An Interactive Robot Platform for Introducing Reinforcement Learning to K-12 Students

Ziyi Zhang, Sara Willner-Giwerc, Jivko Sinapov,
Jennifer Cross, and Chris Rogers

Tufts University, Medford MA 02155, USA,
{ziyi.zhang|sara.willner_giwerc|jivko.sinapov|
jennifer.cross|chris.rogers}@tufts.edu,

Abstract. As artificial intelligence (AI) plays a more prominent role in our everyday lives, it becomes increasingly important to introduce basic AI concepts to K-12 students. To help do this, we combined the physical (LEGO® robotics) and the virtual (web-based GUI) worlds for helping students learn some of the fundamental concepts of reinforcement learning (RL). We chose RL because it is conceptually easy to understand but received the least attention in previous research on teaching AI to K-12 students. Our initial pilot study of 6 high school students in an urban city consisted of three separate activities, run remotely on three consecutive Friday afternoons. Students' engagement and learning were measured through a qualitative assessment of students' discussions and their answers to our evaluation questions. Even with only three sessions, students were optimizing learning strategies, and understanding key RL concepts and the value of human inputs in RL training.

Keywords: reinforcement learning, robotics education, K-12 education

1 Introduction

Artificial intelligence (AI) is predicted to be a critical tool in the majority of future work and careers. It is increasingly important to introduce basic AI concepts to K-12 students to build familiarity with AI technologies that they will interact with. Reinforcement Learning (RL), a sub-field of AI, has been demonstrated to positively contribute to many fields, including control theory [1], physics [2] and chemistry [3]. This was demonstrated by the AlphaZero system which attracted widespread public attention by defeating the world's top chess and Go players [4]. For students, basic concepts of RL are intuitive and attractive to learn since it is similar with our cognition of nature of learning [5] and easy to be demonstrated with games. For instance, when researchers introduced Super Mario to undergraduate RL sessions, they argued that it increased students' engagement [6]. However, most current platforms and empirical research on introducing AI to K-12 students are based on training supervised learning models [7, 8]. There are only few activities on the Machine Learning for Kids platform, and a VR system designed by researchers that are intended to introduce RL to K-12 students

[9]. The activities in these RL teaching projects were all fully developed in the simulated world and there is no research on using physical tools like educational robots to introduce RL concepts to K-12 students in the real world.

To address this need, we designed three consecutive sessions combining the LEGO SPIKE Prime robot and a code-free web-based GUI designed for middle school and above students to learn basic RL concepts. Our first two sessions are designed around the LEGO SPIKE Prime robot. We asked students to build and control a robot car to move a specific distance, first with mathematical methods using extrapolation, and then with a RL method. In the third session, students were asked to train a virtual robot agent to finish a 1D and a 2D treasure hunt challenge on a web-based GUI. Prior research has shown that educational robots are powerful tools for introducing AI concepts to students from young age [10] to college students [11, 12]. Through our sessions, we want students to learn RL in both the physical and virtual world and give the students an intuitive, interactive, and engaging learning experience. Due to COVID-19, we designed our system to be easily built and explored at home, with the instructors providing feedback and guidance using common video conferencing platforms. Our system and proposed structured activities cover the following aspects of RL: 1) Key concepts in RL, including state, action, reward and policy; 2) Exploration and exploitation and how the agent chooses whether to explore or exploit; 3) Q-table and agent's decision making based on it; 4) Episodes and termination rules; and 5) Impacts of human input. We ran a pilot study in December 2020 to test our system design and session plan with 6 high school students in Boston area remotely and described the key results and takeaways in this paper.

2 Background and Related Work

It has become increasingly important to introduce AI concepts to pre-college level or even younger students, resulting in a number of approaches and methodologies [7, 10]. Recent AI education platforms include Google's Teachable Machine [13], Machine Learning for Kids, and MIT's Cognimate [14]. Google's Teachable Machine provides a web-based GUI for students to build classification models without any specialized technical expertise and to easily distribute designed courses and tutorials to others. Other web-based AI education platforms demonstrate AI-related concepts by implementing servers and web interfaces for students to design and test AI agents to play games such as Rok [15].

When designing an activity to teach AI to middle and high school students, we focused on how to lower the barrier to entry and keep it engaging. Game-like activities can be engaging to K-12 students and many methodologies for teaching AI have used video games as educational tools [6, 16]. Research has shown that games are a promising tool for introductory AI classes [17, 18] and has demonstrated the potential to engage non-major students [12]. These prior works indicate that students are more motivated and engaged when learning AI through games [19]. Educators also consider "toy problems" as satisfactory approaches to introduce basic AI knowledge [20]. Our activity is inspired by classic treasure hunt games which we make it tangible by combining the web GUI with an educational robotics platform.

Educational LEGO robots and others have been studied in many K-12 education contexts including physics [21], mathematics [22] and engineering [23, 24]. These studies have shown that robotics kits can improve the students' engagement and facilitate understanding of STEM concepts. LEGO robots have also been used in studies for teaching AI and robotics to different age group students [25, 26, 10–12, 27]. These researchers have reported that LEGO robots could make the learning more interactive, attractive and friendly to students who do not have an AI or robotics background, which provided inspiration for us on how to utilize the advantages of physical robots when designing our RL activity.

3 Reinforcement Learning Background

Reinforcement learning is a class of problems where an agent has to learn how to act based on scalar reward signals detected over the course of the interaction. The agent's world is represented as a Markov Decision Process (MDP), a 5-tuple $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma >$, where $\mathcal{S}$ is a discrete set of states, $\mathcal{A}$ is a set of actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$ is a transition function that maps the probability of moving to a new state given action and current state, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ gives the reward of taking an action in a given state, and $\gamma \in [0, 1)$ is the discount factor. We consider episodic tasks in which the agent starts in an initial state s_0 and upon reaching a terminal state s_{term}, a new episode begins.

At each step, the agent observes its current state, and chooses an action according to its policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The goal of an RL agent is to learn an optimal policy π^* that maximizes the long-term expected sum of discounted rewards. One way to learn the optimal policy is to learn the optimal action-value function $Q(s, a)$, which gives the expected sum of discounted rewards for taking action a in state s, and following policy π after:

$$Q^*(s, a) = \mathcal{R}(s, a) + \gamma \sum_{s'} \mathcal{T}(s'|s, a) \times max_{a'} Q^*(s', a')$$

A commonly used algorithm used to learn the optimal action-value function is Q-learning [28]. In this algorithm, the Q-function is initialized arbitrarily (e.g., all zeros). Upon performing action a in state s, observing reward R and ending up in state s', the Q-function is update using the following rule:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(R + \gamma\, max_{a'} Q(s', a') - Q(s, a))$$

where α, the learning rate, is typically a small value (e.g., 0.05). The agent decides which action to select using an ϵ-greedy policy: with small probability ϵ, the agent chooses a random action (i.e., the agent explores); otherwise, it choose the action with the highest Q-value in its current state (i.e., the agent acts greedily with respect to its current action-value function).

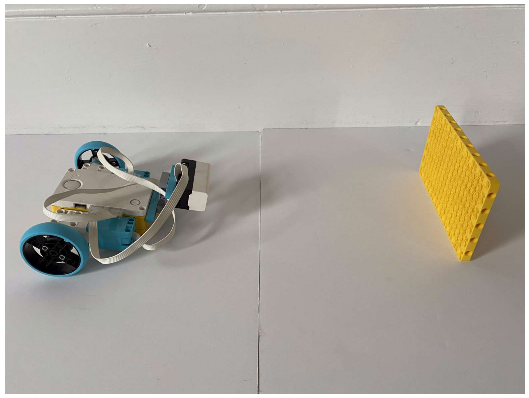

Fig. 1. An example physical build of the robot car and activity environment

RL can be computationally expensive and may require large amounts of interaction. To speed up learning, researchers have proposed human-in-the-loop RL methods. For example, in the learning-from-demonstration (LfD) framework, human teachers take over the action selection step, often providing several trajectories of complete solutions before the agent starts learning autonomously. In a related paradigm, the agent can seek "advice" from its human partner, e.g., what action to select in a particular state for which the Q-values are thought to be unreliable due to lack of experience. One of the goals of our system and proposed activity is to demonstrate to students how human partners can help a robot learn through interacting with it.

4 Technical Approach

4.1 System Overview [1]

The following system components are required for setting up the activity: a physical robot platform, a custom code-free web-based GUI environment to better visualize and control the RL training process, and instructions to guide students as they go through the activity. We used the LEGO SPIKE Prime robot, which supports MicroPython and Scratch-like block-based coding, and Bluetooth communication between the robot and a computer. It comes with several sensors including distance sensor, force sensor and color sensor. It also has an intuitive programming application for students to code and monitor the status of the robot. In our activity, students are guided through a challenge to build a robotic car like the one shown in Figure 1 and program it to move a specific distance, first using a mathematical approach, and then using a RL approach.

We also designed an interactive, web-based GUI for both the 1D and 2D treasure hunt activities. The objective of the GUI is to facilitate student learning

[1] All code and materials available at: https://github.com/ZyZhangT/rlplayground

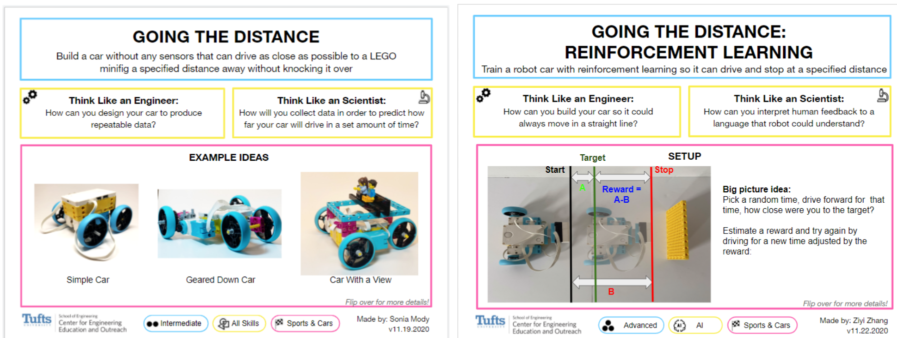

Fig. 2. The front side of the placemat instruction used in session 1 and 2. We provide students some example builds and hints to the solution of the challenges. High-resolution images are available at: https://github.com/ZyZhangT/rlplayground/tree/main/Placemats

by illustrating key concepts and processes that are hard to demonstrate with the physical robot. Also, we wanted to provide students with a friendly and efficient approach to manually control the training process of an RL agent. To serve such purpose, the GUI system has the following features: 1) An interface for students to visualize how key RL variables (e.g., state, action, reward) change during the training process; 2) Opportunities for students to participate in the training process by providing rewards to the robot to demonstrate how human input can affect training; 3) An interface for students to tweak the ϵ value to learn the difference and importance of exploration and exploitation; 4) Visual aide for teachers to explain or demonstrate RL concepts especially when implementing this activity remotely; and 5) Repeatable, automated training platform for the agent while students explore the activity independently.

Placemat instructions are one-page (double-sided) instructions that supplement the GUI. They provide students with a few images of example builds, some guidance to get them started, and activity specific background knowledge. However, they do not provide step-by-step instructions or dictate the creation of a single "correct" solution [29]. For this activity, students used two different placemat instructions showed in Figure 2. The goal of using the placemat instructions was to support students in getting started with the RL activities without telling them exactly what to do and how to do it.

4.2 Graphical User Interface for Treasure Hunting Activities

The first part of our GUI is based on an 1-D treasure hunt activity as we showed in Figure 3. The agent is located in a six-state environment, which is located at the center of the interface. Students can differentiate between various states, including a starting state, a trap (restart from the starting state) and a treasure (termination). The agent can choose its action to be either moving left or moving right. We set the rewards for the trap, the treasure and other states to be -20,

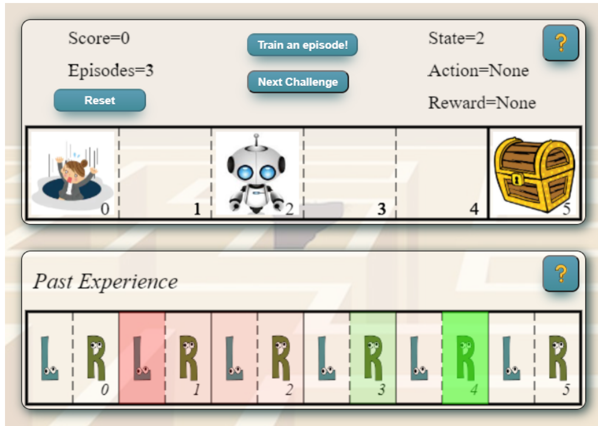

Fig. 3. GUI layout of 1-D treasure hunting. (The upper part is for students to visualize virtual robot's move in the maze and interact with the GUI. The lower part is for students to visualize agent's assessment on taking different actions in each state.

20, and -1 correspondingly. Students can let the agent train itself for an episode using Q-learning algorithm by clicking on the button above the map and inspect the learning process by watching how the agent moves around in the environment and also from the information provided in the user interface, such as cumulative return of the episode, current state, last action and reward. We also provide buttons for students to reset the training process, or to navigate to the 2-D treasure hunt challenge. To help students acquire a better understanding of the learning procedure, we provide a visual aid to illustrate the current estimated Q-value of each state-action pair, which is located under the virtual maze. For each state-action pair, if the corresponding box turns greener, it means the agent has a more positive value estimation; if the corresponding box turns redder, it has a more negative value estimation. But how each Q-value is calculated is not exposed to the student since it would be too complicated. We also have two buttons with question marks to help students with some common confusions they might have during the training process.

Through 1-D treasure hunting activity, we hope to introduce basic RL concepts such as state, action, episodes to students and help them get familiar with our interface. After that, students are guided to proceed to the 2-D treasure hunting activity.

As shown in Figure 4, the agent is now in a 2-D maze with an addition of an exit and some barriers between states to make the problem more complex. To put students' in the robot's perspective, the maze structure starts out hidden from students and as the training proceeds, the states visited by the agent and barriers detected by the agent would be revealed on the map. The maze has 16 states, including a starting state, two traps (restart from the starting state), a treasure, and an exit (termination). The rewards for actions that make the

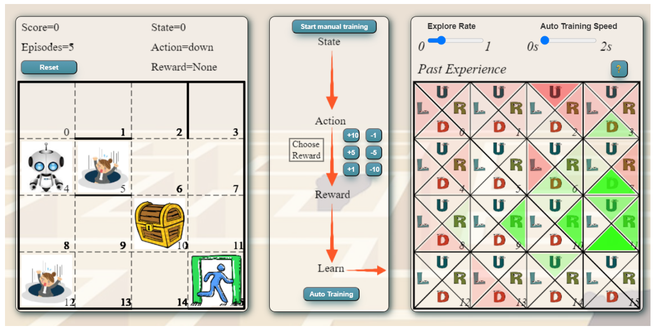

Fig. 4. GUI layout for 2-D Treasure Hunt. The general layout is the same as 1D version except we added a mid part of the GUI for students to play with different training modes. And on top of the right part of the GUI, students could tweak exploration rate and the auto training speed.

robot reach traps, treasure or exit, or hitting a barrier are -10, 20, 50, and -10 correspondingly. Otherwise, the reward is -1. And the agent now has 4 available actions: moving up, down, left, or right. For 2-D treasure hunt, we provide buttons for both automatic training and manual training, which means students could choose to let the agent train itself based on the reward table we provides, or to give the agent reward by themselves at each step to train it in their own way. if students wish to take a step-by-step look at the training process, we also provide adjustable automatic training speed so they can slow the training process. Students could also tweak the exploration rate (ϵ) during the training to see how it affects the agent's learning process. We provide the same kind of information about the training process above the maze map and visual illustration on the Q-value estimation on the right side of the interface. Through 2-D treasure hunt activity, we expect students to have a deeper comprehension on basic RL concepts, also understand exploration and exploitation, and the impact of human inputs.

4.3 Lesson Structure

The lesson was divided into three one-hour sessions. The first two sessions leveraged the LEGO SPIKE Prime robot. In the first session, students built a robot car for our "Going the Distance" challenge. Students used the placemat instruction shown in Figure 2 as a support resource. The goal of this activity was to get students to program their car to get as close as possible without running over a LEGO mini-figure placed between 5 inches and 25 inches away from their cars. They were not allowed to use a sensor for this challenge. The objective of this was to get them to think about interpolation and extrapolation and get more familiar with programming their robot using mathematical relationships. While this activity did not leverage RL or other artificial intelligence concepts, it acted as a setup to the second activity which did introduce RL.

In the second session, the students used reinforcement learning to solve the going the distance challenge. The goal of this challenge was to help students obtain a general understanding of what RL is and let them try to code a simple RL method. This session contained 4 phases. Since we assumed that students had little to no prior experience with artificial intelligence, the first phase was a 5 minutes introduction to the general concepts of AI and reinforcement learning. Phase 2 is the main part of this session in which we would send out the placemat showed in Figure 2, and students would try to implement the RL method we provided in placemat to accomplish the challenge individually. In phase 3, we asked students to show how far they got. The goal of this section was for them to share ideas with each other, and also for us to evaluate how much the students learned from this session and whether the activity design was achieving aspects of the desired outcomes. The last phase was to get feedback from students and introduce some of the concepts for the final session.

The third session was designed around our web platform. The learning objectives were to let students: 1) Understand more specific RL concepts (state, action, reward and episode); 2) Identify the differences between exploration and exploitation in the context of reinforcement learning and understand why exploration is important and how is affects the training process if an inappropriate ϵ is given; 3) Associate a specific action performed by the agent with the corresponding policy function, i.e., relate the agent's choice with values in the Q-table; and 4) Understand how humans can be part of the learning process and facilitate the agent identifying more desirable actions in a shorter amount of time.

We separated this session into 4 phases. In the first phase, students played with the 1D treasure hunting game to get familiar with the GUI and basic RL concepts. Then they started the 2D treasure hunting challenge, in which they first watched how the robot trained itself and learned about the environment. Next, they manually trained the robot by giving it reward with six options (+10, +5, +1, -1, -5, -10) for several episodes to compare the two different methods. After that, we asked students to reset the environment, set ϵ to zero, automatically train the robot for one episode, and then see how the robot kept moving to the treasure box and fails to find the exit. We let them discuss why this happens in an effort to introduce the concept of exploration and exploitation. Due to the limitation of time, we interspersed our assessment on students' learning outcome throughout the whole training process in the form of discussion questions. These questions included asking them to explain why specific phenomena happen during training and to describe their understanding of RL concepts.

5 Pilot Study

5.1 Background

We conducted a pilot study with 6 high school students in a city just outside of Boston, Massachusetts. These six students were members of a robotics club that had been meeting weekly for three months to do activities with the LEGO SPIKE Prime robotic kit. Two of the students had previous robotics and coding

Fig. 5. A screenshot from our pilot study. One student was testing his robot while others were watching or preparing their robot cars

experience, and four of them were new to both robotics and coding. The students had no prior knowledge of AI. The testing includes three separate sessions as we proposed in the former chapter. The first session was carried out on December 4^{th}, and the following two sessions were held on two consecutive Fridays after. Each session lasted for an hour, and all the sessions were held remotely on Zoom.

5.2 Robot-Centered Sessions

The first two sessions was based on the SPIKE Prime robot. Students were asked to build a robot car and figure out the relationship between running time and moving distance using a mathematical (Session 1) model and RL (Session 2) to move the robot forward by a specific distance. The students were able to build their car in 15 minutes (with the help of the placemat instruction) and came up with different ways to control their robot, rather than simply using the relationship between running time and moving distance. By the end of the session, all of the students were successful or close to being successful in completing the challenge. In Session 2, students were asked to program a simple RL method to solve the same challenge as in Session 1. We included code snippets and flowchart in the placemat instruction, but some of the students still had difficulties coding the RL method. Most of the confusion was about how to establish a reward system by code. By the end of the session, two students came up with the correct code structure of the RL method. One of them actively expressed his excitement about the content of this session. The two students who were located in the same household and therefore worked together had hardware issues and so failed to catch up and complete the activity. One student who didn't finish during the session contacted us after the session and was able to complete the challenge. Overall, we think the challenge had an appropriate difficulty level. Even though not all of the students successfully programmed the RL method by the end of the session, they understood the general methodology of RL with the help of the placemat instruction and explanations from the instructor.

The LEGO SPIKE Prime robot and the placemat instructions were two key components of the first two sessions. In our study, we found the robot not only increased student engagement, but also inspired them to be creative. Unlike a virtual platform where students have limited ways of interacting with the environment, a physical robot allows them to manipulate the robot in space and time and control not only the robot itself but also the robot's environment. For example, one student used the relationship between the rotations and driving distance to control his robot in Session 1. They also built different robot cars and were responsible for finding ways to test them. The placemat instructions for Session 1 and 2 acted as a shared artifact between the students and the instructors. In Session 1, the placemat instruction was useful for helping students get started with building their robot car. In Session 2, the placemat instruction served as a resource for the instructors to point students towards answers when they had questions about how to code using RL. The diagram on the front of the placemat instruction helped spark dialogue about what the relevant variables in the system were, and the flowchart on the back helped students combine those variables in a single program to achieve the desired outcome.

5.3 Web-based Session

In the third session, centered around the web platform, students were first asked explore the 1D treasure hunt game to get familiar with the GUI layout, and they all successfully trained the virtual robot in 10 minutes. The students pointed out that with more training episodes, the robot became smarter, and it learned to choose the shortest route to the treasure, indicating that they understood what "training" means in RL. After the 1D challenge, they explored automatic and manual training in the 2D maze. Automatic training went smoothly, but the students struggled with the manual mode. Since the ϵ-greedy policy used to choose actions was hidden from the students, it was difficult for them to understand some of the agent's random moves. However, the students also showed creativity and tried out interesting ideas, including different training strategies. One student shared that it was more effective to use small rewards at the beginning and then use larger rewards to cement right moves into the agent's policy. These types of comments showed that students started to build up their own understanding of the importance of the reward system. When asked whether they preferred automatic or manual training, a student answered with the manual option since she could control the agent's rewards and it was fun to play around with different rewards. Next, students tweaked the exploration rate (ϵ) to see how it affected training, and were asked several questions (shown in Table 1) about exploration vs. exploitation. One student proposed that we could give the robot a high exploration rate at the beginning and then decrease it, which is exactly what decaying ϵ-greedy training does in RL. Finally, we evaluated students' learning outcomes on RL concepts by asking them several questions (a sample of answers is shown in Table 1). After the session ended, one student stayed on to ask many technical questions, including how long it would take to design a platform like the one he had just used, and how hard it was to accomplish the

Table 1. Evaluation questions and answers from the students

Questions	**Answers from students (pseudonym name)**
1: Do you prefer automatic training or manual training?	A: “I enjoy the manual training, you kinda choose what it gets and doesn’t get...though it takes a while to understand what is happening, it’s still kind of fun to play around.”
2: When the exploration rate was 0, why did the robot keep moving to the treasure box?	B: “Because whenever it hits the treasure, it gets 20 points.”; C: “...Because it just always go to wherever it has the highest reward.”
3: What would happen when the exploration rate is 1?	C: “...It would completely ignore all of the stuff it had found out and just move randomly.”
4: Use one sentence to describe what you think reinforcement learning is?	B: “It takes time.” D: “I think reinforcement learning is you go through many trials...test paths you remembered from experience and optimize it to find the most efficient way...Basically reinforcement learning is learn from experience.” C: “It’s trial and error.”
5: What do you think is the most important part in RL?	C: “I think it’s having a good system for figuring out how much the reward should be...”
6: For the activity in the last session, what do you think the states and actions are? How many states do we have for that activity?	C:“...While the distance sensor seems to be able to detect anything from 0-200 centimeters, I would say there are 200 states, and the action is moving some amount of time or distance.”

RL algorithm for the treasure hunting activity. He also checked the source code behind the web GUI, which showed a high level of interest and engagement.

5.4 Limitations

Since the pilot study only had 6 participants, we didn’t collect and analyze quantitative data from our testing. For session 2, we think the time was tight and some students spent too much time programming the robot and debugging syntax errors rather than coming up with the logic of the RL algorithm. In the future, we would provide students more pre-written code chunks so they could focus on writing the algorithm. In Session 3, since students were all working on their own screens, we found it was hard to track how far they got. Going forward we would use multiple screen shares functions so that we can see what students are doing in real time, which would be especially helpful in the manual training phase. Also, we found that if one student answered a question, other students were sometimes unwilling to answer it again. Therefore, we need to find a way to encourage students to talk more about their own thinking and reduce the influence of other students’ answers so we could better evaluate their learning outcome.

5.5 Results

Overall, students demonstrated satisfactory learning outcomes on general and specific RL concepts. Though most of the time we let the students independently explore the activity we designed, they still gained an understanding of the AI concepts we were hoping they would learn about and successfully triggered their interest in the AI and RL fields. In Session 2 specifically, we expected students to learn the logic of a general RL algorithm through hands-on coding experience of solving a problem. With the exception of the two students who had hardware issues, most of the students were able to generate the correct structure of the RL method by the end of the session. In Session 3, we hoped that students would learn about specific RL concepts including reward, action, explore and exploit, etc. As shown in Table 1, students could correctly answer most of our evaluation questions. The interesting ideas shared by the students during the session revealed that some students not only had clear understanding, but even thought one step further than we expected on concepts covered in this session.

6 Conclusion and Future Work

In this paper, we integrated the LEGO SPIKE Prime robotics kit and designed a web platform to introduce Reinforcement Learning (RL) to age 13 and above students. Centered around this platform, we designed three sessions for students to learn about general RL ideas and specific RL concepts, and try to use RL to solve a real problem. We conducted testing with six high school students and the results showed that our activity and GUI design was engaging and intuitive for students to use. Through our activity, students learned about several key RL concepts and showed excitement and engagement throughout and after the sessions. In the future, we hope to further evaluate the data we acquired from the testing to improve the activity and GUI design. Moving forward, we plan to design more systematic means to measure student learning of AI topics and to scale up our platform for future testing with larger groups of students. Finally, Augmented Reality (AR) interfaces (e.g., [30, 31]) for human-robot interaction have shown promise at revealing the robot's sensory and cognitive data to users as to establish common ground between the human and the robot. We plan to introduce such AR interfaces to our activities as to enable students to visualize in context what the robot is learning and what factors it is using to make its decisions as it attempts to solve a problem, as well as to enable students to debug their robot in real time [32].

References

1. B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis. Optimal and autonomous control using reinforcement learning: A survey. *IEEE Trans. on Neural Networks and Learning Systems*, 29(6):2042–2062, 2018.
2. Thomas Fosel, P. Tighineanu, T. Weiss, and F. Marquardt. Reinforcement learning with neural networks for quantum feedback. *Physical Review X*, 8:031084, 2018.

3. Zhenpeng Zhou, Xiaocheng Li, and Richard Zare. Optimizing chemical reactions with deep reinforcement learning. *ACS Central Science*, 3, 12 2017.
4. David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
5. Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* A Bradford Book, Cambridge, MA, USA, 2018.
6. Matthew E Taylor. Teaching reinforcement learning with mario: An argument and case study. In *Proceedings of the 2011 AAAI Symposium Educational Advances in Artificial Intelligence*, 2011.
7. Henriikka Vartiainen, Matti Tedre, and Teemu Valtonen. Learning machine learning with very young children: Who is teaching whom? *International Journal of Child-Computer Interaction*, page 100182, 06 2020.
8. Bawornsak Sakulkueakulsuk, Siyada Witoon, Potiwat Ngarmkajornwiwat, Pornpen Pataranutaporn, Werasak Surareungchai, Pat Pataranutaporn, and Pakpoom Subsoontorn. Kids making AI: Integrating machine learning, gamification, and social context in stem education. In *2018 IEEE Intl. Conf. on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 1005–1010. IEEE, 2018.
9. Youri Coppens, Eugenio Bargiacchi, and Ann Nowé. Reinforcement learning 101 with a virtual reality game. In *Proceedings of the 1st International Workshop on Education in Artificial Intelligence K-12 (August 2019).*
10. Randi Williams, Hae Won Park, and Cynthia Breazeal. A is for artificial intelligence: The impact of artificial intelligence activities on young children's perceptions of robots. In *CHI '19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 04 2019.
11. Simon Parsons, Elizabeth Sklar, et al. Teaching AI using LEGO mindstorms. In *AAAI Spring Symposium*, 2004.
12. Bram van der Vlist, Rick van de Westelaken, Christoph Bartneck, Jun Hu, Rene Ahn, Emilia Barakova, Frank Delbressine, and Loe Feijs. Teaching machine learning to design students. In Zhigeng Pan, Xiaopeng Zhang, Abdennour El Rhalibi, Woontack Woo, and Yi Li, editors, *Technologies for E-Learning and Digital Entertainment*, pages 206–217, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
13. Michelle Carney, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. Teachable machine: Approachable web-based tool for exploring machine learning classification. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–8, New York, NY, USA, 2020. Association for Computing Machinery.
14. Stefania Druga. *Growing up with AI. Cognimates: from coding to teaching machines.* PhD thesis, Massachusetts Institute of Technology, 2018.
15. Stefan Friese and Kristian Rother. Teaching artificial intelligence using a web-based game server. In *Proceedings of the 13th Koli Calling international conference on computing education research*, pages 193–194, 2013.
16. Du-Mim Yoon and Kyung-Joong Kim. Challenges and opportunities in game artificial intelligence education using angry birds. *IEEE Access*, 3:793–804, 2015.
17. A. McGovern, Zachery Tidwell, and Derek Rushing. Teaching introductory artificial intelligence through java-based games. In *AAAI 2011*, 2011.

18. John DeNero and Dan Klein. Teaching introductory artificial intelligence with pac-man. In *First AAAI Symposium on Educational Advances in Artificial Intelligence*, 2010.
19. Thomas Hainey, Thomas M Connolly, Elizabeth A Boyle, Amanda Wilson, and Aisya Razak. A systematic literature review of games-based learning empirical evidence in primary education. *Computers & Education*, 102:202–223, 2016.
20. Michael Wollowski, Robert Selkowitz, Laura E Brown, Ashok Goel, George Luger, Jim Marshall, Andrew Neel, Todd Neller, and Peter Norvig. A survey of current practice and teaching of AI. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
21. Pavel Petrovič. Spike up prime interest in physics. In *Robotics in Education*, pages 146–160. Springer International Publishing, 2021.
22. Sonia Mandin, Marina De Simone, and Sophie Soury-Lavergne. Robot moves as tangible feedback in a mathematical game at primary school. In *Robotics in Education*, Advances in Intelligent Systems and Computing, pages 245–257. Springer International Publishing, Cham, 2016.
23. Jeffrey Laut, Vikram Kapila, and Magued Iskander. Exposing middle school students to robotics and engineering through LEGO and matlab. In *120th ASEE Annual Conference and Exposition*, 2013. 120th ASEE Annual Conference and Exposition ; Conference date: 23-06-2013 Through 26-06-2013.
24. K. Williams, I. Igel, R. Poveda, Vikram Kapila, and Magued Iskander. Enriching K-12 science and mathematics education using LEGOs. *Advances in Engineering Education*, 3, 01 2012.
25. Lawrence Whitman and Tonya Witherspoon. Using LEGOs to interest high school students and improve K-12 stem education. *Change*, 2, 12 2003.
26. Robert I Selkowitz and Debra T Burhans. Shallow blue: LEGO-based embodied AI as a platform for cross-curricular project based learning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 3037–3043, 2014.
27. Mark Cheli, Jivko Sinapov, Ethan E Danahy, and Chris Rogers. Towards an augmented reality framework for K-12 robotics education. In *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 2018.
28. Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.
29. Sara Willner-Giwerc, Ethan Danahy, and Chris Rogers. Placemat instructions for open-ended robotics challenges. In *Robotics in Education*, pages 234–244. Springer International Publishing, 2021.
30. Andre Cleaver, Faizan Muhammad, Amel Hassan, Elaine Short, and Jivko Sinapov. SENSAR: A visual tool for intelligent robots for collaborative human-robot interaction. In *AAAI Fall Symposium on "Artificial Intelligence for Human-Robot Interaction"*, 2020.
31. Kishan Chandan, Vidisha Kudalkar, Xiang Li, and Shiqi Zhang. ARROCH: Augmented reality for robots collaborating with a human. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
32. Bryce Ikeda and Daniel Szafir. An AR debugging tool for robotics programmers. In *International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interaction (VAM-HRI)*, 2021.