CSCE 631 — Summer 2026 Course Project

Dr. Alan Kuhnle
Texas A&M University

Summer I 2026

Weight: 50% of final grade (Proposal 5% + Final Deliverable 45%)
Format: Individual only

1 Overview

The course project is your opportunity to apply the concepts from CSCE 631 to a focused research or implementation problem. You will select a topic from the curated list below (or propose your own with instructor approval), write a brief proposal, and deliver a code artifact plus a written report by the end of the course.

Projects are **individual** — no group submissions. With 20 students and 10 curated topics, multiple students may work on the same topic. This is expected and welcome; you will work independently, and different approaches to the same problem make grading more informative.

2 Timeline

Date	Milestone
May 26 (Week 1)	Project topics released (this document)
June 6 (Week 2)	**Proposal due** — 1-page PDF submitted via Canvas
June 29 (Week 5)	**Final submission due** — code/notebook + written report via Canvas

3 Topic Selection

1. Review the 10 curated topics below.
2. **Rank your top 3 choices** and submit your ranking along with your proposal by June 6.
3. The instructor will assign topics with a cap of 2–3 students per topic to ensure variety.
4. **Custom topics** are welcome but require instructor approval. Include a clear problem statement, planned approach, and connection to course material in your proposal. The instructor will confirm feasibility by the end of Week 2.

4 Curated Topics

Topic 1: Convergence of Regret-Matching Variants

Connects to: Weeks 2–3 (Lectures 7–9)

Implement and experimentally compare regret matching (RM), RM+, and predictive RM on a suite of normal-form games. Visualize convergence rates and exploitability as a function of iterations. Investigate how game structure (e.g., number of actions, zero-sum vs. general-sum) affects convergence speed.

Expected deliverables: Implementation of all three variants, experiments on at least 5 games of varying size, convergence plots, and a discussion of when each variant is preferable.

Starting points: Lectures 7–9 on regret minimization; Hart & Mas-Colell (2000) for RM; Tammelin (2014) for RM+.

Topic 2: CFR on a Game Beyond Kuhn Poker

Connects to: Weeks 2–3 (Lectures 13–14)

Implement tabular Counterfactual Regret Minimization for Leduc Hold'em, Liar's Dice, or another small extensive-form game. Measure exploitability convergence over iterations and compare to the theoretical $O(1/\sqrt{T})$ bound.

Expected deliverables: Game tree implementation, CFR solver, exploitability computation, convergence plots with theoretical bound overlay, analysis of information-set complexity.

Starting points: Lecture 13 (CFR); Zinkevich et al. (2007); Lanctot et al. (2009) for Leduc Hold'em rules.

Topic 3: Game Abstraction and Solution Quality

Connects to: Week 4 (Lectures 15–16)

Implement a simple card or action abstraction for a small poker variant (e.g., Leduc Hold'em). Solve the abstracted game with CFR, then measure how much exploitability increases relative to the unabstracted solution as the abstraction gets coarser.

Expected deliverables: Abstraction implementation (at least 3 coarseness levels), CFR solutions for each, exploitability measurements, plots showing the accuracy-compression tradeoff.

Starting points: Lectures 15–16 on game abstraction; Gilpin & Sandholm (2007) for abstraction techniques.

Topic 4: Deep CFR Replication

Connects to: Weeks 4–5 (Lectures 15–18)

Reproduce a simplified version of Deep CFR on Kuhn or Leduc poker using a small neural network for advantage estimation. Compare the learned strategy's exploitability to tabular CFR at matched iteration counts. *Note: This is an ambitious project.* Implementing a neural advantage network plus CFR training requires comfort with both deep learning frameworks and game-solving algorithms.

Expected deliverables: Neural network implementation for advantage estimation, training loop, exploitability comparison with tabular CFR, analysis of where the neural approximation helps or hurts.

Starting points: Lecture 18 (Deep CFR); Brown et al. (2019) "Deep Counterfactual Regret Minimization."

Topic 5: LLM Debate Protocol Design

Connects to: Week 5 (Lecture 5.2); extends PA2

Vary the structure of multi-agent LLM debate: simultaneous vs. sequential arguments, number of rounds, with or without a judge agent, asymmetric model capabilities. Analyze how protocol design affects argument quality, persuasiveness, and convergence of positions.

Expected deliverables: Implementation of at least 3 debate protocol variants using the TAMU API, systematic evaluation across multiple debate topics, analysis of how game-theoretic structure influences outcomes.

Starting points: Lecture 5.2; PA2 codebase and TAMU API Guide; Du et al. (2023) "Improving Factuality and Reasoning in Language Models through Multiagent Debate."

Topic 6: Red-Teaming LLMs with Game-Theoretic Search

Connects to: Week 5 (Lectures 5.1, 5.3)

Design adversarial prompts using minimax or best-response reasoning. Formalize the attacker-defender interaction as a two-player game, define the action and outcome spaces, and analyze the strategic structure. Evaluate whether systematic game-theoretic search finds vulnerabilities that ad hoc red-teaming misses. *Note: This is an ambitious project.* Formalizing the attacker-defender game and building a search-based attack strategy requires careful modeling up front.

Expected deliverables: Formal game model of the red-teaming interaction, implementation of a search-based attack strategy, comparison with baseline (random/manual) red-teaming, analysis of attack success rates and defensive implications.

Starting points: Lectures 5.1 and 5.3; Perez et al. (2022) "Red Teaming Language Models with Language Models."

Topic 7: Tree-Search Reasoning for LLM Agents

Connects to: Week 5 (Lecture 5.1)

Implement a simplified MCTS or beam-search wrapper around an LLM, inspired by RAP (Hao et al., 2023) or Tree of Thoughts (Yao et al., 2023). Compare against single-pass generation on a reasoning benchmark (e.g., Game of 24, multi-step math, or planning tasks).

Expected deliverables: Search wrapper implementation, evaluation on at least one benchmark, comparison of accuracy vs. compute budget, analysis of when search helps most.

Starting points: Lecture 5.1; Hao et al. (2023) "Reasoning with Language Model is Planning with World Model"; Yao et al. (2023) "Tree of Thoughts."

Topic 8: Multi-Agent Negotiation as an Extensive-Form Game

Connects to: Weeks 3 + 5 (Lectures 10–12, 5.2)

Build two LLM agents that negotiate over a resource allocation problem (e.g., splitting items with private valuations). Map the interaction to an extensive-form game, define information sets and payoffs, and analyze whether the agents converge to anything resembling a known equilibrium or bargaining solution. *Note: This is an ambitious project.* The full deliverable list spans environment design, agent implementation, and formal game modeling — start with the EFG formalization.

Expected deliverables: Negotiation environment, LLM agent implementation, formal game-theoretic model of the interaction, experimental results across multiple negotiation scenarios.

Starting points: Lectures 10–12 on extensive-form games; Lecture 5.2; Nash (1950) "The Bargaining Problem."

Topic 9: Equilibrium Computation at Scale

Connects to: Weeks 1–2 (Lectures 5–6)

Implement support enumeration or the Lemke-Howson algorithm for two-player normal-form games. Benchmark runtime scaling as game size increases (number of actions from 3 to 50+). (Note: Lemke-Howson and support enumeration are practical to approximately 20–30 actions; the 50+ range is for LP-based methods.) Compare to LP-based Nash equilibrium computation.

Expected deliverables: Implementation of at least two equilibrium computation methods, runtime benchmarks on games of increasing size, scaling plots, analysis of practical limits and which method dominates in which regime.

Starting points: Lectures 5–6 on computing equilibria; Nisan et al. (2007) *Algorithmic Game Theory* Ch. 3.

Topic 10: Do LLMs Play Nash? Strategic Behavior of Language Models

Connects to: Weeks 1 + 5 (Lectures 1–4, 5.1)

Present an LLM with game-theoretic scenarios: prisoner's dilemma variants, coordination games, battle of the sexes, bargaining problems. Evaluate whether its responses approximate known equilibrium strategies. Systematically analyze where and why it deviates.

Expected deliverables: Suite of at least 10 game-theoretic scenarios with known equilibria, LLM response collection across multiple models or temperature settings, quantitative comparison to equilibrium play, analysis of systematic biases (e.g., cooperation bias, focal point effects).

Starting points: Lectures 1–4 on normal-form games and solution concepts; Akata et al. (2023) "Playing Repeated Games with Large Language Models."

5 Deliverables

5.1 Proposal (Due June 6 — 5% of final grade)

A **1-page PDF** containing:

- **Chosen topic** (or custom topic description)
- **Topic ranking** (your top 3 from the curated list)
- **Problem statement** (1–2 sentences: what question are you answering?)
- **Planned approach** (how will you tackle it? what tools/methods?)
- **Connection to course concepts** (which lectures and ideas does this build on?)
- **Expected deliverables** (what will you submit at the end?)

The proposal is intentionally lightweight. Its purpose is to confirm you have a feasible plan and to let the instructor flag scope issues early.

5.2 Final Submission (Due June 29 — 45% of final grade)

Two components:

Written report (4 pages max, excluding references)

- **Introduction:** problem statement and motivation
- **Method:** approach, algorithms, implementation choices
- **Results:** experiments, measurements, visualizations
- **Discussion:** interpretation, connection to course theory, limitations
- **References**

Use any reasonable format (LaTeX, Word, Markdown-to-PDF). Figures and tables count toward the page limit. Aim for clarity over length — a tight 3-page report is better than a padded 4-page one.

Code/notebook

- Well-commented and reproducible
- Include a README or top-cell instructions explaining how to run the code
- If using the TAMU API, include instructions for cookie setup (see TAMU-API-Guide.md)
- Jupyter notebooks are encouraged but not required

6 Grading Rubric

Component	Weight	Criteria
Proposal	5%	Clarity of problem statement; feasibility of plan; connection to course material
Technical execution	20%	Correctness of implementation; depth of experiments; code quality and reproducibility
Analysis and insight	15%	Quality of interpretation; connection to game-theoretic concepts; honest discussion of limitations
Writing quality	10%	Clarity of exposition; logical organization; effective use of figures and tables
Total	**50%**	

7 Academic Integrity

This is an individual assignment. You may discuss ideas with classmates at a high level, but all code and writing must be your own. If you use LLM assistance (ChatGPT, Claude, Copilot, etc.), you must:

1. Disclose the tool and how you used it (e.g., "Used Claude to debug my CFR implementation").
2. Be prepared to explain every part of your submitted code and analysis.

Using LLMs as a tool is fine; submitting work you don't understand is not. Standard university academic integrity policies apply.

8 Tips

- **Start early.** Five weeks is short. The proposal forces you to commit to a plan by Week 2 (June 6) — use Week 1 to read relevant papers and think about scope.
- **Scope down, not up.** A clean, well-analyzed result on a simple version of your problem is better than a half-finished attempt at something ambitious. You can always note extensions in your report.
- **Connect to theory.** The strongest projects use the course's formal vocabulary — equilibrium concepts, regret bounds, information sets, game trees — to frame and interpret their results. Don't just run experiments; explain what the results mean through a game-theoretic lens.
- **Talk to the instructor.** If you're unsure about scope, feasibility, or direction, ask. The proposal is the formal checkpoint, but informal questions are welcome anytime.
- **Figures matter.** A well-labeled convergence plot or game tree diagram communicates more than a paragraph of description.