Balancing Act: Mastering Beam-and-Ball Control with Reinforcement Learning

Tanushree Burman, Jevon Coney, Chris Rogers, Jennifer Cross, and Jivko Sinapov

Tufts University, Medford MA 02155, USA

Abstract. Current AI education concepts often emphasize supervised machine learning algorithms, while reinforcement learning (RL) education remains limited, typically focusing on introductory concepts. To address the need of a more in-depth RL educational experience, we propose two hands-on robotic activities to introduce foundational RL concepts and foster exploration of RL system design principles. These activities include a LEGO® car task and a beam-and-ball balance robot simulation. We conducted a pilot study featuring this two-part curriculum with 13- to 16- year old students and analyzed their learning outcomes quantitatively. Our findings indicate that middle and high school students developed an understanding of basic RL concepts as measured by post-activity reflections. We also discuss next steps to enhance the curriculum, including providing a more interactive experience with RL system design.

Keywords: Reinforcement Learning, AI Education, Robotics

1 Introduction

Recent advancements in artificial intelligence have significantly diversified how we interact with AI-powered technologies and devices. Some of these technologies introduce automated systems such as fraud detection and autonomous vehicle services such as Waymo. Others offer personalized interactions through platforms such as home voice assistants and recommendation systems in streaming platforms. This transformation of AI technology in everyday applications has resulted in initiatives such as AI4K12 to integrate AI education in K-12 curricula. AI4K12 highlights foundational concepts that all students should know about AI such as how computers perceive the world through sensors, how agents represent their surroundings and how they learn from data (1).

Publicly accessible AI tools such as *Teachable Machine*[1] and *AI for Oceans*[2] are available to introduce concepts of supervised machine learning to young programmers. While some pre-college programs utilize these tools, others integrate AI concepts through robotics for more tangible exploration. The micro:bit and

[1] https://teachablemachine.withgoogle.com
[2] https://code.org/oceans

the ml-machine.org [3] webpage were developed to enable students to redesign artifacts such as a life vest and whisk through supervised machine learning (2). Additionally, the Smart Motor system was developed to teach students about classification by training motors to respond to different sensor inputs (3). Students can train the motor to run to various positions corresponding to different sensor inputs that enable them to bring their engineering projects to life.

Much of the rapidly growing AI education concepts focus on supervised machine learning algorithms such as classification, and often overlook other types of machine learning such as reinforcement learning (RL). Reinforcement learning is a type of machine learning in which the agent learns an optimal behavior through positive and negative rewards received through repeated interactions with its environment. The limited but growing reinforcement learning education technology typically focuses on introductory concepts through grid-world environments and virtual exploration. In an effort to introduce a more in-depth RL educational experience, we propose two hands-on robotic activities to introduce basic concepts and foster exploration of RL system design principles. Our work expands on previous RL education efforts that utilize the LEGO® Spike Prime robot to introduce core concepts (4). The activities in this study include a LEGO® car activity and an interactive beam and ball balance robot and simulation. Our curriculum introduces core concepts such as agent-environment interaction, reward system, epsilon-greedy policies as well as more advanced topics such as observation space parameters and the impact of system design on agent performance. We evaluate this two-part curriculum with nine 13- to 16-year old students and conduct a quantitative analysis of students' learning outcomes. Our findings indicate that students developed an understanding of basic RL concepts. We also discuss next steps to enhance the curriculum, including providing a more interactive experience with RL system design.

2 Related Work

Some of the limited but growing reinforcement learning education available focuses on introducing basic concepts through virtual exploration. In efforts to introduce the core ideas behind RL to the general audience, a game was developed with Virtual Reality where the player takes the role of the autonomous agent and must learn the shortest path to a hidden treasure through experience (5). In this game, the playing user has limited observations, using direct experiences to demonstrate how new knowledge is acquired and exploited by the agent. This application allows users to get a first hand experience as the autonomous agent. In a similar application, ARtonomous is a tablet-based tool for learning about and generating reinforcement learning models for robot navigation (6). This iPad application was used to teach middle school students how to train a school bus to follow a bus route course of their choice.

Efforts have also been made to encourage tangible exploration of reinforcement learning. One study introduces RL to high school students through a robot-

[3] https://ml-machine.org/

based activity using a LEGO® Spike Prime robot and an Augmented Reality mobile app to visualize the training process (4). The students designed their own LEGO® Spike robot without wheels and trained it to move in a straight line using RL. The AR mobile app allowed for visualization of the Q-table to better understand the decision-making process. Another paper discusses the use of LEGO® Mindstorms robots for interactive RL instruction in a master's level course (7). Here, students were provided with code templates to implement a wandering task and were then challenged to extend it by incorporating additional states and actions.

To incorporate hands-on exploration of RL concepts using a LEGO® robot and introduce more advanced topics that encourage students to explore additional RL parameters, we developed a two-part curriculum targeted at middle and high school students. This curriculum, accompanied by a quantitative analysis of learning outcomes, was designed to be accessible without requiring prior programming experience.

3 Reinforcement Learning Background

Reinforcement learning is a sequential decision-making process where an agent learns to map states to actions through repeated interactions with its environment by maximizing a numerical reward signal (8). The problem of reinforcement learning can be formalized through a Markov Decision Process (MDP) that is defined by the tuple $\langle S, A, T, R, \gamma \rangle$. Here, S represents a set of states, A is a set of actions that can be taken in each state, T is the state transition probability, R is a set of possible rewards and $\gamma \in [0, 1]$ is the discount factor.

At each time step t, the agent receives some representation of the environment's state, $S_t \in S$, and selects an action, $A_t \in A(S_t)$. One time step later, as a result of the action, the agent receives a numerical reward, $R_{t+1} \in R$ and is in a new state S_{t+1}. The agent's goal is to maximize the total amount of reward it receives over time, or the expected return. At each time step, the agent maps states to probabilities of selecting each possible action. This mapping is called the agent's policy $\pi : S \to A$. The value of a state s under a policy π, denoted $v_\pi(s)$, is the expected return when starting in s and following π (8). The agent interacts with its environment for a fixed number of timesteps or until it reaches the goal state, forming a single episode. The agent is then reset to its initial position and this process is repeated for a specified number of episodes.

The goal of an RL agent is to learn an optimal policy and an optimal value function that guarantees the maximum expected return. Q-Learning, a method for this, first initializes state-action pairs to zero. The agent then selects an action, finds itself in a new state and receives a reward. The value of this state given the action it selected and the reward it received is then updated using the following equation: $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$. Here α is the learning rate or the step size parameter that controls how much new information is incorporated into $Q(s, a)$. It is usually a value $\alpha \in [0, 1]$. The agent selects the action with an ϵ-greedy policy where given some probability $\epsilon \in [0,1]$, an action with the highest Q-value is selected or a random action is selected to encourage exploration of new states and actions. The observation space is information the agent receives from its environment that is used as input to the RL algorithm.

4 System Overview

Our goal is to introduce basic RL concepts to students through (1) a hands-on LEGO® Spike Prime robot activity and (2) encourage exploration of RL system design through a beam and ball balance robot and simulation.

4.1 LEGO® Spike Prime Robot Activity

In this activity, the students were given a pre-built LEGO® Spike Prime car robot and tasked to train the car to move to a goal color on a colorful path. This robot has a color sensor that is used to determine the state or color which it is currently on, 2 motors, 4 wheels and the robot can move forwards or backwards (Figure 1). The students were first given 5-8 pieces of different colored construction paper used to form a custom linear path for the car. Students were then provided with the RL Python code. Students used the *CodeRobots.ai*[4] webpage to connect to their robots, enter the Python code and train their robots in real time. Students were able to edit the starting/goal state for the robot, the ϵ value, and also the reward function to their liking. No Python experience was required, and key code sections were explained to students for easy modification.

Students connected their robot to a laptop, set it at the initial state and pressed the right button on the robot to begin training. The robot moved for a certain number of time steps per episode, with the Q-table displayed on the website after each episode. Students repeated this process until the robot learned to reach the goal state in a few time steps.

4.2 Beam and Ball Balance Robot Activity

The beam and ball system is a classic control problem where the goal is to stabilize the position of a ball to a goal point as it rolls on a beam by adjusting the beam's angle. This is a challenging problem that can be implemented using a Proportional Integral Derivative (PID) controller but determining optimal parameters can be complex and can require a lot of trial and error. Additionally, it requires knowledge of how the parameters in the PID control loop affect the system to successfully tune the controller. We drew inspiration from this system and implemented this activity with reinforcement learning as it can provide a more intuitive way to understand the system. A physical prototype of the beam and ball system was first developed using LEGO® pieces. Then a simulation of the system was developed in PyBullet and trained using RL.

[4] CodeRobots.ai

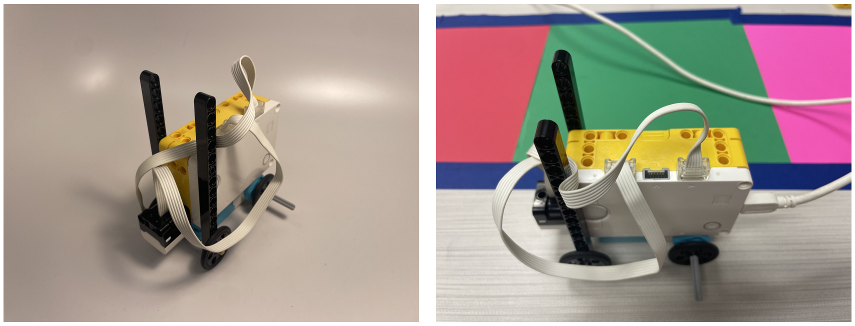


Fig. 1: (left) LEGO® Spike Prime Robot Activity: Students were provided with a pre-built robot car with 4 wheels, two motors and a color sensor. (right) Students were given different colored construction paper so they can form a colorful linear path for the robot to navigate to the goal color/state.

In this activity, students were first provided with a physical variation of the beam and ball system where they had to balance a car instead of a ball (Figure 2 right). This beam is attached to a motor on one end that controls the angle of the beam and a distance sensor on the other end to sense the location of the car. Students were tasked to balance the car at the center of the beam by controlling the angle of the beam with a remote. After gaining hands-on experience with the beam and car balance robot, students were given the opportunity to explore the RL design of the system with a website.

We developed an interactive website that has a beam and ball balance simulation that students can train (Figure 3). The application was built using a combination of Flask for the web framework and PyBullet for the simulation, and is easily accessible through any web browser. The website includes multiple interactive pages to help guide users through the RL process. A schematic of the website composition is shown on Figure 2 (left).

Simulation Page: This page allows users to manually control the beam and ball system, providing them an opportunity to understand the simulation from the agent's point of view. Users can attempt to balance the ball at the goal setpoint (which initially starts at the center of the beam) by using their left and right arrow keys to adjust the beam's angle in real time. The performance of the user is visualized through a live plot titled "Reward Over Time." Positive rewards (displayed as green bars) are awarded when the ball aligns with the goal state (marked by a blue bar) while penalties (red bars) are shown when the ball deviates from the goal state. Additionally, the page includes custom features that users can adjust before launching the simulation. These options allow users to set the ball's starting position and the goal setpoint position.

New Training Page: The New Training page introduces students to the agent's observation space. Students can customize the system by selecting observation space parameters, or how much information the agent knows from its environment. This includes the position of the ball, velocity of the ball and/or

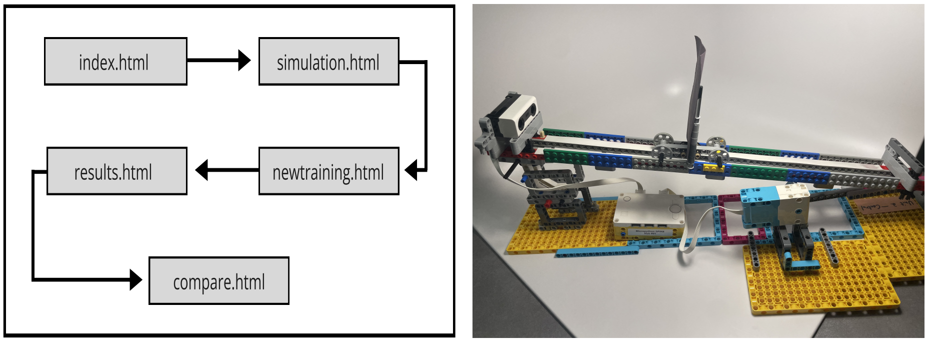


Fig. 2: (left) Website Composition Schematic: From the home page, the Simulation page, New Training page, Results page and the Compare page can be accessed. (right) Beam and Car Balance Robot: Students were tasked to balance the car at the center of the beam by controlling the angle of the beam with a remote.

current angle of the beam. Students can select any combination of these parameters in the observation space and examine how they affect agent performance.

Results Page: This page presents training outcomes based on the selected observation space, featuring three GIFs showing results after 1000, 3000, and 5000 training episodes. This page also includes a reward-over-episodes graph averaged over five runs.

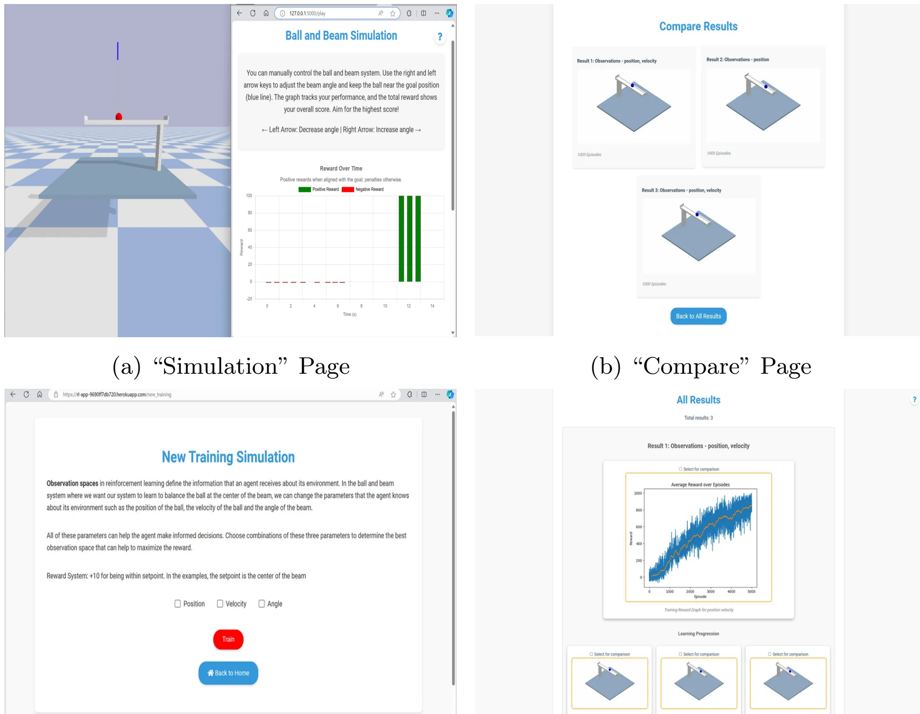
Compare Page: The Compare page allows students to analyze plots and GIFS from trainings with different observation spaces. It encourages exploration of how observation space parameters and training episodes impact the agent's performance.

Reinforcement Learning Environment and Model: The website's beam and ball balance system is pre-trained using Q-Learning with all observation space combinations to accelerate training, as real-time learning is slow. The results are stored such that when students train their model, the pre-trained results are displayed. In this reinforcement learning system, the available actions include adjusting the angle of the beam up, down, or keeping it unchanged by a discrete amount. The reward function assigns a reward of +10 if the ball is within a threshold around the goal point, +5 if the ball is moving toward the goal point, and -1 otherwise. As tabular Q-Learning is used, all continuous values are converted into discrete values.

5 Methodology

We conducted a study to assess how well the robotics activities achieve our learning objectives:

1. To what extent do the activities enhance students' understanding of Reinforcement Learning?



(a) "Simulation" Page (b) "Compare" Page

(c) "New Training" Page (d) "Results" Page

Fig. 3: Interface of Beam and Ball Website

2. How effectively can students grasp the design of a Reinforcement Learning system and factors that influence its performance?

We developed post-activity survey questions for students guided by these research questions.

5.1 Pilot Study Overview

A 2.5 hour study was conducted with nine 13- to 16-year-old students (5 male and 4 female). Our goal in this study was to introduce students to basic RL concepts through a hands-on robotics activity and to introduce how RL system design affects agent performance. The study was conducted at the Tufts Center for Engineering Education Outreach, involving student volunteers from the local Medford area who were compensated for their participation. Students had a diverse range of prior coding experience and limited familiarity with RL. 56% of students identified as beginners in coding, 22% as intermediate, and 11% as advanced, while 11% reported having no prior coding experience. The majority of students (67%) indicated they had no prior familiarity with RL, while the remaining one-third (33%) reported some level of familiarity.

5.2 Data Collection

The primary data collection method were survey questions to assess student's understanding of RL concepts and its application to the activities. After each session, students answered questions based on the activity. Video recordings of the sessions were also captured. Responses to the post-activity survey questions were reviewed and coded by a group of 12 education and engineering graduate students, with each response reviewed by six group members. The first author performed a thematic analysis of these open codes, creating a list of themes that was later reviewed by the second author. These finalized themes were then used to recode the questions, and the resulting analysis are presented here.

5.3 Study Design

The study was conducted in two sessions, each consisting of a one-hour long activity. The first session consisted of the LEGO® robot activity, while the second session featured the beam and ball balance activity. Prior to the first session, students received a 10-minute introductory presentation on RL. This presentation covered key concepts, including agent-environment interactions, actions, rewards, observation spaces, Q-tables and the ϵ-greedy policy balancing exploitation versus exploration. Following the presentation, students participated in the LEGO® Spike Prime robot activity. Working in pairs, they designed a path with five distinct colors, each representing a unique state. They were tasked with specifying both a starting color and a goal color for the LEGO® car. The objective was for the car to autonomously learn to navigate from the starting state to the goal state. Students observed how the car adjusted its movements based on environmental interactions, gaining insight into how reinforcement signals shaped its learning process. This hands-on activity provided a concrete demonstration of RL concepts, such as states, actions, rewards, and penalties.

The second session centered on the beam and ball balance robot and web application. The web app simulated a ball and beam system, where the objective was to learn how to balance the ball at a specific goal point on the beam. The students explored the web application individually, training the system with different combinations of the observation space. Students experimented with the system, observed the agent's learning process, and analyzed how different observation parameters impacted the agent's performance.

At the end of each session, students completed 10 minute post-activity reflections to assess their understanding of RL concepts and to document their learning experiences. These reflections provided insight into how well students comprehended the material and where further clarification might be needed.

6 Results

6.1 LEGO® Spike Prime Robot Activity

After the first session, the post-activity reflections consisted of questions aimed at evaluating students' understanding of basic RL concepts and its application to the LEGO® car activity. The responses to each question are described below.

What is Reinforcement Learning? Our analysis revealed that 78% of students mentioned rewards, 56% referred to training and 34% acknowledged an agent. We categorized the survey responses into three tiers based on their accuracy. The first tier, representing 33% of responses, provided the most accurate descriptions, correctly defining reinforcement learning as teaching or training through actions that are rewarded. The intermediate tier, comprising of 22% of responses, mentioned training through rewards but did not specify actions. The remaining responses were categorized as vague, as they included key reinforcement learning terms but lacked details about the process. Examples of these vague responses include statements like "Rewarding an algorithm when it does its job successfully" or "Teaching through rewards, aiming at completion or efficiency." Notably, only one response incorporated all key elements– agent, environment, actions and reward– stating that "Reinforcement learning is training an agent using an environment which gives rewards based on actions." Additionally, we observed that all responses focused exclusively on positive reinforcement, omitting the role of negative rewards.

How did you use RL in the LEGO® car activity? We found that 88% of the responses included at least one RL-related term, such as training, reward, goal, Q-value, or episode. We also categorized this question into three tiers based on accuracy. We found that 22% of the responses were classified into the first tier where students provided detailed explanation of how they applied RL in the activity. In these responses, students mention positive rewards associated with correct actions and negative reward with incorrect actions, with one student noting "The algorithm tries to get the most points in 1 episode." Another 33% were in the intermediate tier, where students referenced some part of RL used in the activity, like selecting actions based on the highest q-value or receiving negative reward for a "wrong move." The remaining 44% focused on the goal of the activity rather than explaining how RL was applied, with examples like, "I used it to train the car to go to the red square" or "We trained the LEGO® car to go to the color we wanted it to reach."

What did you implement as extra credit during the LEGO® car activity? Students were given the opportunity to implement extra credit by modifying parameters in the code, such as the number of episodes, timesteps, or epsilon value. We found that 67% of students implemented extra credit and most opted to adjust epsilon. Responses revealed a variety of approaches, with some students experimenting with different values and extremes to increase randomness. One student shared,"We changed the epsilon to 0.2, which showed more errors in learning since the robot took more random actions than 0.1." Another student explored the impact of varying epsilon between extremes, saying, "I changed the epsilon values to 1 (completely random) and 0, mostly rational. The higher the epsilon is, the higher the timesteps." In both cases, students highlighted the higher epsilon value leading to more random behavior, with the second response also noting the long-term impact of a higher epsilon. Two other students mentioned adjusting the goal states and timesteps.

What would happen if you set the exploration rate to 0? What if you set it to 1? We found that 77% of students recognized the impact of the exploration rate on action selection, focusing mainly on immediate effects such as more random actions due to a higher exploration rate. Only 33% considered long-term outcomes, noting how exploration rate affects learning performance and training time. One student explained, "If you set the exploration rate to 0. The robot will learn the correct path faster since it'll take no random actions. With a exploration rate of 1 it will take a lot longer to go to the correct color since the robot will take more random decisions." While this response is partially correct– since an exploration rate of zero prevents exploration of new states and may lead to a suboptimal solution– it demonstrates the student's attempt to understand the long-term effects of exploration on agent learning and performance.

Describe a new example of RL All students provided RL examples, varying in detail. 44% included clear goals, rewards, and feedback for correct or incorrect actions. One student described, "A mouse in a maze that wants to find cheese. When it finds cheese, it will get a reward, but if it runs into a cat, it will get punished." Another mentioned, "Little kids doing naughty behavior then getting yelled at by parents to not anymore." Other responses were more broad, such as "Teaching a baby to stand up" or "Getting a robot to solve a maze." Students also extended their understanding to robot training for optimal racetrack navigation.

6.2 Beam and Ball Balance Robot Activity

After the second session, students reflected on agent performance and RL design. The first two questions required students to view combinations of observation space parameters in the website and report the most effective and least effective for training the system. Subsequent questions introduced a new RL scenario, prompting students to explain how they would approach designing an RL system for this situation.

Most Effective and Least Effective Observation Space Parameters In the beam and ball balance website, students could select combinations of "Position," "Velocity," and "Angle" to observe how these observation space parameters influenced learning performance. "Position" represents the current position of the ball, "Velocity" refers to its speed, and "Angle" indicates the beam's current tilt. Since the system was trained using Q-Learning, the chosen combinations of observation space parameters impacted both the size of the Q-Table and the agent's performance. This activity aimed to encourage students to explore various parameter combinations, theorize why certain combinations might enhance the agent's performance, and gain insights into learning curves.

For the most effective observation space, all students were able to identify that the learning curve was better when there was a combination of at least two of the parameters. "I found the most effective combination to be position, velocity and angle. I believe this is because if it has more information about its parameters it can draw more specific conclusions on what worked and what didn't." 33% of the students connected the performance to the amount of information in the observation space. Another student wrote, "Angle and position is most effective. I would expect angle + position + velocity to get more reward but it was worse." This student observed that using all three parameters resulted in worse performance compared to using just two, highlighting a discrepancy between their expectations and the actual outcome. One of the reasons for this observation was that all three parameters in the observation space significantly increased the size of the Q-Table resulting in a greater number of states for the agent to explore and ultimately slower learning. All students noted that having only one of the observation space parameters resulted in a poor performance. One student wrote "Just angle and just position performed poorly because they didn't have enough data to make changes based off."

New RL Scenario The new scenario involved training a LEGO® Spike Prime robot to move in a straight line without wheels. This question was inspired by one of the robotic activities in (4). Students were first asked what information they would include in the observation space if they were developing an RL system for this problem. In other words, what would the robot have to know from its "environment" to learn to move in a straight line. The goal of this question was to assess if students can apply concepts of RL to a new scenario. We found that 67% of the students correctly mention motor speed and/or motor angle would be important. The rest of the students mention distance of the robot.

The next part of the question asked students to consider what additional components are essential in designing a reinforcement learning system for enabling the LEGO® Spike Prime robot to learn straight-line movement. The purpose of this question was to assess how many students recognize the important aspects of an RL system such as the reward function, the action and state space. We found that 56% of the students mentioned having a reward system and 22% of students referred to the current state of the robot.

The final question asked students what additional parameters they would modify on the beam and ball website if given the option. This question aimed to evaluate students' curiosity and their understanding of the adjustable parameters within an RL system. Responses varied widely, with 22% of students expressing a desire to adjust physical parameters of the system, such as the weight or size of the ball, and the thickness of the beam. Another 22% proposed the ability to modify key reinforcement learning elements, such as the reward system and exploration rate. The rest of the students were unsure of what additional changes they would find useful.

7 Discussion

We observed that the student's enthusiasm for both the LEGO® car activity and the beam and ball balance activity was high, particularly among those new to LEGO® Spike Prime robots. However, since this was a single-day study, student fatigue was evident in the second activity. More hands-on elements in the latter half of the workshop or the separation of activities over multiple days may have further boosted engagement. Our quantitative evaluation demonstrates that students developed an understanding of fundamental RL concepts. All students described a new RL example and identified key terms. However, the depth of their discussion on all the key components of RL varied. We observed that students retained a general idea of the purpose of exploration rate, but future studies can provide opportunities for students to explore how it impacts the long-term learning process and other exploration strategies such as epsilon decay.

Students generally had a more challenging time answering the questions after session 2. The students were able to identify the impact of different combinations of observation parameters on learning performance. However, their reasoning for these differences can be improved through more interactive demonstrations of the agent's perspective. Students encountered difficulties determining appropriate observation space components for the new straight-line LEGO® car scenario. To address this, future iterations can include a more interactive website where students can explore the impacts of the chosen observation space in more detail. For example, students can view not only the learning performance, but also the specific parameter values that are used as input to the algorithm and how it affects the size of the Q-table. The website can be enhanced with interactive elements, such as toggles, to explore other key aspects of RL systems, including different reward policies. This increased interactivity would offer a more comprehensive learning experience and a smoother transition applying the concepts learned in the LEGO® car activity to the beam and ball balance activity.

8 Conclusion and Future Work

In this paper, we proposed two robotic activities to introduce core reinforcement learning concepts to middle and high school students and encourage exploration of RL system design principles, without requiring programming experience. These activities included a LEGO® car activity and an interactive beam and ball balance robot and simulation. Through a quantitative evaluation of our pilot study, we found that our curriculum does support students understanding of RL concepts and is a stepping stone in facilitating their comprehension of RL system design. For future studies we hope to have a more interactive website for students to explore various parameter controls and their effect on agent learning performance. We also recognize that because this was a one day workshop, much of the material was condensed for students. In the future, we will separate the sessions into multiple days so students have the opportunity to deepen connections between the different activities. In addition, this study was conducted with students who volunteered to participate in the workshop. To collect more generalizable data, we plan to integrate future studies with middle and high school classrooms with a larger number of students.

Bibliography

[1] Touretzky, D., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019). Envisioning AI for K-12: What Should Every Child Know about AI?. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 9795-9799. https://doi.org/10.1609/aaai.v33i01.33019795

[2] Bilstrup, K.-E. K., Kaspersen, M. H., Sørensen, M.-L. S. K., & Petersen, M. G. (2022). Opportunities and challenges of teaching machine learning as a design material with the micro:bit. In Adjunct Proceedings of the 2022 Nordic Human-Computer Interaction Conference (NordiCHI '22 Adjunct) (Article 45, pp. 1–6). Association for Computing Machinery. https://doi.org/10.1145/3547522.3547689

[3] Dahal, M., Kresin, L., Peres, A., Pereira, E. B., & Rogers, C. (2023). International collaboration to increase access to educational robotics for students. Proceedings of the 2023 IEEE Frontiers in Education Conference (FIE), 1–5. https://doi.org/10.1109/FIE58773.2023.10343494

[4] Zhang, Z., Lavigne, K., Church, W., Sinapov, J., Rogers, C. (2023). Introducing Reinforcement Learning to K-12 Students with Robots and Augmented Reality. In Proceedings of the International Conference on Robotics in Education.

[5] Coppens, Y., Bargiacchi, E., & Nowé, A. (2019, August). Reinforcement learning 101 with a virtual reality game. In Proceedings of the 1st international workshop on education in artificial intelligence K-12.

[6] Dietz, G., King Chen, J., Beason, J., Tarrow, M., Hilliard, A., & Shapiro, R. B. (2022, June). ARtonomous: Introducing middle school students to reinforcement learning through virtual robotics. In Proceedings of the 21st Annual ACM Interaction Design and Children Conference (pp. 430-441).

[7] Martínez-Tenor, Á., Cruz-Martín, A., & Fernández-Madrigal, J. A. (2019). Teaching machine learning in robotics interactively: The case of reinforcement learning with Lego® Mindstorms. Interactive Learning Environments, 27(3), 293-306.

[8] Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction (Vol. 1, No. 1, pp. 9-11). Cambridge: MIT press.