CSCE 631-D: Intelligent Agents — Computational Game Solving

Course Overview — What to Expect

Alan Kuhnle

Fall 2026

Texas A&M University

Learning Objectives

By the end of this overview, you should be able to:

1. Explain the course question: how should intelligent agents act when other agents also choose?
2. Locate the theory core and LLM-agent thread in the semester roadmap.
3. Identify the work required for programming assignments and the course project.
4. Describe the TAMU API credential model and complete the first-week setup without exposing personal credentials.

Today's Map

Pacing

This overview is designed for 30 minutes: the course question and roadmap (10 minutes), how the work is assessed (5), API setup (10), and first-week actions (5). Detailed API work continues in the Canvas guide and notebook.

The Question Behind the Course

Single-agent question
Which action best advances my objective?

Multi-agent question
Which action is sensible when my outcome also depends on what other agents do?

Game theory gives us a language for the second question:

- who chooses;
- what each participant can do and observe; and
- how each participant values the resulting outcome.

Module 1 gives this language its formal notation; today we use it to see where the course is headed.

Why Game Theory Matters for AI Agents

An agent that plans around a user, a tool, a model, or another agent is often inside a strategic interaction.

Classical settings

- Chess and other adversarial search
- Auctions and matching markets
- Network routing and resource allocation

Current agent settings

- LLM negotiation and debate
- Tool-using and collaborative agents
- Monitoring agents that may resist oversight

The central discipline is to specify the interaction before drawing conclusions from a model's behavior.

A 60-Second Prediction: A Coordination Game

Two LLM agents independently choose a protocol for a joint task. Each chooses ADOPT or WAIT; each entry is a **payoff**: a numerical value representing the two agents' preferences, listed as (Agent 1, Agent 2).

	Agent 2	
Agent 1	ADOPT	WAIT
ADOPT	$(\mathbf{2}, \mathbf{2})$	$(0, 0)$
WAIT	$(0, 0)$	$(\mathbf{1}, \mathbf{1})$

Take 30 seconds: write down one predicted joint action, then compare it with a neighbor or post it in the course discussion. What part of the prompt, history, or protocol could change that prediction?

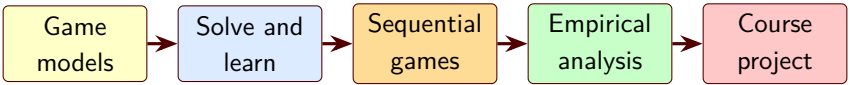
What Counts as a Stable Prediction?

A **pure-strategy Nash equilibrium** is a joint action from which no single player can improve its payoff by changing its action alone.

For the coordination game, both diagonal outcomes qualify: no player can improve their payoff by changing alone; here, every unilateral change in fact lowers it. The game therefore raises a coordination question: which equilibrium will the agents reach?

Return to the prediction: a shared prompt, history, or protocol can make one of the two highlighted equilibria easier for both agents to select.

The Course Arc



Every module pairs a **theory core** with an **agent thread**. Classical results provide the vocabulary and guarantees; modern agents provide the systems we will analyze and build.

Semester Roadmap: A Preview of the Vocabulary

M denotes module. These are previews; each term is defined when its module begins.

When	Module	Focus
Weeks 1–2	M1: Strategic games	Normal-form games; minimax (worst-case play); Nash equilibrium
Weeks 3–4	M2: Equilibrium algorithms	Auctions; complexity; correlated equilibrium (recommendations no player wants to disobey)
Weeks 5–7	M3–M5: Learning and self-play	Regret, game trees, CFR (counterfactual regret minimization)
Weeks 8–10	M6–M8: Agent systems	Abstractions, EGTA (empirical game-theoretic analysis), negotiation, monitoring
Weeks 11–15	M9, PA5, and project	Applications, evaluation, project reports

Take one minute: which row connects most directly to a multi-agent system you already know?

What You Will Build, Analyze, and Be Graded On

Programming assignments

PA1: payoff estimation; PA2: regret minimization; PA3: CFR self-play; PA4: debate plus a mechanism-design extension (rules that shape incentives); PA5: monitoring and AI control. Together: 60%. Three total 24-hour slip days apply to programming assignments; after those, late work loses 10% per day for up to three days.

Course project

Frame a research question about an intelligent-agent interaction, build a reproducible evaluation, and defend the conclusions with evidence.

Proposal: Week 6; progress report: Week 11; presentation: Week 14; final report: Week 15. The project is 40%.

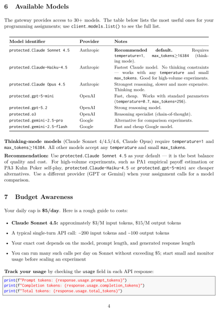
There are no exams.

To conduct this work from Week 1, each student needs a safe, reproducible way to call and evaluate a model.

TAMU AI API: Your Setup Reference

- `https://chat.tamu.ai/api` gives us controlled, reproducible access to real models from Week 1.
- The Canvas guide has the live model list, streaming example, troubleshooting, and $5 daily budget. Start with a small test, record its cost, then scale up.

The instructor will demonstrate one streamed request; the Canvas notebook is your hands-on reference.



6 Available Models

The gateway provides access to 30+ models. The table below lists the most useful ones for your programming assignments; use `client.models.list()` to see the full list.

Model identifier	Provider	Notes
`protected.Claude Sonnet 4.5`	Anthropic	**Recommended default.** Requires `temperature=1`, `max_tokens≥16384` (thinking mode).
`protected.Claude-Haiku-4.5`	Anthropic	Fastest Claude model. No thinking constraints — works with any `temperature` and small `max_tokens`. Good for high-volume experiments.
`protected.Claude Opus 4.5`	Anthropic	Strongest reasoning, slower and more expensive. Thinking mode.
`protected.gpt-5-mini`	OpenAI	Fast, cheap. Works with standard parameters (`temperature=0.7, max_tokens=256`).
`protected.gpt-5.2`	OpenAI	Strong reasoning model.
`protected.o3`	OpenAI	Reasoning specialist (chain-of-thought).
`protected.gemini-2.5-pro`	Google	Alternative for comparison experiments.
`protected.gemini-2.5-flash`	Google	Fast and cheap Google model.

Thinking-mode models (Claude Sonnet 4/4.5/4.6, Claude Opus) require `temperature=1` and `max_tokens≥16384`. All other models accept any `temperature` and small `max_tokens`.

Recommendation: Use `protected.Claude Sonnet 4.5` as your default — it is the best balance of quality and cost. For high-volume experiments, such as PA1 empirical payoff estimation or PA3 Kuhn Poker self-play, `protected.Claude-Haiku-4.5` or `protected.gpt-5-mini` are cheaper alternatives. Use a different provider (GPT or Gemini) when your assignment calls for a model comparison.

7 Budget Awareness

Your daily cap is **$5/day**. Here is a rough guide to costs:

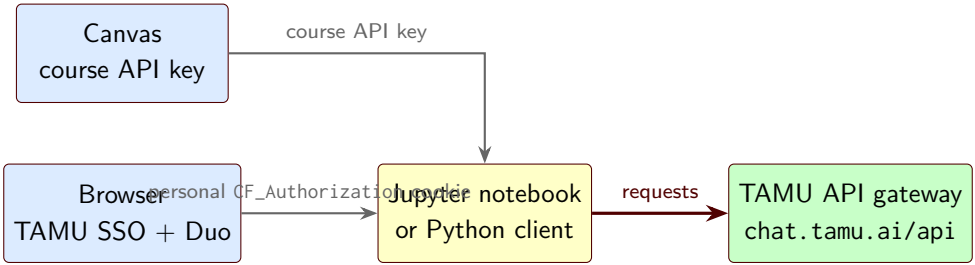
- **Claude Sonnet 4.5:** approximately $3/M input tokens, $15/M output tokens
- A typical single-turn API call: ~200 input tokens and ~100 output tokens
- Your exact cost depends on the model, prompt length, and generated response length
- You can run many such calls per day on Sonnet without exceeding $5; start small and monitor usage before scaling an experiment

Track your usage by checking the `usage` field in each API response:

```
print(f"Prompt tokens: {response.usage.prompt_tokens}")
print(f"Completion tokens: {response.usage.completion_tokens}")
print(f"Total tokens: {response.usage.total_tokens}")
```

4

API Setup: Two Credentials, One Test



The API key identifies the course application; the cookie authenticates your NetID and budget. The checklist carries the procedural detail.

First-Week API Setup Checklist

1. In Canvas, retrieve the course API key and open `TAMU-API-Guide.pdf` plus the supplied notebook.
2. Log into `https://chat.tamu.ai` with TAMU SSO and Duo; copy your personal `CF_Authorization` cookie using browser developer tools.
3. Install the OpenAI Python client and run the Canvas notebook; verify one streamed test request and save its output.
4. Keep both credentials out of public repositories. A 401 or 403 usually means your cookie expired: sign in again and copy a fresh value.

For Claude thinking models

Use `stream=True`, `temperature=1`, and `max_tokens >= 16384`. The setup guide explains why these parameters are required by the gateway for extended-thinking models.

What “Graduate-Level” Means Here

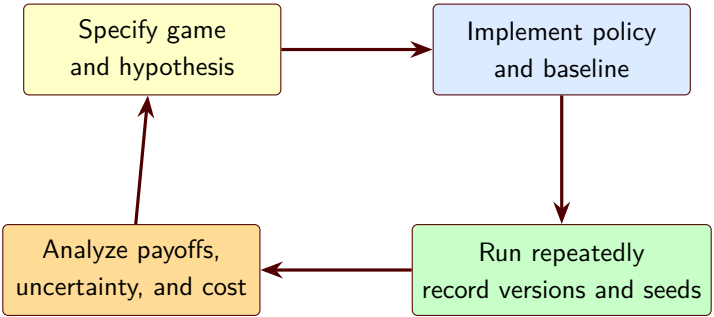
- **Precision:** state the game, assumptions, data, and metric.
- **Reproducibility:** leave a repository and instructions another student can rerun.
- **Explanation:** connect behavior to an algorithm or game-theoretic concept.
- **Judgment:** name limits from variance, costs, and threats to validity.

A useful question for every submission

What claim does this result support, and what evidence would weaken that claim?

For example, in PA1 precision means stating the agent interaction and payoff measurement before reporting the estimated results.

A Reproducible Experiment Has a Trace



This loop is the recurring structure of the programming assignments and the course project.

Next: the course resources and support available when you need help.

How We Will Work Together

- **Class and recording:** Zach 592, TR 9:35–10:50; recordings serve the asynchronous section.
- **Questions:** Canvas announcements are official; use Canvas messaging or email, and bring concrete artifacts to office hours (Monday 9:00–9:50, Thursday 3:00–3:50, PETR 421; async/remote access: `https://tamu.zoom.us/my/kuhnle`).
- **Resources:** Canvas holds the syllabus, accessible notes, API guide, notebooks, and discussion space.

Working standard

Submit work you can explain, reproduce, and defend. Follow each assignment's collaboration rules, acknowledge permitted assistance, and never share credentials, private data, or another student's work. If a course artifact creates an access barrier, let the instructor know early.

Your First Week

1. Read the syllabus and scan the M1 reading list.
2. Set up Python, Jupyter, Git, and the TAMU API client.
3. Run the supplied Canvas API notebook and keep its output for reference.
4. Attend or watch Lecture 1: games, agents, and strategic interaction.
5. Write down one multi-agent system whose behavior you want to understand better.

Three Ideas to Carry Forward

1. An intelligent agent's success can depend on other agents' decisions.
2. A convincing evaluation specifies the interaction and preserves enough evidence to reproduce it.
3. This course joins game-theoretic foundations with experiments on modern LLM agents.

Questions before we begin Module 1?