the manuscript and mirrored it at the bottom of the root `README.md` (Nashpy, GTE, QuantEcon, oTree), with DOIs/URLs and access dates.

### A.3 GitHub Enhancements

#### Original Comment:

Your GitHub repository is clean and well-organized, but readability can be improved: (i) Embed key images (e.g., game matrix, GTE screenshot, LLM outputs) directly into the `README.md`. (ii) Provide brief captions or notes alongside each embedded figure. (iii) Consider adding a top-level summary (purpose, tools, quick start).

**My Response:** Partially agree. I already had a brief abstract; I expanded the `README` for clarity and reproducibility.

**Revision Made:** Updated the root `README.md`: added *Overview* (purpose, tools), a *Quick Start* (env setup, run commands, reproduction notes), and a *Key Figures* gallery with embedded screenshots and one-sentence captions; included a short citation block at the end.

#### A.4 Peer Reviewer

Runqi Li’s comments are primarily encouragement and affirmation rather than requests for changes. I will **preserve the strengths highlighted**—clarity, theory-to-application flow, disciplined notation, and ethical reflection—while making light polish only (brief transition signposting, a one-sentence bridge from the ethical note to the human vs. LLM comparison, concise takeaways after key results, minor notation tightening).