Artificial Intelligence Course Overview: Exploring the Foundations of Intelligent Systems

Welcome to the Artificial Intelligence course experience! This presentation will provide an overview of the concepts, algorithms, and practical applications covered during the Jan–June 2025 session, led by instructor **Razorshi Prozzal Talukder**. Presented by **Oli Ahmed** (ID: **0562310005101019**).

Purpose of This Course: AI & Agents

Understand AI's Thinking

Grasp how AI systems perceive, reason, and make decisions, mimicking human cognitive processes.

Develop AI-Powered Games

Apply AI algorithms to create intelligent game opponents that offer challenging and dynamic gameplay.

Utilize AI Tools

Become proficient in using cutting-edge AI tools for content creation, automation, and problem-solving.

Strengthen Problem-Solving

Enhance critical thinking and analytical skills to tackle complex real-world problems with AI solutions.

AI agents are central to building intelligent systems. We explored various types:

- **Reactive Agents:** Respond immediately to current perceptions without internal state or memory.
- **Model-Based Agents:** Maintain an internal model of the world to track states and predict outcomes.
- **Goal/Utility-Based Agents:** Choose actions that achieve specific goals or maximize expected utility, often involving planning and prediction.

This equipped us to build systems that think and act intelligently across diverse applications.

CSPs, Knowledge Representation & Logic

Constraint Satisfaction Problems (CSPs)

We delved into CSPs, a fundamental AI paradigm for solving problems by finding a state that satisfies a given set of constraints. Key applications include:

- **Graph Coloring**: Assigning colors to regions such that no adjacent regions share the same color.
- **Scheduling:** Allocating resources over time while respecting various dependencies and limitations.
- **Sudoku:** Filling a 9x9 grid with digits such that each column, row, and 3x3 subgrid contains all digits from 1 to 9.

Techniques like Branch and Bound and K-Consistency were vital for efficiently navigating the solution space.

Knowledge Representation & Logic

Understanding how AI systems represent knowledge is crucial. We explored:

- **If-Then Logic:** Expressing rules and relationships in a clear, conditional format.
- **CNF & DNF Forms:** Converting logical statements into Conjunctive Normal Form and Disjunctive Normal Form for systematic processing.

Inference Techniques

To reason with represented knowledge, we studied:

- **Modus Ponens:** A basic rule of inference that allows us to derive conclusions from premises.
- **Resolution:** A powerful inference rule for automated theorem proving.
- **Forward Chaining:** A data-driven approach that moves from facts to conclusions.
- **Backward Chaining:** A goal-driven approach that works backward from the desired conclusion.

These methods taught us to encode rules and solve problems systematically, forming the bedrock of intelligent reasoning.

Overview of Key AI Algorithms

Uninformed Search Algorithms

Methods like Breadth-First Search (BFS) and Depth-First Search (DFS) explore states without prior knowledge of the goal's location, ensuring completeness but often less efficiency.

Informed Search Algorithms

A* Search and Best-First Search leverage heuristic functions to guide their exploration, significantly improving efficiency for complex problems by prioritizing promising paths.

Local Search Techniques

Algorithms such as Hill Climbing and Beam Search optimize solutions by iteratively improving a single state or a small set of states, suitable for continuous optimization problems.

Adversarial Search

Minimax and Alpha-Beta Pruning are crucial for multi-agent environments where agents have conflicting goals, commonly seen in games like chess or checkers.

Advanced Techniques

Other sophisticated algorithms, including AO* Search and Bidirectional Search, offer specialized solutions for various AI challenges, enhancing problem-solving capabilities.

Favorite Algorithms We Explored

These algorithms are widely applied in pathfinding, robot navigation, and solving complex puzzles, demonstrating their versatility and importance in AI.

Minimax

Simulates an opponent's optimal moves in game theory, crucial for strategic decision-making in turn-based games, ensuring the AI makes the best possible play.

Alpha-Beta Pruning

An optimization technique for Minimax that drastically reduces the number of nodes evaluated in the search tree, speeding up decisions without compromising optimality.

A* Search

A highly efficient pathfinding algorithm that finds the shortest path between two points in a graph, widely used in navigation systems and complex puzzle-solving.

Hill Climbing

A local search algorithm that iteratively moves towards a better solution by selecting the steepest ascent, ideal for optimization problems where quick, potentially suboptimal, moves are acceptable.

AI in Games – Strategic Decision Making

Game AI leverages advanced algorithms to create intelligent and engaging opponents.

- **Minimax & Alpha-Beta:** These core algorithms enable AI to make optimal or near-optimal moves, simulating human-like strategic thinking in complex game scenarios.

Applications Built:

Our hands-on experience included developing AI for classic games:

- Tic Tac Toe
- Chess (Mini Version)
- Connect Four

Why is Game AI Important?

- **Mimics Human Opponent Strategies:** Provides realistic and challenging gameplay experiences.
- **Showcases AI's Abilities:** Demonstrates AI's capacity for planning, competition, and dynamic adaptation in response to player actions.

These projects provided a practical understanding of how theoretical AI concepts translate into real-world applications, enhancing problem-solving skills in a fun and interactive way.

Optimization-Based AI Techniques

These optimization techniques are crucial for solving complex problems where finding the absolute global optimum is computationally expensive or impossible. They prioritize finding good, practical solutions efficiently.

Beam Search

A memory-efficient variant of best-first search, Beam Search keeps only the top 'k' most promising paths at each step, significantly reducing computational load while still exploring a wide range of solutions.

Hill Climbing

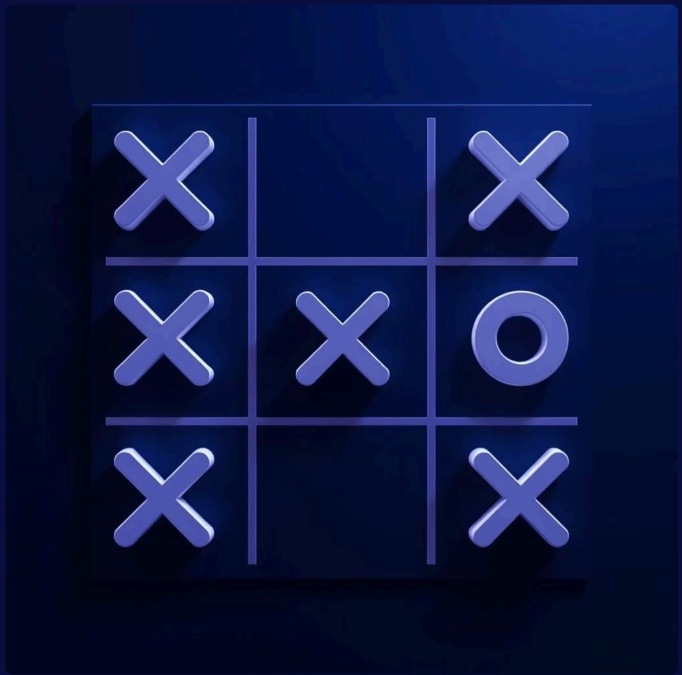
A local search algorithm that iteratively moves towards a better solution by always choosing the neighbor state that offers the greatest improvement, aiming for incremental progress towards an optimal state.

Applications:

- Speech Recognition Systems: Optimizing models for accurate transcription.
- Automated Planning and Scheduling: Efficiently allocating resources and tasks.
- Optimizing Large Game States: Finding optimal moves in vast game trees.
- Solving Combinatorial Optimization Problems: Tackling challenges like the Traveling Salesperson Problem.

AI Games Developed

Tic Tac Toe

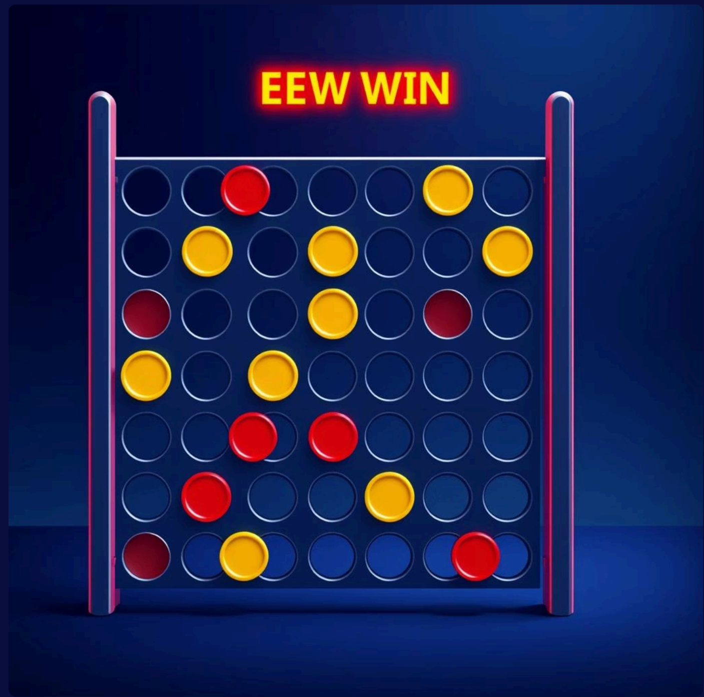


Developed with a Python GUI, this game features an unbeatable AI powered by the Minimax algorithm, ensuring strategic depth and a challenging experience for players.

Chess (Mini Version)

A simplified Chess game implemented in Python, featuring Alpha-Beta pruning for an intelligent and challenging AI opponent, demonstrating efficient search techniques.

Connect Four



Built with a heuristic-enhanced Minimax strategy, this game includes robust win detection across horizontal, vertical, and diagonal lines, providing complex gameplay.

Features Across All Games:

- Human vs. Computer gameplay mode
- Authentic AI decision-making processes
- Intuitive Graphical User Interfaces (GUI)
- Well-documented code with clear comments.

AI Tools & Key Learning Outcomes

Tools Used

- **Python:** The primary language for AI implementation and game development.
- **Tkinter:** Utilized for creating intuitive Graphical User Interfaces (GUIs) for our projects.
- **GitHub:** Essential for collaborative development, version control, and project management.
- **Canva & Gamma:** Employed for designing visually appealing presentations and course materials.
- **Pictory & Invideo AI:** Used for generating and editing AI-powered video content.

These tools collectively enhanced our ability to develop, manage, and present AI projects effectively.

What I Learned

- **Real-World AI Implementation:** Gained hands-on experience in applying theoretical AI concepts to practical problems, building functional intelligent systems.
- **Professional Documentation & Showcase:** Developed skills in documenting projects thoroughly and showcasing them professionally, a crucial aspect of software development.
- **Confidence in Complex Challenges:** Built significant confidence in approaching and solving intricate logical and computational challenges, fostering a problem-solving mindset.

Conclusion & Acknowledgement

This Artificial Intelligence course has been a truly transformative journey. From exploring core concepts like search strategies and heuristic planning, to developing interactive games with real AI logic, I have acquired a robust set of skills that seamlessly bridge theoretical knowledge with practical application.

I am grateful for the invaluable support and profound insights provided by our esteemed supervisor, **Razorshi Prozzal Talukder**. His guidance has been instrumental in navigating the complexities of AI, fostering a deep understanding and passion for the subject.

This course has not only deepened my understanding of Artificial Intelligence but also empowered me with the courage and capability to create, present, and boldly dream within the ever-evolving landscape of AI. The knowledge gained here will undoubtedly serve as a strong foundation for future endeavors in this exciting field.