

AI Unplugged: Embodied Interactions for AI Literacy in Higher Education

Jennifer M. Reddig^{*1}, Scott Moon^{*2}, Kaitlyn Crutcher¹, Christopher J. MacLellan¹

¹Georgia Institute of Technology

²Independent Researcher

jreddig3@gatech.edu, scottqmoon@gmail.com, kcrutcher3@gatech.edu, cmacell@gatech.edu

Abstract

As artificial intelligence (AI) becomes increasingly integrated into daily life, higher education must move beyond code-centric instruction to foster holistic AI literacy. We present a novel pedagogical approach that integrates embodied, unplugged activities into a university-level Introduction to AI course. Inspired by the effectiveness of CS Unplugged in K-12 education, our physical, collaborative activities gave students a first-person perspective on AI decision-making. Through interactive games modeling Search Algorithms, Markov Decision Processes, Q-learning, and Hidden Markov Models, students built an intuition for complex AI concepts and more easily transitioned to mathematical formalizations and code implementations. We present four unplugged AI activities, describe how to bridge from unplugged activities to plugged coding tasks, reflect on implementation challenges, and propose refinements. We suggest that unplugged activities can effectively bridge conceptual reasoning and technical skill-building in university-level AI education.

Materials — <https://github.com/jmreddig/Intro-to-AI-Teaching-Resources>

Introduction

Artificial Intelligence (AI) has become increasingly embedded into everyday life. From social media, to music recommendation, to generative text and images, AI is playing a larger role in how people communicate and make decisions. As AI becomes more ubiquitous, people need to understand how it works, what it can and cannot do, and how to evaluate when a situation calls for an AI solution. This is true especially for those that design, build, and deploy these systems.

However, most student perceptions of AI are shaped not by the technical realities of AI, but by pop culture representations. Hingle and Johri (2023) found that student perceptions of AI are formed through everyday interactions and media portrayals, leading to conceptualizing of AI as a magical, superhuman technology. Students rarely enter the classroom with a clear sense of what AI is or how it fits into the broader socio-technical landscape (Marx, Leonhardt, and Bergner 2022). Students may have a hard time reconciling

their understanding of AI through media and the routine AI algorithms they learn to code in class. As such, many students are learning to build systems in computer science classes they do not fully understand.

Many introductory AI courses emphasize algorithmic techniques, mathematical formulas, and code-heavy assignments. Instruction is often delivered through lectures, and critical analysis of AI is reserved for the end of the semester, if time allows. K-12 AI education frameworks, such as the framework from the United Nations Educational, Scientific, and Cultural Organization (UNESCO; Miao and Shiohira 2022) or the framework proposed by Ng et al. (2021), emphasize the need to go beyond simply building AI systems. University education should pick up where K-12 education leaves off and also teach fundamental AI concepts, discuss whether and when an AI solution is appropriate, and explore the ethical implications of AI use.

To address potential gaps in university-level AI education, we developed a curriculum inspired in part by the CS Unplugged concept used in K-12 settings. In particular, our unplugged activities emphasize embodied cognition, the theory that human cognition is shaped through physical, interactive experiences (Varela, Thompson, and Rosch 2017). Abramson and Lindgren (2014) say that “conceptual reasoning originates in physical interaction and becomes internalized as simulated actions.” Our curriculum focuses on using embodied cognition to give students a first-person experience of AI decision-making to support their conceptual understanding of these complex algorithms. We provide several opportunities for students to simulate reasoning as AI agents.

We offer a potential model for teaching AI in ways that are both rigorous and reflective, preparing students to build intelligent systems with a deep understanding of their complex algorithmic underpinnings. In this paper, we:

- Present four unplugged activities for teaching AI,
- Describe how to bridge from unplugged activities to plugged coding tasks,
- Probe the viability of using an unplugged curriculum for AI in higher education, and
- Identify opportunities and challenges for unplugged instruction at the university level.

We describe a sample implementation of a joint unplugged/plugged Introduction to Artificial Intelligence course taught

^{*}These authors contributed equally.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

during the summer 2025 semester. We reflect on student interactions and provide suggestions for further iterations.

Background

AI literacy, as defined by Long and Magerko (2020), is a set of competencies that enable people to communicate, work, and live with AI technologies. These competencies involve understanding different definitions of intelligence, how computers make decisions, and how training data informs the results from AI algorithms. They also provide some design considerations for supporting learner-centered AI education. While all of the design considerations are important for building a holistic understanding of AI, we focused on three design considerations when building this course. *Embodied Interactions* give students a first-person perspective of how AI agents reason and make decisions. The *Critical Thinking* design consideration encourages students to become critical consumers of AI. The *Low Barrier to Entry* design consideration aims to ease learners' insecurities around math and CS abilities by providing supports so learners can build the necessary prerequisite skills.

Recent research in K-12 best practices for AI education continues to emphasize not only the technical, but also the ethical and conceptual dimensions of AI literacy. Lao (2020) developed the Machine Learning Education Framework with three main outcomes of the curriculum: Knowledge (general knowledge of AI methods), Skills (problem scoping, creating artifacts, analysis of results), and Attitudes (identity, community, self-efficacy). Ng et al. (2021) proposed four instructional goals for AI education: learners should know and understand AI, use and apply AI methods, evaluate and create AI systems, and engage with ethical issues. Miao and Shiohira (2022) (in association with UNESCO) examined these and many more K-12 AI curricula and identified three key domains: AI Foundations like algorithmic thinking and data literacy, Using and Developing AI systems, and Ethics and Social Impact of AI applications within and outside the Information and Communication Technology domain. These frameworks stress the importance of not only understanding AI algorithms and building systems, but analyzing the results and reflecting on ethical and social issues.

Higher education, however, is increasingly focused on the technological outcomes of AI courses. Game-based and project-based learning are the main approaches for building AI understanding and applying AI methods (Ng et al. 2023). Studies like those conducted by Markov et al. (2005), Vargas et al. (2020), and De Barros, Paiva, and Hayashi (2023) show that projects improve engagement. These gamified projects promote hands-on application and programming fluency, but students may lose sight of broader concepts as they grapple with implementation details. The current state of AI higher education focuses on the Using and Developing AI Systems and General Knowledge components of AI curriculum frameworks, neglecting ethical reasoning, analyzing and evaluating AI systems, and foundational knowledge like data literacy and bias. As AI becomes more embedded into everyday life, higher education must evolve to include holistic AI education like the K-12 frameworks.

CS Unplugged is a powerful pedagogical tool for making abstract computing ideas concrete and engaging in K-12 computer science classes. These technology-free activities use physical, often collaborative tasks that give students a first-person, embodied experience to support CS understanding. CS Unplugged activities lower the barrier to entry, increase motivation, and build conceptual understanding of CS concepts (Battal, Afacan Adanir, and Gülbahar 2021; Huang and Looi 2021). Recent research focuses on extending CS Unplugged activities to include AI concepts. AI Unplugged, coined by Lindner, Seegerer, and Romeike (2019), incorporated AI Unplugged activities into a curriculum to build conceptual understanding. Alraddady, Luong, and Young (2014) used embodied analogy to teach parallel processing algorithms. Long, Moon, and Magerko (2021) proposed an activity to help students understand how agents make connections using a semantic network. Lim et al. (2024) designed a board game to teach middle school students about facial recognition technology. Ma et al. (2023) demonstrates that unplugged activities can be more than a stand-alone activity and are most effective when part of a lesson that transitions to plugged activities. We use CS Unplugged inspired activities to build conceptual understanding of AI techniques.

Unplugged activities pull from the theory of embodied cognition (Varela, Thompson, and Rosch 2017), giving students interactive experiences to support a growing conceptual understanding. Hu, Huang, and Li (2024) used unplugged activities to build computational thinking skills in young children, demonstrating that the embodied experience facilitated learning. Kwon et al. (2025) connected abstract concepts to physical experiences through CS Unplugged and mixed-reality. Dai (2024) supplemented the unplugged approach with plugged activities and reflection components, enhancing embodied analogy to support AI literacy. Embodied classroom experiences can enhance attention, memory, and conceptual understanding (Sullivan 2018). This perspective aligns with work on embodied interaction, which emphasizes how students' sensorimotor engagement with learning environments can support conceptual development by surfacing intuitive, action-based ways of thinking (Abrahamson et al. 2020; Yang et al. 2024). These embodied experiences allow for a richer mental encoding of information which facilitates improved recall and use of that information, resulting in improved outcomes across subjects like reading, writing, math, and physics (Fugate, Macrine, and Cipriano 2019). We use the same principles to enhance the student experience of learning AI techniques.

Despite the effectiveness of CS Unplugged for engaging and motivating students, unplugged activities are rare in higher education. This research aims to fill the gap. A well-structured unplugged to plugged to reflection learning cycle could turn conceptual understanding into practical skill building and ethical awareness, aligning with the UNESCO framework for holistic AI education. To promote this cycle, our course curriculum incorporates unplugged activities and embodied interactions not only to teach AI concepts, but to support deeper engagement and reflection.

The Course

We present an exploratory case study using embodied, unplugged activities teaching artificial intelligence in higher education. The course, Introduction to Artificial Intelligence, is a general overview course of AI techniques and has four main topics: 1) Search, 2) Reasoning with Uncertainty (Markov Decision Processes and Q-Learning), 3) Probabilistic Reasoning through Time (Dynamic Bayes Nets, HMMs, Filtering), and 4) Machine Learning (Neural Networks, Decision Trees, Classification, Regression). The course is taught over the 12-week Summer 2025 semester.

Each of the main topics has an in-depth coding project. Students use Jupyter Notebooks to 1) develop algorithms to find a sequence of actions to navigate a house and find treasure, 2) approximate an optimal policy through repeated trials, 3) localize a hidden object, and 4) build a neural net from the ground up. After each coding project, students reflect on the applied algorithms' uses, benefits, and potential issues through Socratic Mind (Hung et al. 2024). This assignment structure is consistent with the Fall and Spring offerings.

During the summer semester, we modified the in-class lecture time to focus on unplugged experiences. One activity was conducted each class session, sometimes repeating an activity multiple times over several days to add complexity to the base set-up. Following the unplugged experience, students were guided through a plugged reconstruction of the simulation in Python Jupyter notebooks. The in-class instruction explicitly outlined how the unplugged game can be represented as an AI problem, transitioning from unplugged intuition to mathematical formalization, building toward the practical implementation in each project.

Activities

Our design goals for these activities were a combination of the defining characteristics of unplugged activities and more specific goals derived from the context and content of the course. Like any unplugged activity, students do not use computers, and instead use physical objects with paper and pencil. There is special emphasis on interaction with other students in a game-like atmosphere to encourage playing with concepts collaboratively. Where possible, we used familiar mechanics like dice and cards to foster a sense of play, to capitalize on students' prior experience with probabilistic reasoning, and to emphasize that probabilities involved in a problem are not necessarily arbitrary parameters, but can be an intrinsic part of the problem or application.

Because these activities are AI specific, we wanted to align students' decisions throughout the activities with the decision making process of AI agents. By highlighting first-person decision making, the activities expose the limitations an AI agent faces. They allow students to experience the steps of an algorithm for themselves and intuit how each step fits into the broader process. We frequently use hidden information to encourage step-by-step decision making instead of guessing a solution from a birds-eye view.

As tools for a university-level course, we wanted these activities to transition well into a more technical discussion. The activities are designed to fit neatly into the mathemat-

ical formalism of the broader problem, and they are simple enough to be solved by hand without being trivial. By using the activity to motivate the formal problem statement, the mathematics should feel more natural as well.

The activities are designed to be fully integrated into the course. After each activity, we had students articulate their reasoning and strategies in order to explicitly connect the ways students make decisions to the ways AI make decisions. Then we followed up with a mathematical formalism, now motivated by the activity and their own experience. Finally, students constructed an AI agent that uses the formalism to make decisions using Python, transitioning from unplugged to plugged.

We present in this paper four activities, but this is not an exhaustive list. We chose these activities to represent the broad scope of AI topics that could be taught using unplugged instruction. Complete instructional materials and additional activities are available.

Search

How does an agent know what actions to take to achieve a goal? We designed an activity, *Becoming Search*, that puts students in a first-person perspective and utilizes distributed information to search for a goal state. Search algorithms are typically taught from a birds-eye view, where all connections and state information are available at once. An AI agent using a search algorithm does not have the same perspective.

In prior semesters, we have observed that students often do not understand how an agent can make decisions only seeing one state at a time, when they, as humans, can see all states and connections in the search graph. Students also often misunderstand the value of a heuristic as an estimate and instead try to compute the actual distance of a state to the goal, rather than using local state information to estimate the distance. This activity is designed to combat these misunderstandings before students try to code an agent.

Students form groups of four. Each student in the group has a role. Each role has access to a subset of information about the world, represented on index cards like in Figure 1. The roles are:

- **The Successor Dictionary** - This role has access to all the connections in the state space graph in the form of a dictionary. When asked about a particular state, this role looks up the state in the dictionary and reports all of the connections.
- **The Goal Test** - This role is aware of goal conditions. If the goal is a single state, this role knows the state id. If the goal is a set of state information (e.g., 'You win when you are within 10 miles of Budapest.' or 'The key is in your inventory and you returned to the start.'), this role processes each state to see if the goal conditions have been met.
- **The Frontier** - This role uses a stack of index cards or handwritten notes to keep track of states to be visited. It inserts new states into the stack and when prompted, tells the Algorithm which state to process next.
- **The Algorithm** - This role orchestrates the algorithm's functionality by utilizing all the other roles. They keep

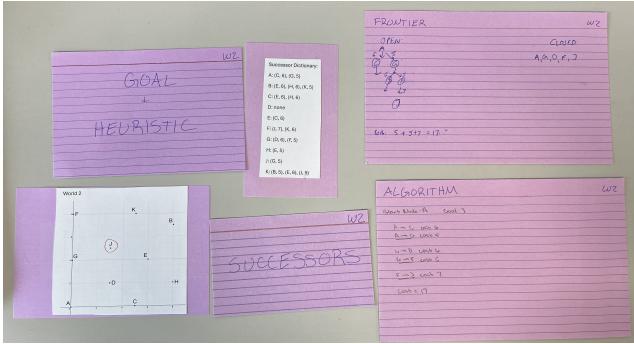


Figure 1: Sample cards that are given to each role with distributed information. Students use the cards to share information and trace through the search scenario.

track of the visited states and parent states. This role follows the pseudocode of the algorithm and calls upon the other roles to search for the goal.

Students start with an initial state, and the Algorithm uses the Successor Dictionary to populate the Frontier. The Frontier starts by using a queue, adding new states to the end and pulling the next state from the front. The Goal Test evaluates each state from the Frontier until a goal state is found. Students then swap roles and repeat the exercise on a new scenario so each student gets to experience every role.

The initial setup models Breadth-First Search, but simple changes in one role's function can change the process as a whole. To model Depth-First Search, change the function of the Frontier from using a queue, to using a stack (add new states to the end and pull the next state from the end). To highlight the importance of action sequences, have students report not just the states passed through to get to the goal, but the actions taken from each state. In this variant, the Successor Dictionary has not just the connections, but the names of those connections, such as 'go left', 'go right', 'go down stairs', etc. To model Uniform-Cost Search, modify the information given to the Successor Dictionary to include action costs, and have the Frontier use a priority queue (order the states by accumulated costs). To model Greedy Search or A* Search, modify the Goal Test role to also estimate the distance a state is from achieving the goal by specifying a heuristic and have the Frontier order states using the estimate. With this simple setup, students can see how minute changes in components allow the algorithm to handle different scenarios, without changing the algorithm's process.

Markov Decision Processes

How can an agent make decisions when the outcomes of those decisions are uncertain? We developed a game called *Red and Black Jack* (inspired by Blackjack or "21") to introduce students to Markov Decision Processes (MDPs), which formalize planning in stochastic environments.

Red and Black Jack is a single player game, but the activity breaks students into pairs so they can take turns acting as player and dealer to collaborate on strategies and check scores. Using a standard 52-card deck, setup a 'Hit' deck

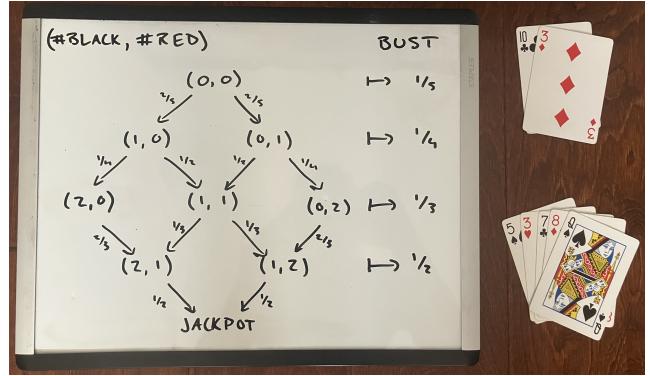


Figure 2: Sample Hit and Stand decks for Red and Black Jack. The "Hit" transitions for MDP that represents this game can be expressed in this grid format.

and a 'Stand' deck for each pair. In the 'Hit' deck, put two red cards, two black cards, and one face card (the other cards should be non-face cards to distinguish them). In the 'Stand' deck, put one red card and one black card. Each turn, the player can either 'Hit' or 'Stand'. When executing the 'Hit' action, the player draws one card from the 'Hit' deck, and as long as a face card was not drawn, play continues. Drawing a face card causes the player to 'bust', ending the game and rewarding the player with negative points. When executing the 'Stand' action, the player draws one card from the 'Stand' deck and the game is over.

At the end of the game, the player calculates their score based on the maximum number of cards of any single color. If the player drew a face card, they 'Bust' and earn -5 points, ignoring any other cards. If the player took the 'Stand' action and has at most 1 black or red card in their hand, they earn a 'Single' and score 1 point. If the player took the 'Stand' action and has at most 2 black cards or 2 red cards, they earn a 'Double' and score 5 points. If the player took the 'Stand' action and has at most 3 red or black cards, they earn a 'Triple' and score 15 points. If the player manages to 'Hit' and draw every card *except* the face card, they score a 'Jackpot' and earn 30 points.

Students can play multiple rounds and build an intuition for what they should do in each situation, and then discuss how to formalize the game as an MDP. In our formalization, we had 9 non-terminal states (the player has drawn 0 red cards and 0 black cards, or 1 red card and 2 black cards, etc.) and 5 terminal states ('Bust', 'Single', 'Double', 'Triple', and 'Jackpot') as visualized in Figure 2. Playing through the game eases the discussion of filling in the transition probabilities and the notion of a policy as a solution to the MDP. The size of the problem is quite small and has built in symmetry (having 2 red, 1 black is functionally the same as 1 red, 2 black), so students can perform value iteration by hand with relatively few computations. The algorithm converges within a few iterations, and we chose the rewards so the end values would be "nice" numbers.

Following formalizing MDPs, students can explore how different changes in setup can result in differing policy be-

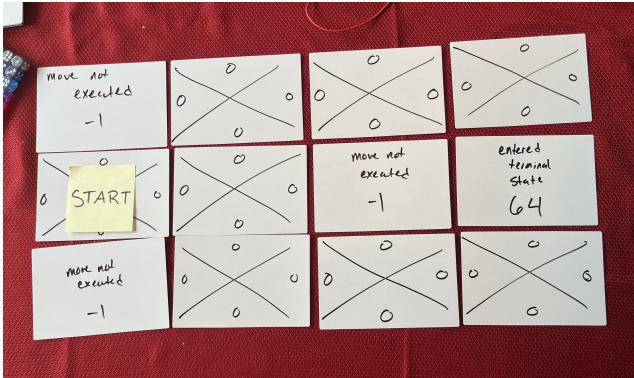


Figure 3: A sample GridWorld environment for Q-Maze.

aviors. For example, significantly increasing the value of ‘Jackpot’ will encourage more risky behavior, while a higher cost to drawing the ‘Bust’ card or having more ‘Bust’ cards will make the optimal policy appear more cautious. Students can make theoretical predictions about policy changes, then enact them by simulating the game in code or performing value iteration.

Q-Learning

How can an agent make decisions from rewards without prior knowledge of the transition probabilities? We designed an activity, *Q-Maze*, that uses each student as one pass of a q-learning agent through the environment. Collectively, students simulate the repeated trials of an agent and watch as the q-values distribute through q-table. Students often remember how to *use* a q-table (pick the action with the highest estimated future reward), but can struggle trying to *build* a q-table (picking the maximum valued action when all the q-values are zero may not be helpful). In addition, students often confuse the q-value, the agent’s future reward estimate, with a reward which is given from the environment. This first-person perspective is designed to help students differentiate between the two as well as understand how actions should be chosen before the q-table is filled up.

Start by setting up a grid on the floor (3x3, 4x4, 3x4, 5x5, etc as space allows). Each cell represents one state in GridWorld. Each state has a piece of paper or whiteboard on top of it. In the center of the paper is the ‘reward’ for this particular state. Along the outside of the paper are the actions available in each state and the q-value for each of those actions, as show in Figure 3. Each student makes one pass through the maze, with the number of allowable actions being the length of the grid plus the width of the grid. One ‘episode’ through the grid looks like:

- Flip over the whiteboard for the starting position to see the available movement options and anticipated reward.
- Roll a dice to see if you explore and pick a random action, or exploit the existing q-table and take the action with the maximum q-value
- Flip over the whiteboard for the chosen action and look at the actual reward

- Use the Bellman Equation to update the q-value of the action taken.
- Flip back over the previous state
- Repeat until reaching a terminal state or using up all actions

Each student should experience navigating through the grid when the q-values are all zeroes, and navigating when the q-values have mostly converged. This requires multiple grids for students to navigate at different points in time. Through trials in early states and in late stages, students will see the need for exploration in early trials and exploitation of the learned q-values in later trials. Once all students have explored the grids, flip over all the whiteboards and lead a discussion on the Q-table that the class converged to, how their individual experiences lead to a more complete picture of the optimal action in each state, and how to translate the converged q-values into an action policy. As an extension, you could modify the algorithm parameters, adjusting how frequently students explore/exploit or how much of the future reward transmits, or what percent of the original estimate is kept on each update. Students can feel the impact of each parameter by repeating the activity, or run their own experiments in a code implementation.

Hidden Markov Models

How can an agent determine its state given noisy sensor data? Up until now, an agent’s state has always been known deterministically. We designed an activity, *Two Spies*¹, to introduce students to Hidden Markov Models (HMMs). Students must hunt down and capture a spy roaming the country in deep cover. The players collect a series of observations after each action, which they put together into an informal ‘belief’ model to track down the most-likely true location of the hidden spy. The deep-cover spy simulates a Hidden Markov Model. Students roll dice to determine what state to transition into and again to determine what the observations report. Students get hands-on experience simulating both the transition model and the sensor model of an HMM.

During game play, the Deep Cover Spy transitions then reports an observation. Through play-testing, we found that making the observations refer to regions rather than individual cities led to more interesting inferences. The Deep Cover Spy records their actual state and the observation. The Hunter Spy then picks an action - Move, Stay, or Capture. The Hunter Spy moves deterministically to an adjacent state, stays in the current state, or attempts to capture the Deep Cover Spy in the current location. The Hunter Spy records the observation and their chosen action. Given the small state space, the players are given six rounds to either capture the opposing player or evade capture. Students play the game in pairs, swapping roles after three games.

After playing several rounds in both roles, students can formalize the HMM using mathematical notation. They define the transition matrix and sensor matrix and work through each discrete inference to see how these mathematical calculations reinforce their informal belief. They can also

¹Inspired by the mobile game Two Spies

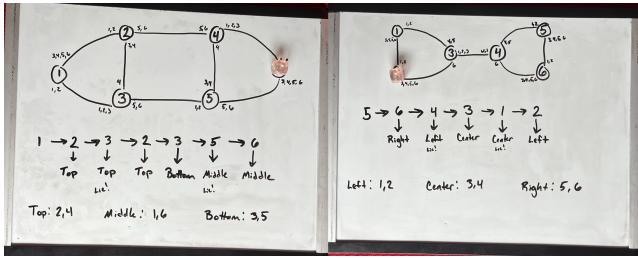


Figure 4: Two sample maps with the HMM transitions and observations, from the perspective of the Deep Cover Spy that simulates the HMM.

explore how the belief informs sequential decision-making as they build a policy for when to execute each action. After writing the exact inference algorithm in code, students can recreate and reassess their previously played games.

Reflection

Successes One of the most notable successes was high student engagement. Though we did not require attendance as part of the course grade, class sessions routinely attracted 75-80% of students through the end of the semester. In prior semesters with more lecture-based instruction, less than 25-30% of students still attended class sessions at the end of the semester. When students were in class, they were discussing loudly with each other their wins, decisions, and strategies. Every student offered up insight into their discoveries either verbally during the class-wide debrief or through text submission of their personal reflection.

When converting the unplugged simulation to a technical formalization of the problem, numbers that may appear as arbitrary choices when using an example like GridWorld now appear as obvious properties of the problem. Students could apply their familiarity with dice and card probabilities to decision-making under uncertainty. More than just being familiar with game mechanics, students could translate their own thinking and decision-making processes into algorithms that an AI agent uses.

Lindner, Seegerer, and Romeike (2019) noted that connecting student thought processes to the ways a machine learns makes it easier for K-12 students to comprehend artificial intelligence. This finding appears to hold in higher education. Students were able to reason about each scenario and design rough approximations of the algorithms used, often without prompting. Across all activities students linked their intuitions to formal solutions. For example,

- Groups collaborated and shared information to search for a path to a goal, then coded helper functions that mimicked each role in Becoming Search.
- Students quickly grasped the explore-exploit trade-off in Q-Maze, hoping for the “explore” dice roll when the Q-table was full of zeros and preferring to pick the maximum-valued action once more entries were filled in. They then used their experience to tune parameters in their Q-learning coding project.

- Students expressed joy at seeing their predicted optimal policy emerge by following the value iteration algorithm in Red and Black Jack, and explained how to modify the MDP to encourage different policy behaviors.

- Students spontaneously used the extra dice as tokens to represent their belief states, narrowing down the possibilities until only one state was plausible in Two Spies.

When we followed each activity with both a manual walk-through and a coding session, translating the activity into a computer-readable algorithm seemed obvious, since the scenario constraints were familiar properties of the game. Students were able to reason about the algorithm well enough to create a technical implementation in another scenario.

Challenges One of the challenges of a summer semester is that the course must be taught in twelve weeks instead of sixteen. As a survey course, Introduction to Artificial Intelligence exposes students to a variety of methods and losing four weeks reduces the amount of content we were able to cover. In addition, one of the main barriers to introducing active learning techniques is that professors can achieve greater content coverage via traditional lecture than by using active learning techniques (Petersen et al. 2020; Bonwell and Eison 1991). Running these activities did require extra time for setup, exploration, discussion, coding, and reflection. However, by spending more time on each activity, students were able to connect hands-on experiences to formal algorithmic representations, articulate the reasoning behind computational decisions, and retain key ideas beyond the specific examples presented in class. Judging a course only by how much content is included may not be the best representation of student takeaways.

In addition, AI courses in higher education often focus on technical details and require a strong foundation in mathematics and programming, barriers to students with math anxiety and low self-efficacy (Allen, McGough, and Devlin 2021). Students need extra support to feel confident and build the prerequisite skills required to succeed in university AI curriculum. For unplugged AI learning to succeed in higher education, it needs to support student confidence and scaffold instruction so students can take an active role in their learning.

Scale is still an issue for unplugged learning in higher education. This course had only 40 students and, while larger than a traditional K-12 class, this roster is significantly smaller than the Fall and Spring semesters which can have up to 300 students in a single section. Orchestrating these activities required significant involvement from the instructor and TAs. Making active learning accessible and easy for instructors to implement is a goal of this case study, and we hope that we have outlined activities that are self-contained and able to be self-managed by student groups.

Finally, the precedent for passive lecture in university course has been long established. Students may resist active learning because it violates their expectations of passive instruction (Gaffney, Gaffney, and Beichner 2010), and has a higher perceived workload (Shekhar et al. 2020). Despite these perceptions, evidence shows that active strategies produce better learning outcomes (Deslauriers et al. 2019). Cul-

tivating genuine student engagement may be a challenge, but it is what allows active learning to have its greatest impact.

Opportunities One of the biggest advantages to applying an unplugged curriculum at the university level was students' prior experience in algorithms and mathematics. Data Structures and Algorithms was a prerequisite to Introduction to AI, so most students were familiar with stacks, queues, multi-dimensional arrays, and algorithmic thinking. Rather than approaching topics like Breadth-First Search and Depth-First Search as brand-new topics, we demonstrated a new application and perspective of these algorithms. The transition from simulating our distributed perspective of search algorithms with index cards to coding a general version with helper methods was made easier because we could rely on the students' prior knowledge and experience in computer science.

We saw similar benefits when learning about MDPs and HMMs. Students had prior experience at least with the format of matrices, multi-dimensional arrays, and probability. Instead of introducing the concept for the first time, we could focus on refreshing and remediating to build confidence with a familiar data structure. Since students already had foundational probability skills, they could easily reason about the chance of drawing a face card from a diminishing deck or traversing a graph from dice rolls. The instructional team could spend less time working through foundational mathematics and programming concepts and accelerate to more complex technical details while providing individual support and remediation where necessary.

Most unplugged instruction is aimed at younger students in K-12 courses. As such, instruction focuses on conceptual understanding but can gloss over the deep technical details. Even transitioning to plugged activities in Scratch or Teachable Machine can still hide some of the unique complexities of AI algorithms. Younger students may not have the necessary background to understand highly technical implementation details, but university students often do. Unplugged instruction in higher education has the opportunity to build toward a holistic AI education, from conceptual understanding to the technical underpinnings to implementation from scratch because of the advanced learning and cognitive abilities of university students.

Future Versions We found that not all students were willing to vocalize their reasoning during the activity and post-activity discussions. To make the benefits of reflection more accessible, we would ask *all* students to explicitly outline their procedure in writing or verbally. We believe this could help students better compare their informal understanding to the algorithm's formal representation. For example, in the Two Spies activity, we did not fully unpack how students' inferences based on observations relate to the transition model, nor did we work through the matrix multiplications by hand. In the future, we would walk through these calculations step-by-step to show how the belief probabilities align with their informal guesses of the spy's location at each time-step. We would also have students propose an algorithm for how to combine the transition probabilities with the observation probabilities to update their belief. Asking

"What did you do, and how is that similar to what the machine is doing to learn?" can help students recognize parallels between their strategies and the underlying algorithms.

In order to create more continuity between modules, it may be helpful to use multiple AI approaches with each scenario. GridWorld is a powerful example because it is flexible and can be used to demonstrate search algorithms, action stochasticity, sensor stochasticity, and a combination of real-world robot navigation challenges. Our activities can also be solved by a variety of AI methods. For example, to use Q-learning with Red and Black Jack, we could hide the contents of the 'Hit' and 'Stand' deck and scoring rewards and let students approximate the transition probabilities and rewards through playing the game multiple times. Or to apply Particle Filters to Two Spies, students could use small tokens like M&M's or Skittles to approximate the particles.

We would also adjust the pacing of the activities by spreading them out across multiple days, allowing us to introduce new components and complexities incrementally. In our current schedule, we ran several variants of the Search activity over two days, which limited the attention students could give to the impact of each change, like the frontier structure, action costs, and heuristics. Extending these over more class sessions would provide space for deeper analysis of each modification. Similarly for the other activities, a multi-day structure would allow students to internalize the mechanics and be better prepared for additional scenario complications. For Red and Black Jack, another session could introduce variants like changing the scoring system or adjusting the composition of the Hit or Stand decks. Students could explore the impact of different parameters, like the future reward discount, the reward structure, or the transition probabilities on the optimal policy. Similarly, for Q-Maze, we would follow-up by adjusting the parameters (like the exploration rate, discount factor, learning rate) to explain the impact of each parameter. Students could play another round of Two Spies attempting to narrow down the true location faster using the inference algorithm, or simulate a particle filter on the map instead of exact inference.

Conclusion

This case study proposes that integrating embodied, unplugged activities into higher education AI courses can foster sustained engagement and stronger connections between human decision-making and algorithmic reasoning. By giving students a first-person perspective on AI techniques before transitioning to formal mathematical models and coding, we lowered barriers to entry, improved retention of core concepts, and encouraged active participation. This approach gave students a stronger intuitive grasp of core concepts and increased student confidence in applying them. Future work should explore scaling these methods to larger classes, incorporating multi-day activity structures, and applying several AI methods to each activity. Ultimately, unplugged AI education at the university level has the potential to bridge the gap between conceptual reasoning and technical skill, preparing students not just to implement AI systems, but to understand and evaluate them in real-world contexts.

Acknowledgments

This project is supported by National Science Foundation under Grant No. 2247790 and Grant No. 2112532. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Abrahamson, D.; and Lindgren, R. 2014. *Embodiment and Embodied Design*, 358–376. Cambridge Handbooks in Psychology. Cambridge University Press.

Abrahamson, D.; Nathan, M. J.; Williams-Pierce, C.; Walkington, C.; Ottmar, E. R.; Soto, H.; and Alibali, M. W. 2020. The future of embodied design for mathematics teaching and learning. In *Frontiers in Education*, volume 5, 147. Frontiers Media SA.

Allen, B.; McGough, A. S.; and Devlin, M. 2021. Toward a framework for teaching artificial intelligence to a higher education audience. *ACM Transactions on Computing Education (TOCE)*, 22(2): 1–29.

Alraddady, S.; Luong, D.; and Young, G. 2014. A Study of Kinesthetic Learning Activities Effectiveness in Teaching Computer Algorithms. In *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)*, 1. The Steering Committee of The World Congress in Computer Science, Computer

Battal, A.; Afacan Adanır, G.; and Gülbahar, Y. 2021. Computer science unplugged: A systematic literature review. *Journal of Educational Technology Systems*, 50(1): 24–47.

Bonwell, C. C.; and Eison, J. A. 1991. *Active learning: Creating excitement in the classroom*. 1991 ASHE-ERIC higher education reports. ERIC.

Dai, Y. 2024. Integrating unplugged and plugged activities for holistic AI education: An embodied constructionist pedagogical approach. *Education and Information Technologies*, 1–24.

De Barros, V. A. M.; Paiva, H. M.; and Hayashi, V. T. 2023. Using PBL and agile to teach artificial intelligence to undergraduate computing students. *IEEE Access*, 11: 77737–77749.

Deslauriers, L.; McCarty, L. S.; Miller, K.; Callaghan, K.; and Kestin, G. 2019. Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. *Proceedings of the National Academy of Sciences*, 116(39): 19251–19257.

Fugate, J. M.; Macrine, S. L.; and Cipriano, C. 2019. The role of embodied cognition for transforming learning. *International Journal of School & Educational Psychology*, 7(4): 274–288.

Gaffney, J. D.; Gaffney, A. L. H.; and Beichner, R. J. 2010. Do they see it coming? Using expectancy violation to gauge the success of pedagogical reforms. *Physical Review Special Topics—Physics Education Research*, 6(1): 010102.

Hingle, A.; and Johri, A. 2023. A Research Study on Student Conceptions of Artificial Intelligence. In *ASEE Southeast Section Conference*.

Hu, W.; Huang, R.; and Li, Y. 2024. Young children’s experience in unplugged activities about computational thinking: From an embodied cognition perspective. *Early Childhood Education Journal*, 52(4): 769–782.

Huang, W.; and Looi, C.-K. 2021. A critical review of literature on “unplugged” pedagogies in K-12 computer science and computational thinking education. *Computer Science Education*, 31(1): 83–111.

Hung, J.-T.; Cui, C.; Popescu, D. M.; Chatterjee, S.; and Starner, T. 2024. Socratic Mind: Scalable Oral Assessment Powered By AI. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, 340–345.

Kwon, K.; Kim, H.; Kim, K.; Seo, M.; and Brush, T. A. 2025. Computational thinking through embodied learning: a comparison of mixed-reality and unplugged activities. *Journal of Research on Technology in Education*, 1–17.

Lao, N. 2020. Reorienting machine learning education towards tinkerers and ML-engaged citizens. *Massachusetts Institute of Technology Cambridge*.

Lim, H.; Min, W.; Vandenberg, J.; Cateté, V.; and Mott, B. 2024. Unplugged K-12 AI Learning: Exploring Representation and Reasoning with a Facial Recognition Game. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 23285–23293.

Lindner, A.; Seegerer, S.; and Romeike, R. 2019. Unplugged Activities in the Context of AI. In *International conference on informatics in schools: Situation, evolution, and perspectives*, 123–135. Springer.

Long, D.; and Magerko, B. 2020. What is AI literacy? Competencies and design considerations. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, 1–16.

Long, D.; Moon, J.; and Magerko, B. 2021. Unplugged assignments for K-12 AI education. *AI Matters*, 7(1): 10–12.

Ma, R.; Sanusi, I. T.; Mahipal, V.; Gonzales, J. E.; and Martin, F. G. 2023. Developing machine learning algorithm literacy with novel plugged and unplugged approaches. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 298–304.

Markov, Z.; Russell, I.; Neller, T.; and Coleman, S. 2005. Enhancing undergraduate AI courses through machine learning projects. In *Proceedings Frontiers in Education 35th Annual Conference*, T3E–21.

Marx, E.; Leonhardt, T.; and Bergner, N. 2022. Brief summary of existing research on students’ conceptions of AI. In *Proceedings of the 17th Workshop in Primary and Secondary Computing Education*, 1–2.

Miao, F.; and Shiohira, K. 2022. K-12 AI curricula. A mapping of government-endorsed AI curricula. *UNESCO Publishing*, 3: 60.

Ng, D. T. K.; Lee, M.; Tan, R. J. Y.; Hu, X.; Downie, J. S.; and Chu, S. K. W. 2023. A review of AI teaching and learning from 2000 to 2020. *Education and Information Technologies*, 28(7): 8445–8501.

Ng, D. T. K.; Leung, J. K. L.; Chu, S. K. W.; and Qiao, M. S. 2021. Conceptualizing AI literacy: An exploratory review. *Computers and Education: Artificial Intelligence*, 2: 100041.

Petersen, C. I.; Baepler, P.; Beitz, A.; Ching, P.; Gorman, K. S.; Neudauer, C. L.; Rozaitis, W.; Walker, J.; and Wingert, D. 2020. The tyranny of content: “Content coverage” as a barrier to evidence-based teaching approaches and ways to overcome it. *CBE—Life Sciences Education*, 19(2): ar17.

Shekhar, P.; Borrego, M.; DeMonbrun, M.; Finelli, C.; Crockett, C.; and Nguyen, K. 2020. Negative student response to active learning in STEM classrooms: A systematic review of underlying reasons. *Journal of College Science Teaching*, 49(6): 45–54.

Sullivan, J. V. 2018. Learning and embodied cognition: A review and proposal. *Psychology Learning & Teaching*, 17(2): 128–143.

Varela, F. J.; Thompson, E.; and Rosch, E. 2017. *The embodied mind, revised edition: Cognitive science and human experience*. MIT press.

Vargas, M.; Nunez, T.; Alfaro, M.; Fuertes, G.; Gutierrez, S.; Ternero, R.; Sabattin, J.; Banguera, L.; Duran, C.; and Peralta, M. A. 2020. A project based learning approach for teaching artificial intelligence to undergraduate students. *Int. J. Eng. Educ.*, 36(6): 1773–1782.

Yang, W.; Hu, X.; Yeter, I. H.; Su, J.; Yang, Y.; and Lee, J. C.-K. 2024. Artificial intelligence education for young children: A case study of technology-enhanced embodied learning. *Journal of Computer Assisted Learning*, 40(2): 465–477.