COMPSCI-ECON206 Problem Set 1

Ai Zhou

September 2025

1 Part 1

Subgame Perfect Nash Equilibrium

Definition (Paraphrase)

Let an extensive-form game with perfect information be denoted by

$$\Gamma = \langle N, H, P, (A(h))_{h\in H}, (u_i)_{i\in N}\rangle,$$

where

- $N = \{1, \ldots, n\}$ is the finite set of players.
- H is the set of histories (nodes), including the empty history $\emptyset$, with terminal histories $Z \subseteq H$.
- $P : H \setminus Z \to N$ assigns a player to each nonterminal history.
- $A(h)$ is the finite set of actions available after history h.
- Each player $i \in N$ has a payoff function $u_i : Z \to R$.
- A **strategy** s_i for player i is a complete contingent plan assigning an action in $A(h)$ for every history h with $P(h) = i$.
- A **strategy profile** is $s = (s_1, \ldots, s_n)$.

A **subgame** of Γ is defined as the restriction of Γ to any history h that constitutes a decision node not cutting across information sets.

[Subgame Perfect Nash Equilibrium] [pp. 93–95] Shoham2009, Rubinstein1994

A strategy profile s^* is a *Subgame Perfect Nash Equilibrium (SPNE)* if and only if, for every subgame $\Gamma(h)$ of Γ, the restriction $s^*|_h$ induces a Nash equilibrium of that subgame:

$$u_i(s^*|_h) \ \geq \ u_i(s_i, s^*_{-i}|_h), \quad \forall i \in N,\ \forall s_i \in S_i(h),\ \forall h \in H.$$

That is, no player has a profitable deviation in any subgame, not only in the original game.

Existence Theorem (Paraphrased)

For any finite extensive-form game with perfect information, there exists at least one strategy profile $s^* = (s^*_1, \ldots, s^*_n)$ such that s^* forms a subgame perfect Nash equilibrium. Formally,

$$\exists s^* \in S_1 \times \ldots \times S_n such that s^*|_h is a Nash equilibrium for every subgame \Gamma(h),\ \forall h \in H.$$

Proof Idea

The proof uses **backward induction**:

1. Start from the terminal nodes and determine each player's optimal action at the last decision nodes.
2. Replace each subgame by the payoff vector resulting from optimal play.
3. Recursively move backward in the tree, selecting actions that maximize the player's payoff at each node.
4. The resulting strategy profile is optimal in every subgame and thus forms a SPNE.

Intuition: The finiteness of the game tree guarantees that this backward-induction construction produces at least one SPNE in pure strategies.

Analytical Solution and Interpretation

Analytical Solution

In this extensive-form game with perfect information, the Subgame Perfect Nash Equilibrium (SPNE) identifies strategies where each player optimally responds in every possible subgame. Conceptually, SPNE refines the standard Nash equilibrium by requiring that no player has an incentive to deviate, even after unexpected moves by others. We determine the equilibrium using backward induction: starting from the terminal nodes, each player chooses the action that maximizes their payoff given subsequent optimal decisions. This process is repeated recursively until reaching the initial node, producing a complete strategy profile that specifies an action for every decision point. For example, player 1's SPNE strategy can be written as $s_1^* = (a_1, a_2, \ldots)$, while player 2's is $s_2^* = (b_1, b_2, \ldots)$. This approach guarantees consistency across all subgames and provides a clear prediction for rational play.

The SPNE outcome is not necessarily socially optimal. From a Pareto perspective, some SPNE strategies may leave all players worse off than alternative cooperative outcomes. Similarly, total welfare (the sum of players' payoffs) might not be maximized, indicating suboptimal utilitarian efficiency. Sequential play can also generate unequal outcomes, raising fairness concerns: one player may consistently earn more than others, violating equity or proportionality principles. Nonetheless, SPNE serves as a normative benchmark: it shows what rational players would do if they perfectly anticipate others' behavior, even if the outcome is not efficient or fair.

Interpretation

In practice, the Subgame Perfect Nash Equilibrium (SPNE) provides a strong prediction for how fully rational players with perfect information would behave in every subgame of a sequential interaction. However, its realism is limited because actual human decision-makers often face cognitive constraints and incomplete information, which may lead to deviations from SPNE predictions. Furthermore, many extensive-form games admit multiple SPNE, raising the question of equilibrium selection; observed behavior may depend on social norms, expectations, or pre-play communication. Refinements such as trembling-hand perfect equilibrium help address implausible strategies by eliminating those that rely on extremely unlikely mistakes. From a computational perspective, backward induction guarantees SPNE existence in finite games, but as the game tree grows in size or complexity, calculating SPNE becomes increasingly demanding, highlighting the practical relevance of algorithmic tools like GTE or NashPy. Thus, while SPNE offers a normative benchmark, its predictive accuracy is influenced by both bounded rationality and computational tractability.

2 Part 2 Computational ScientistsTrust Simple

u(A) = 100 – x + y

u(B) = 3x - y

Player A/B	0 %	50 %	100 %
0	(100,0)	(100,0)	(100,0)
50	(50,100)	(125,75)	(200,0)
100	(0,300)	(150,150)	(300,0)

Figure 1: Payoff matrix, made by Microsoft Word, exported as png.

```
Requirement already satisfied: nashpy in /usr/local/lib/python3.12/dist-packages (0.0.41)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.12/dist-packages (from nashpy) (2.0.2)
Requirement already satisfied: scipy>=0.19.0 in /usr/local/lib/python3.12/dist-packages (from nashpy) (1.16.1)
Requirement already satisfied: networkx>=3.0.0 in /usr/local/lib/python3.12/dist-packages (from nashpy) (3.5)
Requirement already satisfied: deprecated>=1.2.14 in /usr/local/lib/python3.12/dist-packages (from nashpy) (1.2.18)
Requirement already satisfied: wrapt<2,>=1.10 in /usr/local/lib/python3.12/dist-packages (from deprecated>=1.2.14->nashpy) (1.17.3)
Normal-form Trust Game (Multiplier=3):
Bi matrix game with payoff matrices:

Row player:
[[100 100 100]
 [ 50 125 200]
 [  0 150 300]]

Column player:
[[  0   0   0]
 [150  75   0]
 [300 150   0]]

Nash Equilibria (pure and mixed strategies):
Player A strategy: [1. 0. 0.]
Player B strategy: [1. 0. 0.]
---
```

Figure 2: Nash Equilibria calculated by Nashpy using Google colab.

Interpretation (Google Colab)

The NashPy computation returns a pure strategy Nash Equilibrium where Player A chooses to invest 0 and Player B chooses to return 0. This means that, given Player B's strategy of returning nothing, Player A maximizes their payoff by investing nothing. Similarly, given Player A's investment of 0, Player B cannot improve their payoff by returning any positive amount. This result is fully consistent with the Subgame Perfect Nash Equilibrium (SPNE) derived via backward induction in the extensive-form game. In the one-round Trust Game, Player B's optimal action in every subgame is to return 0, and anticipating this, Player A's optimal choice is also to invest 0. Therefore, the SPNE coincides with the pure-strategy NE identified by NashPy.

Game Theory Explorer(GTE)

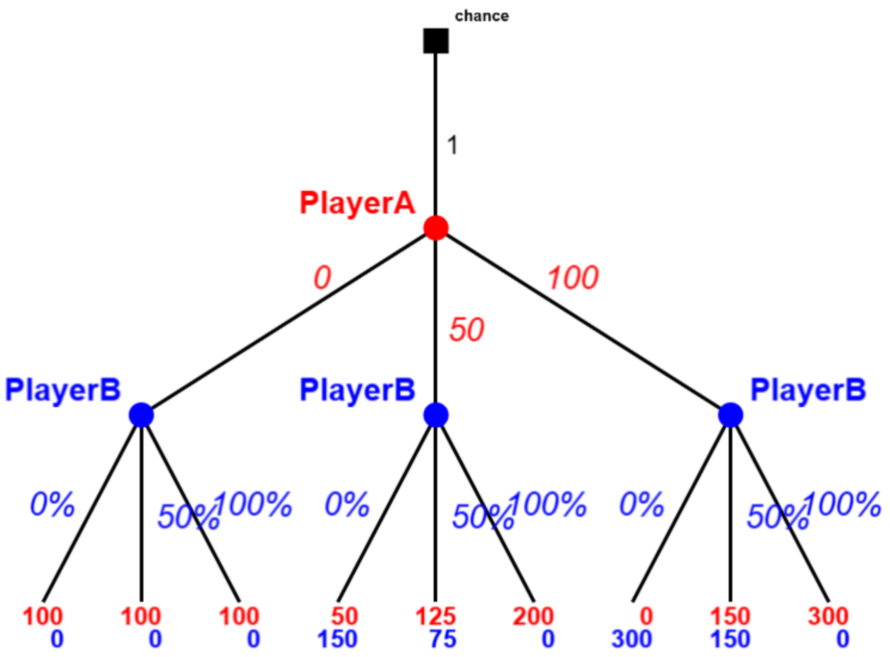


Figure 3: Extensive-form version of Trust Simple in GTE.

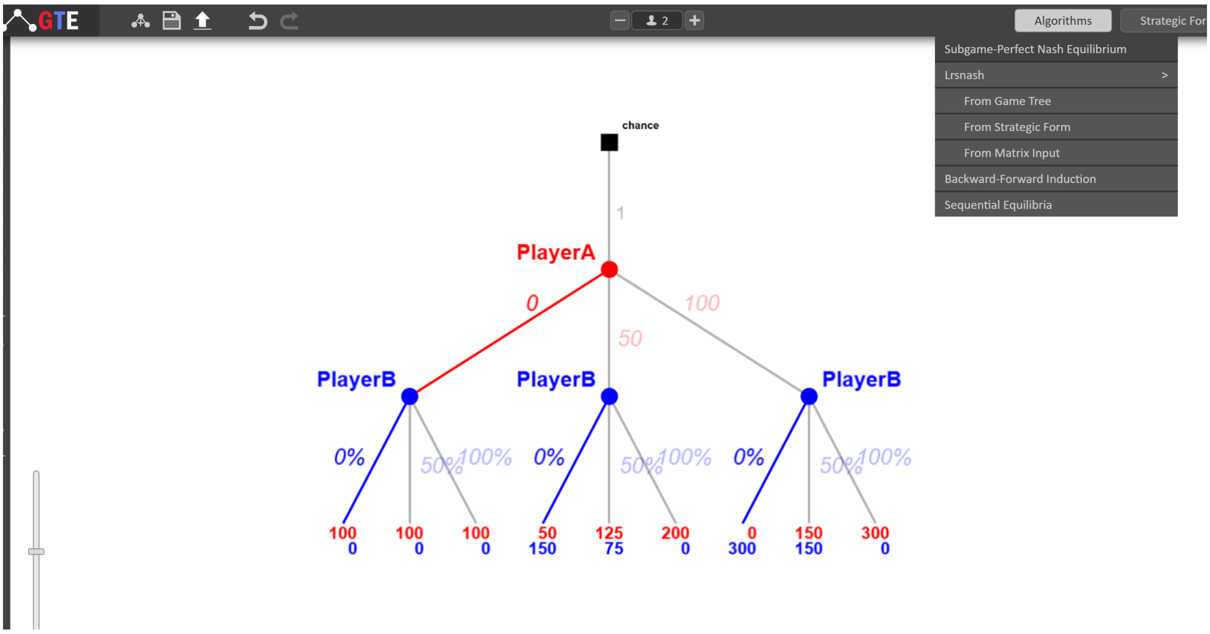
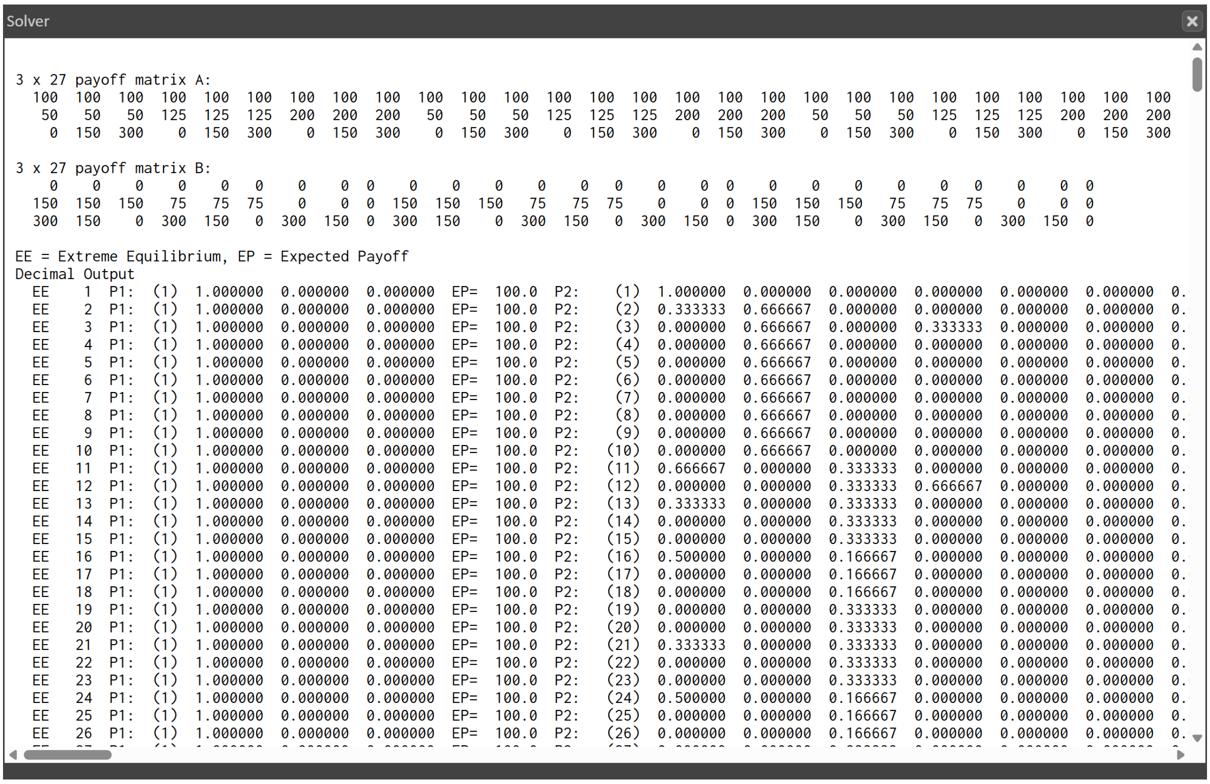


Figure 4: Solving the extensive-form version of Trust Simple with SPNE in GTE.



```
Solver

3 x 27 payoff matrix A:
  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100
   50   50   50  125  125  125  200  200  200   50   50   50  125  125  125  200  200  200   50   50   50  125  125  125  200  200  200
    0  150  300    0  150  300    0  150  300    0  150  300    0  150  300    0  150  300    0  150  300    0  150  300    0  150  300

3 x 27 payoff matrix B:
    0    0    0    0    0    0    0    0 0    0    0    0    0    0  0    0    0 0    0    0    0    0    0  0    0    0 0
  150  150  150   75   75  75    0    0 0  150  150  150   75   75 75    0    0 0  150  150  150   75   75 75    0    0 0
  300  150    0  300  150   0  300  150 0  300  150    0  300  150  0  300  150 0  300  150    0  300  150  0  300  150 0

EE = Extreme Equilibrium, EP = Expected Payoff
Decimal Output
  EE    1  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (1)  1.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.
  EE    2  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (2)  0.333333  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    3  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (3)  0.000000  0.666667  0.000000  0.333333  0.000000  0.000000  0.
  EE    4  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (4)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    5  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (5)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    6  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (6)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    7  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (7)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    8  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (8)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    9  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (9)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE   10  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (10)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE   11  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (11)  0.666667  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   12  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (12)  0.000000  0.000000  0.333333  0.666667  0.000000  0.000000  0.
  EE   13  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (13)  0.333333  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   14  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (14)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   15  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (15)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   16  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (16)  0.500000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
  EE   17  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (17)  0.000000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
  EE   18  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (18)  0.000000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
  EE   19  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (19)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   20  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (20)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   21  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (21)  0.333333  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   22  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (22)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   23  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (23)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   24  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (24)  0.500000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
  EE   25  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (25)  0.000000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
  EE   26  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:  (26)  0.000000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
```

Figure 5: Solving the extensive-form version of Trust Simple with using payoff matrix in GTE, the solving process is a bit complicated.

SPNE and Its Relation to Part 1 and Normal Form

The Subgame Perfect Nash Equilibrium (SPNE) computed in GTE confirms the theoretical analysis presented in Part 1. In the one-round Trust Game, backward induction shows that Player B's optimal action in every subgame is to return 0, and anticipating this, Player A optimally invests 0. This outcome matches the SPNE identified in the extensive-form game.

When we translate the game into simultaneous normal form, as done with the NashPy payoff matrices, the same equilibrium emerges: the pure-strategy Nash Equilibrium is for Player A to invest 0 and Player B to return 0. In other words, the SPNE of the extensive-form game corresponds exactly to the NE of the simultaneous normal-form representation. This illustrates that, for a one-shot, perfect-information Trust Game, SPNE and pure-strategy NE are equivalent, and both capture the strategic incentives of the players consistently.

3 Part 3 Behavioral Scientist (experiment AI comparison)

3.1 3(a) oTree deployment

In this demo, I changed the initial endowment for PlayerA from 10 to 100. This adjustment makes the stakes more substantial and allows clearer differentiation in payoffs when multiplied and returned by Player B, which is especially helpful when observing behavior in both human and LLM sessions. No other structural changes were made: the game still involves two players, one round of play, and a multiplier of 3 for invested amounts. It is necessary to ensure that the numerical outcomes are more intuitive and interpretable, facilitating easier analysis of deviations from the Subgame Perfect Nash Equilibrium (SPNE) and comparison between human and AI behavior.

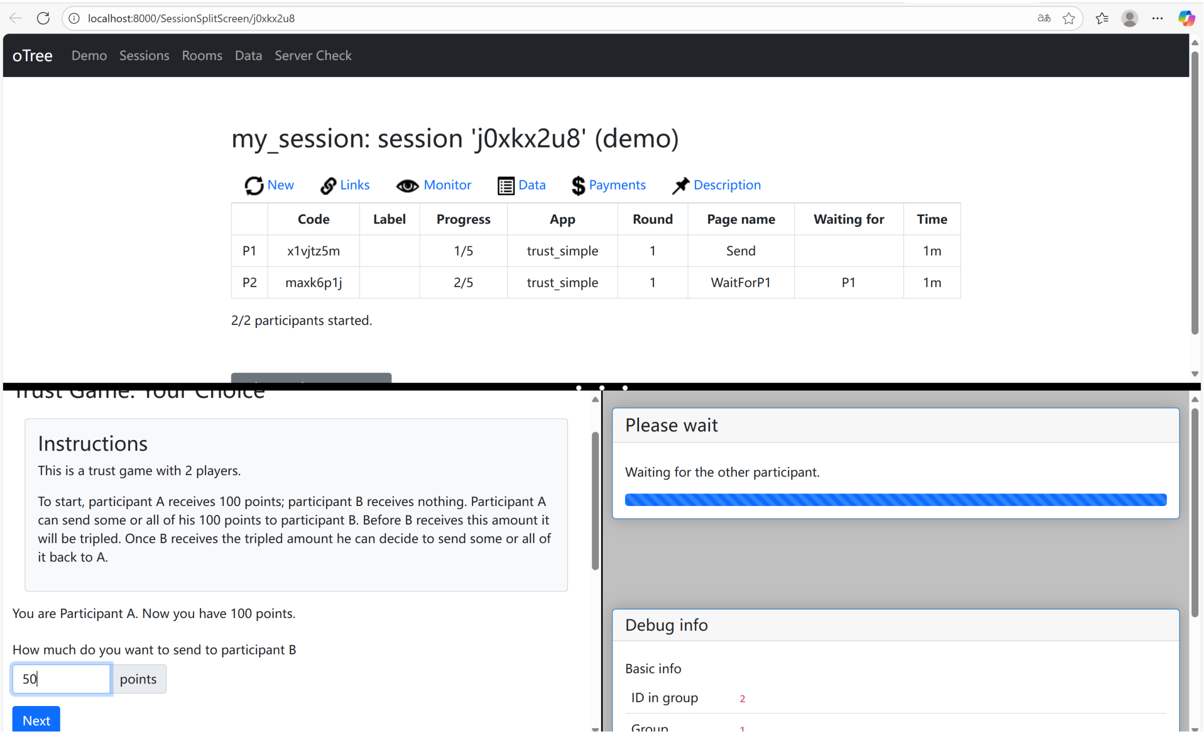


Figure 6: Gameplay: Trust Simple (using otree, step 1).

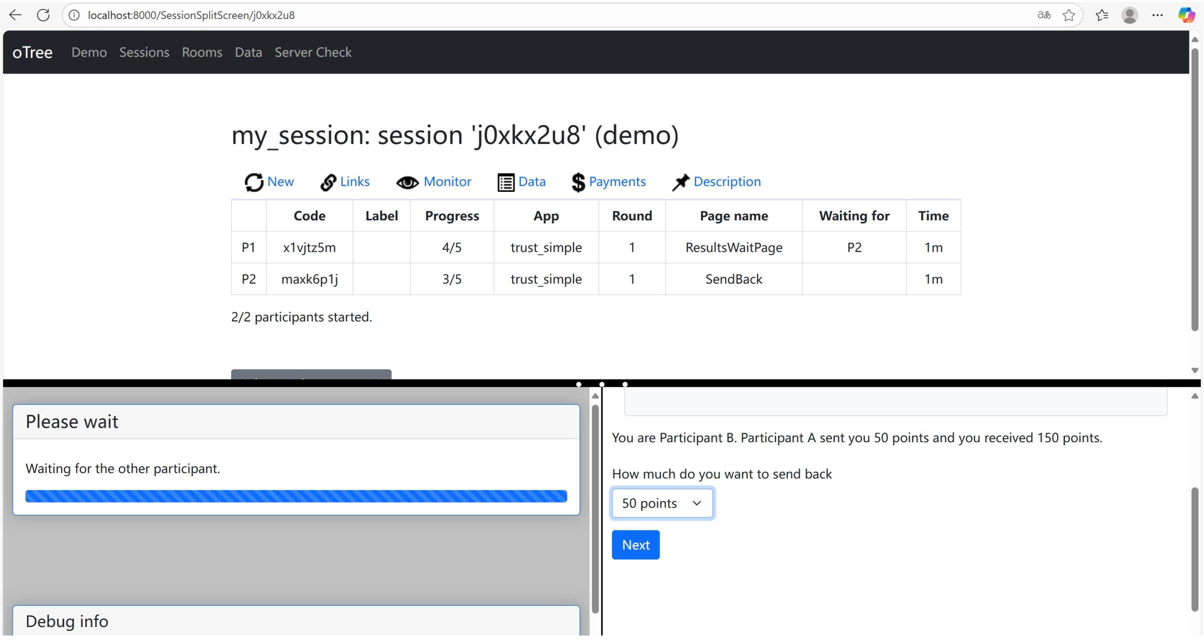


Figure 7: Gameplay: Trust Simple (using otree, step 2).

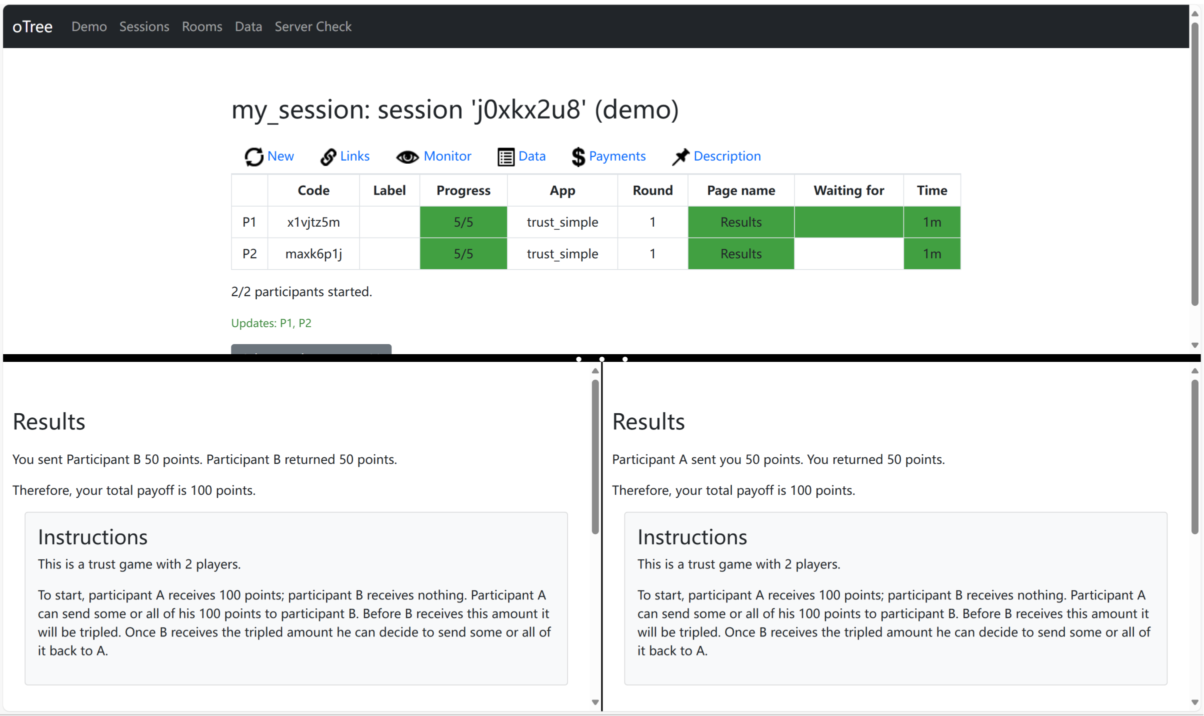


Figure 8: Gameplay: Trust Simple (using otree, step 3).

Post-play Interview Summary: Player A (Yihan) chose a moderate investment, expressing caution and some trust toward Player B. Player B (Ji Wu) returned one third the multiplied amount, citing some fairness and reciprocity considerations. Overall, participants' choices deviated from the SPNE prediction due to trust and social preference factors.

3.2 (3b) LLM "ChatBot" session

Trust Game Session Summary

Experiment Setup: One-round Trust Game, Multiplier = 3. LLM plays as either Player A or Player B depending on round.

Prompt Template for LLM (Player B):

```
You are Player B in a one-round Trust Game.
Player A invested X.
Multiplier is 3. Choose how much to return to Player A (0, 50%, or 100%) and explain your reasoning.
```

LLM Session Rounds:

The table below summarizes the five rounds of the Trust Game played between human and AI(ChatGPT) participants:

Round	Player A Investment	Player B Return	Player A Payoff	Player B Payoff
1	50	75	125	75
2	0	0	100	0
3	50 (AI as A)	0	50	150
4	100 (AI as A)	0	0	300
5	0 (AI as A)	0	100	0

Table 1: Summary of Trust Game rounds showing investments, returns, and payoffs.

Observations: Across the five rounds, the LLM's behavior showed both rational and fairness-oriented patterns. When acting as Player B, it sometimes returned part of the tripled investment (e.g., 50 percent in Round 1) to encourage trust, even though the subgame perfect prediction is always to return nothing. However, in later rounds it consistently converged to the payoff-maximizing choice of returning zero, especially when playing as Player A and anticipating Player B's behavior. This mixture suggests that the LLM balances economic rationality with social reasoning, unlike the strict backward-induction logic of SPNE. Compared to human participants, the LLM's decisions were more stable and transparent, but they also revealed sensitivity to framing: when fairness was emphasized in the prompt, cooperative actions appeared more likely.

3.3 3(c) Comparative analysis theory building

In theory, the SPNE of the one-shot Trust Game predicts that Player A invests 0 and Player B returns 0, which also coincides with the Nash equilibrium in the normal form. In our human session, participants sometimes invested positive amounts or returned part of the investment, despite this lowering their material payoffs (Fehr and Schmidt 1999; Camerer 2003).

During the LLM session, initial rounds showed partial returns, while later rounds converged to returning 0. The LLM behavior was more consistent than humans and included explicit reasoning, but it remained sensitive to prompt framing and payoff visibility (Bubeck et al. 2023).

These discrepancies may reflect broader utility considerations: humans may weigh fairness and reciprocity, whereas LLM outputs depend on training priors and alignment. A potential refinement, Behavioral SPNE, preserves the extensive-form logic of SPNE but replaces material payoffs with social-preference utilities and allows probabilistic choice:

$$U_i = \pi_i + \alpha \cdot F_i, \quad P(a_i) = \frac{\exp(\lambda U_i(a_i))}{\sum_{a_i'} \exp(\lambda U_i(a_i'))}.$$

Here, $\alpha \geq 0$ weights social preference, and λ captures bounded rationality. This formulation can account for cooperative tendencies and prompt-sensitive reasoning while remaining consistent with subgame equilibrium logic (Rabin 1993; Gauthier 2019).

References

Berg, Joyce, John Dickhaut, and Kevin McCabe. 1995. “Trust, Reciprocity, and Social History.” *Games and Economic Behavior* 10(1): 122–142.

Bommasani, Rishi, et al. 2021. *On the Opportunities and Risks of Foundation Models.* https://arxiv.org/abs/2108.07258.

Fehr, Ernst, and Klaus Schmidt. 1999. “A Theory of Fairness, Competition, and Cooperation.” *Quarterly Journal of Economics* 114(3): 817–868.

Google Colab. 2020. *Colaboratory: Python in the Browser.* https://colab.research.google.com/.

Knight, Vincent. 2021. *Nashpy: A Python library for the computation of equilibria of 2-player strategic games.* https://nashpy.readthedocs.io/.

OpenAI. 2023. *GPT-4 Technical Report.* https://openai.com/research/gpt-4.

Rubinstein, Ariel, and Martin Osborne. 1994. *A Course in Game Theory.* Cambridge, MA: MIT Press.

Savani, Rahul, and Bernhard von Stengel. 2015. “Game Theory Explorer – Software for the Applied Game Theorist.” *Computational Management Science* 12(1): 5–33. http://www.gametheoryexplorer.org/.

Shoham, Yoav, and Kevin Leyton-Brown. 2009. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge, UK: Cambridge University Press.

Acknowledgements

The author thanks OpenAI’s ChatGPT (GPT-4) for providing guidance and conceptual suggestions during the design and analysis of the Trust Game experiments. All interpretations and conclusions remain the responsibility of the author.