CSCI 4550/6550 Final Project – Fall 2025

Due: 12/2/2025 (as a Zip file in eLC – see Deliverables)

Required for **Graduate** Students. Optional (Bonus) for Undergraduate Students

Objective

The final project should demonstrate a meaningful application of the ideas and methods discussed in this course. Each project must include working code and clearly show how it connects to key AI principles such as search, reasoning, optimization, or learning.

Scope of Topics

Projects may explore one or more of the following areas:

- **Uninformed search:** Breadth-First Search (BFS), Depth-First Search (DFS), Uniform-Cost Search (UCS), Depth-Limited or Iterative Deepening Search
- **Informed search:** Greedy Best-First Search, A* Search, heuristic design, admissibility and consistency analysis
- **Optimization and local search:** Hill Climbing, Simulated Annealing, Local Beam Search, Genetic Algorithms
- **Adversarial search:** Minimax and Alpha-Beta Pruning for competitive or game-based settings
- **Logical reasoning:** Knowledge-based agents, propositional or first-order logic inference, reasoning in partially observable domains
- **Behavior-based reasoning and planning:** Behavior Tree based agents executing reactive plans in simulated environments
- **Neural networks and machine learning:**
    - Designing, training, or analyzing neural networks on chosen tasks
    - Classical ML algorithms such as SVMs, decision trees, or clustering
    - Reinforcement learning agents in simulated environments (e.g., Canonical Games, gym environments, etc.)
    - Transformers or other modern architectures for structured data, text, or control
    - Integrating search or logic with learning-based approaches

Deliverables:

All components are due together.

1. **Code and Documentation:** Complete, functional implementation of the project **(in Python)** with a brief **README** file explaining the setup and usage.
2. **Project Report (2+ pages):** A clear description of the problem, methods, experiments, and conclusions.
3. **Invited Oral Presentation (if deemed required by the instructor and/or TA):** An 8–10 minute oral presentation summarizing the motivation, approach, and results.

Example Project Ideas

Category	Example Topics
Search and Pathfinding	Implement and compare BFS, UCS, and A* for different graph structures or mazes.
Heuristic Design	Develop and evaluate heuristics for the 8-puzzle, route planning, or navigation problems.
Optimization	Apply Simulated Annealing or Genetic Algorithms to scheduling, placement, or layout problems.
Adversarial Search	Create an AI agent for a two-player game (e.g., Tic-Tac-Toe, Connect-4) using Minimax with Alpha-Beta pruning.
Logical Reasoning	Build an agent that infers safe actions in a Wumpus-style environment using propositional logic.
Neural Networks	Explore how network architecture or activation choices affect performance on a dataset of your choice.
Machine Learning	Compare classical models (SVM, decision tree, k-NN) on a small real or synthetic dataset.
Reinforcement Learning	Train an agent to learn a simple control task or grid-world navigation policy.
Transformers	Fine-tune a small transformer model for classification, translation, or summarization tasks.
Hybrid Systems	Combine heuristic search with a learned model or use ML methods to guide symbolic reasoning.

Grading Rubric (100 points total)

Criterion	Description
Problem Definition (15%)	Clear explanation of the problem and its relevance to course topics.
Technical Implementation (30%)	Correct, functional, and well-structured code.
Creativity and Scope (15%)	Ambitious or original application of AI techniques.
Evaluation and Results (20%)	Appropriate testing, analysis, and interpretation of outcomes.
Report Clarity (20%)	Well-written, organized explanation of goals, methods, and findings.
Presentation (10%) - Optional	Effective oral communication of the project's motivation, approach, and conclusions.

Recommended Report Structure (Optional, but strongly recommended)

- **Background & Motivation** – Define the problem and its relevance to AI search or learning.
- **Related Concepts** – Briefly explain which algorithms from class you build on (cite chapters or slides).
- **Proposed Solution / Design** – Describe the state representation, heuristic, or model architecture.
- **Implementation Details** – Include language, libraries, environment, and data (if any).
- **Evaluation & Results** – Present performance metrics, examples, or analyses (e.g., path cost, accuracy, training curve).
- **Discussion & Analysis** – Reflect on what worked, what didn't, and discuss trade-offs or observations.
- **Conclusion & Future Work** – Summarize findings and outline possible improvements or extensions.