# ARtonomous: Introducing Middle School Students to Reinforcement Learning Through Virtual Robotics

Griffin Dietz
Stanford University
Stanford, CA, USA
dietz@cs.stanford.edu

Jennifer King Chen
Apple
Cupertino, CA, USA
j_kingchen@apple.com

Jazbo Beason
Apple
Cupertino, CA, USA
jontait_beason@apple.com

Matthew Tarrow
Communications by Design
Grand Rapids, MI, USA

Adriana Hilliard
Apple
Cupertino, CA, USA
adri@apple.com

R. Benjamin Shapiro
Apple
Cupertino, CA, USA
nerd@apple.com

## ABSTRACT

Typical educational robotics approaches rely on imperative programming for robot navigation. However, with the increasing presence of AI in everyday life, these approaches miss an opportunity to introduce machine learning (ML) techniques grounded in an authentic and engaging learning context. Furthermore, the needs for costly specialized equipment and ample physical space are barriers that limit access to robotics experiences for all learners. We propose ARtonomous, a relatively low-cost, virtual alternative to physical, programming-only robotics kits. With ARtonomous, students employ reinforcement learning (RL) alongside code to train and customize virtual autonomous robotic vehicles. Through a study evaluating ARtonomous, we found that middle-school students developed an understanding of RL, reported high levels of engagement, and demonstrated curiosity for learning more about ML. This research demonstrates the feasibility of an approach like ARtonomous for 1) eliminating barriers to robotics education and 2) promoting student learning and interest in RL and ML.

## KEYWORDS

reinforcement learning, education, AI, middle school, robotics

**ACM Reference Format:**
Griffin Dietz, Jennifer King Chen, Jazbo Beason, Matthew Tarrow, Adriana Hilliard, and R. Benjamin Shapiro. 2022. ARtonomous: Introducing Middle School Students to Reinforcement Learning Through Virtual Robotics. In *IDC '22: Interaction Design and Children, June 27–June 30, 2022, Braga, Portugal.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3501712.3529736

## 1 INTRODUCTION

Middle school students regularly use artificial intelligence (AI) and machine learning (ML) technologies through applications such as virtual assistants, social media algorithms, and game-playing AIs [5, 34]. Even so, engagement with these everyday technologies does not necessarily translate to confirmed understanding—or even awareness—of AI and ML [22, 23]. Furthermore, the complexity of the technology can make it intimidating to learn and difficult to reason about [6]. This public knowledge gap has led researchers and educators to begin advocating for and studying AI and ML education for young learners [36, 52]. In addition to supporting the development of contemporary computational literacy, AI, ML, and (closely related) data science education can foster critical engagement with data and complement existing programming-centric computing education tools and curriculum [32].

*IDC '22, June 27–June 30, 2022, Braga, Portugal*
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9197-9/22/06.
https://doi.org/10.1145/3501712.3529736

Robotics is currently a common entry point to computing. With over 679,000 students in 110 countries participating in FIRST robotics programs in 2019–2020 [27], some U.S. school districts (e.g., in the St. Vrain Valley of Colorado) adopting robotics programs district-wide [31], and numerous robotics kits on the market (e.g., Sphero [48], LEGO Mindstorms [51], and Cue Robot [57]), robotics has a proven track record for engaging students in computing. However, although robotics is widely used in STEM education, the cost of specialized hardware and kits—often on the order of thousands of dollars for a classroom set—can inhibit broader reach [38]. Additionally, learning about robotics with today's educational robotics products typically requires purchasing ordinary computers or tablets in addition to this specialist hardware. This combination can make robotics too expensive for economically-disadvantaged communities, exacerbating inequities that already exist in computing education [38].

Further, while the state-of-the-art in professional robotics makes heavy use of AI and ML (a subset of AI that learns or improves based on an algorithm), current practice in K–12 robotics education lags behind this standard, reflecting an era that predates widespread AI. Integrating ML into robotics programming can enable learners to create closed-loop, robust systems that are capable of autonomously navigating new environments without tedious manual changes.

Robotics also presents a domain that is a natural fit for exploring reinforcement learning (RL), a type of ML in which agents learn from interactions with their environment that is often leveraged in robotics contexts (see Section 2.3 for a detailed explanation) [45, 50]. Critically, this type of ML has been less studied in the AI education space than other ML approaches (e.g., supervised learning), but we conjecture that it is comparatively easy to understand. Incorporating RL into educational robotics would contribute to efforts in AI education to engage learners in rich (i.e., flexible and adaptive) and authentic (i.e., reflective of professional robotics) computational learning experiences while contributing to our understanding of K–12 education surrounding less-studied types of ML.

In our work, we leverage RL to create a robotics education experience for middle school students (ages 11–14) that increases student awareness of AI/ML, addresses barriers to wider implementation, and tests our conjecture that young learners can develop a conceptual understanding of this type of ML. Specifically, we present ARtonomous, a youth-friendly, tablet-based tool for learning about and generating reinforcement learning models for robot navigation. This system supports use with purely virtual robots, which can foster users' engagement with robotics without the startup overhead of purchasing robotics kits. Further, the autonomous navigation models are closed-loop, flexible, and abstracted, reflecting the data-driven approaches of cutting-edge robotics. These models can be used in our app in concert with user-programmable event callbacks that interface with the robot, allowing the user to customize robot behavior at specific navigational waypoints in the program. With this approach, we introduce young people to additional technical understanding and competencies for our increasingly technological world while eliminating the specialized hardware costs of physical robots.

In this paper we present the following main contributions:

(1) ARtonomous, a low cost, high engagement robotics education tool that helps learners explore reinforcement learning and bridges the gap between educational and professional robotics
(2) A user study evaluating how such a tool can help students develop an understanding of reinforcement learning, positively engage with the content, and build a curiosity toward machine learning more broadly

Overall, we show that accessible and approachable reinforcement learning can bridge the gap in data-driven approaches between educational and state-of-the-art robotics. Furthermore, implementing such a system on general purpose hardware using virtual robots can reduce the cost of participation while still yielding high engagement. Our evaluation demonstrates that ARtonomous effectively introduces reinforcement learning concepts, supports student engagement, and inspires interest among middle school students in learning more about reinforcement learning and machine learning. Finally, we speculate about how the projection of virtual robots into users' physical environments using augmented reality (AR) could further enrich the impact of tools like ARtonomous.

## 2 RELATED WORK

Over the past several decades, there have been countless physical robotics tools ranging from modular kits to pre-constructed robots geared at introducing middle-school students to computing via a hands-on, project-based approach [7, 10]. While these tools are successful in introducing users to programming, they do present cost challenges and very few offer access to the autonomy that is increasingly present in professional robotics technologies. Here we describe the potential for virtual robots in addressing these cost challenges, opportunities for machine learning in robotics education, and existing efforts and open avenues of inquiry in middle school machine learning education.

### 2.1 Addressing Cost Constraints with Virtual Robotics

With a typical robotics kit costing in the hundreds of dollars and a classroom set in the thousands (see Table 1), the cost of robotics tools can present a barrier to entry for educational institutions [14], especially for learners of lower socio-economic status. These robotics tools also usually need to be accompanied by tablets or computers used to write the code to control the bots and enough dedicated physical space in which to operate them [25], further increasing barriers to implementing robotics tools into school curriculum.

Prior work demonstrates that educational robotics tools need not necessarily be expensive. Some research has shown how very motivated learners with expert teachers can re-purpose surplus technology to support robotics education [9]. Further, lower priced, maker-oriented robotics tools can also provide entry points to robotics (e.g., Sparkfun's micro:bot system), though with fewer capabilities than more expensive products. Nonetheless, the high cost of common educational robotics products leads us to wonder if new robotics education technologies that heavily leverage the capabilities of general purpose computing devices [4] can lower economic barriers to entry while cultivating interest and knowledge development that could scaffold [43] deepened participation over time.

Virtual approaches for educational robotics remove the need for costly peripherals on top of the required computer or tablet, thereby making robotics more accessible while still successfully engaging students with key ideas in STEM [8, 21, 24, 56]. AR functionality has the potential to take this engagement and learning even further [13, 16], yet AR-robot integration to date focuses on entertainment [41], communication [53, 54], or debugging features for physical robots [2, 12], rather than on hardware alternatives for education. We therefore aim to lower the cost barrier to educational robotics while still encouraging high engagement by incorporating virtual robots driven by general purpose computing devices.

### 2.2 Autonomy in Robotics Education

Typical robotics kits utilize motor-powered wheels for movement (e.g., mBot [37] and LEGO Mindstorms Robot Inventor [51]). These systems allow users to specify specific amounts of time or distances to move, but using these products on different surfaces (e.g., carpet or tile) can lead to substantially different distances actually traveled due to varying degrees of friction. Users can address these inconsistencies by modifying their code for use in different environments, but this approach allows for only short-term fixes and remains inflexible to changes in route.

Critically, this environmental rigidity is inconsistent with state-of-the-art robotic systems that automatically adapt to changing environments, presenting an opportunity for the introduction of machine learning techniques. To date, navigation of unknown settings in educational robotics relies on information gathered from sensors, but few educational robotics tools enable learners to create autonomous navigation control systems. The DJI RobotMaster S1 uses an on-board camera to acquire location information (e.g., distance from a potential barrier) and will continue to move unless within a specified distance of an obstacle. The S1 can then

| | Cost per Unit | Cost for Classroom Set (12) | Cost for 5 Sets |
|---|---|---|---|
| **Makewonder Dash** | $149.00 | $1,788.00 | $8,940.00 |
| **Sphero Bolt** | $149.00 | $1,788.00 | $8,940.00 |
| **Sphero RVR** | $249.00 | $2,988.00 | $14,940.00 |
| **LEGO Spike Prime** | $359.95 | $4,319.40 | $21,597.00 |
| **LEGO Mindstorms Robot Inventor** | $359.99 | $4,319.88 | $21,599.40 |
| **DJI RoboMaster S1** | $549.00 | $6,588.00 | $32,940.00 |

**Table 1: The price for individual and classroom sets of popular educational robotics kits in U.S. dollars. 12 units per classroom was selected based of the average middle school class size (24.9) in the United States for teachers in departmentalized instruction [26] and the assumption of students working in pairs. This would be a bare minimum and require disassembly between class periods. The last column shows cost for a teacher with five classes who does not want to lose time to disassembly.**

rotate, attempting to locate a safe direction to head, although this decision-making is still contingent on user programming [19].

We envision new educational tools that empower learners to create even more autonomous systems, with projects that get from point A to point B without relying on fine-grained, route-specific programming of the steps in between. Robotics tools that enable novices to build autonomous navigation systems with ML can increase the resiliency and authenticity of what they can create, support AI education (e.g., about RL), and offer rich contexts to integrate ML with programming (such as to respond dynamically to events that arise during a robot's autonomous navigation).

## 2.3 Middle School Machine Learning Education

In recent years, researchers have begun to explore introducing K–12 students to machine learning topics and to develop curricular objectives to this end [36, 52]. From these K–12 AI literacy objectives, we can identify the importance of supporting learners in building a mental model of the algorithm's inputs, representations, and decision-making processes and understanding what learning means for a machine, including the role of data and of the human [36, 52].

Existing machine learning education tools for K–12 tend to focus on the use [40, 42] or construction [1, 20, 33, 61, 62] of classifiers. In cases where students create their own classifiers, they typically do so via supervised learning, and once created, these classifiers can be used within code in common block-based programming environments [1, 20, 33, 61] or operate independently from programming tools altogether as standalone tools for gesture recognition [62]. While using these systems, some learners enacted engineering practices that distinctly encapsulated modeling and programming practices: separately writing code and training models, then independently testing each type of product, before integrating their models and code within their Scratch games [61]. Those results inform a key aim of the present system and study: creating new tools that can help students learn to recognize and apply the complementary capabilities of ML and programming within robotics systems.

Few tools for K–12 AI education exist that go beyond using or creating supervised learning classifiers, although those that do explore the teaching of k-means clustering, a form of unsupervised learning [55] or generative adversarial networks (GANs) [3].

Reinforcement learning (RL) is another genre of machine learning in which computational agents learn from interactions with their environment [50]. Like in supervised learning, RL agents receive feedback as they learn. However, the *kind* of feedback they receive differs. In supervised learning, ML algorithms are provided with correct information about what they should have predicted, typically coming from a human-annotated dataset. In RL, the learning algorithm still receives feedback, but rather than gaining information about what the system's correct action (i.e., prediction) should have been, the system receives only a positive or negative *reward* that it uses to inform future decisions.

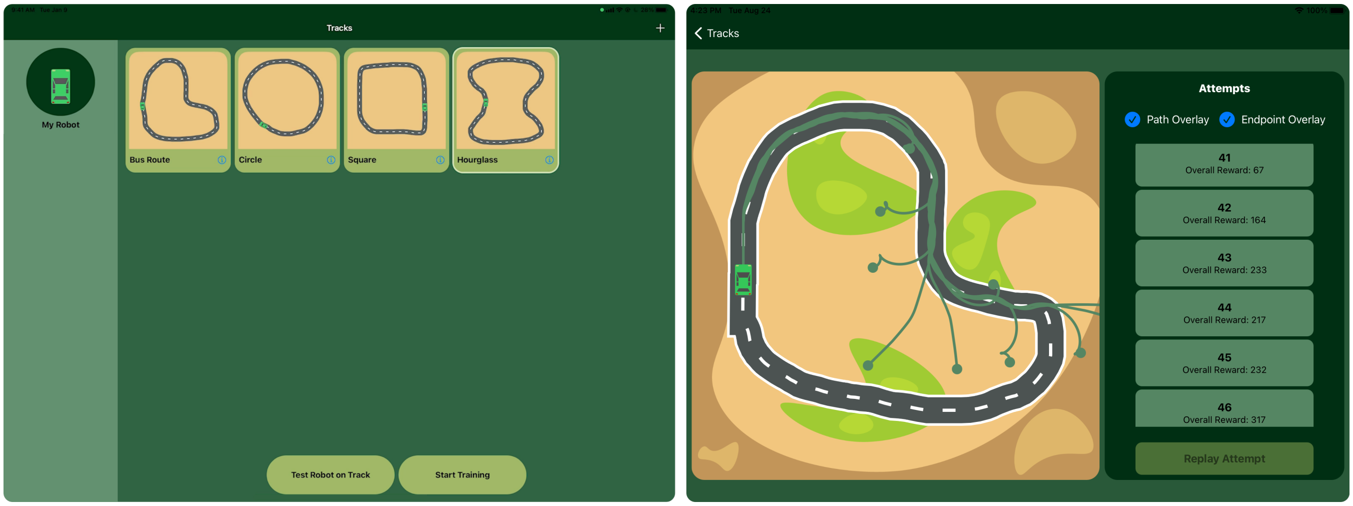
Critically, RL is frequently used in research on autonomous systems, including in robotics [45], self-driving cars [46], and game-playing AIs [47]. At every step, these agents choose an action to take from the action space given the current input, receive some reward or penalty from their environment based on that action, and learn how to take better actions in the future based on the reward/penalty that they received [50].

Despite the importance of robotics to middle school computing education and the role of RL in state-of-the-art robotics, little research to date has described tools and methods for introducing RL to middle school students, in the robotics context or otherwise, although some such systems exist for adult users [35, 39]. Notably, Zhang et al. recently published an RL robotics system that shares in our goal of helping students better understand and train robots [59, 60]. This work includes a pilot assessment of student engagement and learning that demonstrates satisfactory learning outcomes [60]. However, unlike our platform, this tool focuses on training physical robots (LEGO Spike Prime; see Table 1 for cost) and targets high school students as users [60].

We view the training of robotic control systems and the integration of those RL models with code as essential practices for future robotics education. Therefore, building on prior work, we add RL to middle school students' robotics construction toolbox in a manner that develops students' understanding of RL, inspires engagement, and motivates curiosity to learn more.

# 3 SYSTEM DESIGN & IMPLEMENTATION

We developed ARtonomous, an iPad application to bring RL into middle school robotics. This two-part learning tool allows students to first train RL models for autonomous virtual robot navigation and then write imperative code that interfaces with these models to handle specific events.



41
Overall Reward: 67

42
Overall Reward: 164

43
Overall Reward: 233

44
Overall Reward: 217

45
Overall Reward: 232

46
Overall Reward: 317

**Figure 1: *Left:* From the track dashboard, students can create new tracks, test on an existing track, train on an existing track, or select a track to view training details. *Right:* On the training detail page users can view overlays showing the paths and endpoints of each episode on the selected track, along with a list of episodes and their overall reward. Users can select individual episodes to replay.**

## 3.1 Design and Learning Objectives

Based on prior work and our aforementioned research objectives, we aimed to create a low-cost, high engagement robotics experience that helps learners explore AI/ML. Our system design therefore needed to 1) leverage virtual robots to run entirely on a general-purpose device (i.e., an iPad) and eliminate the cost of peripherals and 2) support the creation of RL autonomous navigation models to introduce key ideas in machine learning. Additionally, we aimed to make it easy to get started with this autonomous navigation approach by creating introductory teaching materials that we used within this study to orient participants.

Similarly, based on prior ML education systems research [20, 62] and curricular frameworks [36, 52], we aimed to support learning objectives pertaining to developing understanding of and curiosity for RL. Specifically, students should:

- develop a fundamental understanding of RL (e.g., understanding episodes, rewards, and penalties) while also engaging with coding fundamentals
- use a combination of quantitative performance metrics and qualitative observations to find problems with individual models and generate ideas for how to fix them
- be able to identify which problems in robotics might be solved with code and which might be solved with RL
- use their experiences to develop a broader understanding or curiosity around RL and ML systems outside robotics

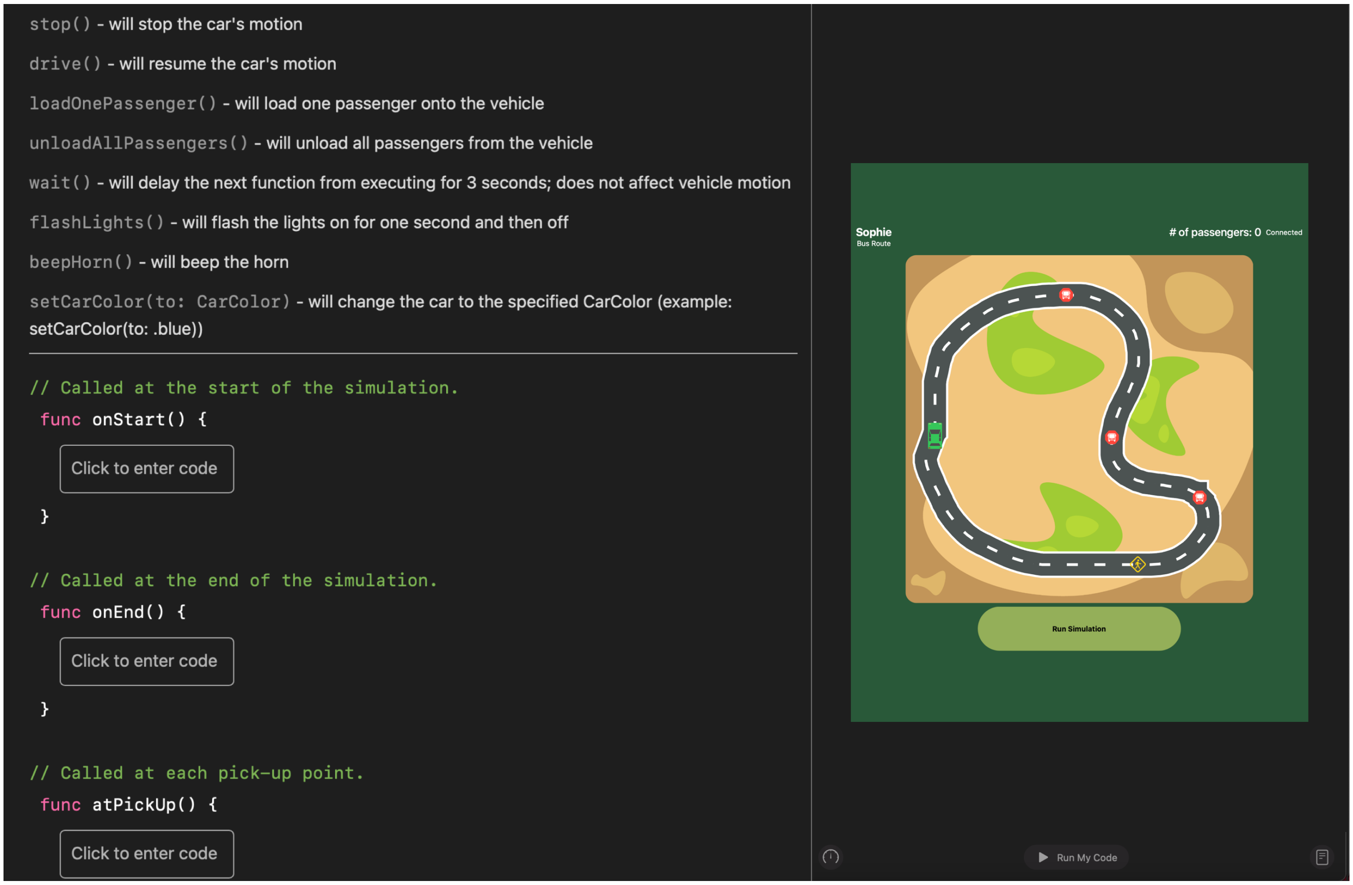
*3.1.1 From Vision to Prototype.* In our present work, we created a system to investigate the feasibility, usability, and excitement generated by the core ideas of these design and learning objectives: individual students use a tablet to trace a route for a robot, iteratively train an RL model for a virtual robot to navigate that route, and then write code to control how the virtual robot behaves at waypoints along its route. The complete system has two components, the ARtonomous model training application and the ARtonomous programming tool.

## 3.2 ARtonomous Model Training Application

With the ARtonomous model training application, users train RL models for virtual autonomous vehicles through an iPad-based interface that connects to a remote server for model training.

*3.2.1 iPad Model Creation.* We developed the iPad application in Swift for iOS 14. In the app, students begin by creating a robot (i.e., a model) and giving it a name (e.g., School Bus). From the main track dashboard, students can draw tracks on which they can both test their current robot model and train their robot model for a specified number of episodes (see Figure 1, Left). Using those tracks, students engage in a process of iterative model improvement by training their robot model on a selected course, testing that robot on the target course, identifying points for improvement based on that testing, and training again on another track to address those key points. Users can also select an individual track to replay simulations of past training episodes, view visualizations of training progress over time (see Figure 1, Right)—including on-course overlays of the robot's path in prior episodes and endpoints for every episode—or select an episode to see overlay visualizations just for that run. We included these visualizations to aid the development of mental models of the algorithm's decision making processes (e.g., state, action, and reward), and we envisioned the iterative model development process would highlight the impact and role of humans in model training [36, 52].

*3.2.2 Reinforcement Learning Environment and Model.* Whenever the student elects to train the robot model, the iPad application communicates with a model training server. This server sets the training environment to the track the student selected and runs a reinforcement learning algorithm to update the model, sending



`stop()` - will stop the car's motion

`drive()` - will resume the car's motion

`loadOnePassenger()` - will load one passenger onto the vehicle

`unloadAllPassengers()` - will unload all passengers from the vehicle

`wait()` - will delay the next function from executing for 3 seconds; does not affect vehicle motion

`flashLights()` - will flash the lights on for one second and then off

`beepHorn()` - will beep the horn

`setCarColor(to: CarColor)` - will change the car to the specified CarColor (example: setCarColor(to: .blue))

```
// Called at the start of the simulation.
 func onStart() {
     Click to enter code
 }

// Called at the end of the simulation.
 func onEnd() {
     Click to enter code
 }

// Called at each pick-up point.
 func atPickUp() {
     Click to enter code
```

**Figure 2: The ARtonomous programming tool’s joint model-code interface allows participants to view their objective and function library, write code, and run simulations that call that code all in one place.**

training information (e.g., path of the robot and action/reward for each time-step) back to the iPad as episodes are completed.

We created the RL capabilities of the system using an adaptation of the OpenAI car racing environment [11], with modifications to decrease training time and improve comprehensibility [58]. This modified environment has a discrete action space with five possible actions (accelerate, brake, left, right, and no change), added functionality for custom tracks, and a new reward function. In addition, we included a simplified scalar grayscale observation space and an expansion to that observation space to include multiple frames (to see velocity and acceleration, not just position), although these changes are purely to speed up model training and are not visible in the user-facing rendering of training or simulations. Model training used the PPO2 algorithm from StableBaselines with a CNN policy, modified to include an episode termination callback that sent episode information via the server back to the iPad application [30].

## 3.3 ARtonomous Programming Tool

We envision a future of machine learning integrated into robotics education, creating an experience where student activity is a synthesis of programming and machine learning model creation. Therefore, while students use machine learning to train models that drive the robot, they can also practice their coding skills to modify other parts of the robot’s behavior. At any point in the model training process, users can switch to a programming interface to write event-driven code for their bot. This ability to switch back and forth between model creation and programming captures the iterative modeling and programming engineering concepts and practices we aimed to foster, including recognizing and applying the complementary capabilities of ML and programming within robotics systems.

The ARtonomous programming tool was built for Swift Playgrounds, a development environment for iPadOS and macOS that supports custom tutorials. For example, Lego and Sphero offer Playground “books” that allow users to program educational robots like the Lego Mindstorms EV3 and Sphero. We similarly built on top of Swift Playgrounds, developing a model-code interface and function library for ARtonomous using Playgrounds’ built-in code compiler and code editor.

This ARtonomous programming tool automatically loads the most up-to-date version of the user’s model (retrieving it from the server), allows them to select a course to run their model on, and provides an interface for writing code to run in tandem with the model. The models integrate with Swift code via two types of callbacks: built-in callback functions and user-defined waypoints. Built-in callbacks are functions we have included in the Playground Book (`onStart()`, `onStep()`, and `onEnd()`) that are called at their respective times in model simulation execution. These functions allow students to set up or clean up their robot environment (e.g., set the color of the car at the start of the run) or run repeating or time-specific commands (e.g., flash lights on and off at every step). User-defined waypoints, on the other hand, allow users to specify specific locations on the course and then run a callback whenever their agent reaches one of these waypoints. Finally, when writing callback functions, users have access to a custom function library we have created for these virtual robots. Using this library, students can change the car's color, beep the horn, flash its lights, load or unload passengers, pause and resume driving, and more.

# 4 EVALUATION

We conducted a study to assess how well ARtonomous achieves our learning and design objectives. Since this research occurred during the COVID-19 pandemic, we conducted testing remotely to protect the health of our participants. Therefore, using a virtual robotics experience, we sought to answer the following research questions pertaining to students' developing understanding of RL, engagement with the system, and curiosity about machine learning more broadly:

1. **RQ1-Developing Understanding** Does this virtual robotics experience help students develop an understanding of reinforcement learning?
2. **RQ2-Developing Understanding** How do students using this system learn to identify problems with individual models and generate ideas for how to fix them?
3. **RQ3-Developing Understanding** Do the joint modeling and coding capabilities build intuition around which problems in robotics might be solved with code and which might be solved with reinforcement learning?
4. **RQ4-Engagement** How engaging do students find the experience of using ARtonomous?
5. **RQ5-Curiosity** After using ARtonomous, do students want to learn more about machine learning or reinforcement learning?

## 4.1 Participants

Two researchers conducted this study remotely via video conferencing software with 15 participants aged 11–14 ($M = 12.73, SD = 1.03$; 6 female, 9 male). The researchers followed the same script and alternated roles of facilitator and note-taker between participants. Each session lasted 90 minutes. Participants were recruited from a school for students with reading-related learning differences ($N = 1$), a hardware and robotics camp in Colorado ($N = 2$), a lower-income middle school in Southern California ($N = 5$), and via an email list to employees at a large tech company ($N = 7$). We chose these sites to recruit participants with a range of geographic, economic, and personal backgrounds, while also deliberately recruiting from schools/camps with robotics programs or mailing lists for families involved in robotics. To be more inclusive of participants who did not have iPad ($N = 5$) and/or Mac ($N = 10$) hardware at home, while also socially distancing due to the COVID-19 pandemic, we used screensharing and remote control features to provide virtual access to the hardware as needed. That is, we shared an iPad simulator or a Mac screen with participants who did not have those devices at home, and gave them remote control over our mouse and keyboard to interact with the interfaces directly. In this way, we were able to remotely run this study for participants on a computer running any operating system, regardless of whether they owned an iPad, without compromising the experimental procedure.

All participants had prior experience with programming and exposure to hardware/robotics (e.g., via FIRST Lego League or in-school robotics classes). When asked what they thought machine learning was at the start of the study, 40.0% talked about a machine getting better on its own, but only 33.3% discussed the use of experiences or data to drive this improvement, and just one participant (6.7%) mentioned the machine's ability to approach novel situations. That is, the majority of participants did not know what machine learning was, and those that did had a limited understanding.

## 4.2 Objective

In the study, we gave participants an objective in which they needed to get a School Bus robot to navigate its route (a prescribed course called Bus Route), stop at three bus stops to pick up passengers, and then stop again at the school to drop all of those passengers off. Therefore, using RL, participants trained a virtual robot to navigate around the bus route course. Then, in the programming interface, they handled events to turn the car yellow at the start (`onStart()`); stop, flash lights, and pick up a passenger at each bus stop (`atPickup()`); and stop, flash lights, wait, and unload all passengers at the school (`atDropoff()`).

## 4.3 Procedure

Every participant completed a four-step study procedure, which included phases for introduction, RL model training, programming, and reflection.

### 4.3.1 *Introduction.*

During the introduction phase, we first walked participants through the app setup process. For those who owned iPads, we guided them through installing ARtonomous's RL training application via iOS Ad-Hoc distribution (an over-the-air install triggered by scanning a QR code shared via video conferencing screenshare) and, for those who owned a Mac, we guided them through installing Swift Playgrounds and downloading our Playground Book onto their machines. Although the Playground book works on an iPad as well, we chose to have participants use the Playground Book on a computer to ease typing out code and standardize the experience for all participants.

After ensuring the participants could use both of the applications, whether locally or via screenshare, we collected information through verbal interviewing on their programming backgrounds and existing understanding of machine learning. We then showed a 5 minute introductory animation to explain the basics of RL, first via a dog training example followed by an explanation of how these concepts might apply to robot navigation. Finally, we introduced participants to the school bus objective.

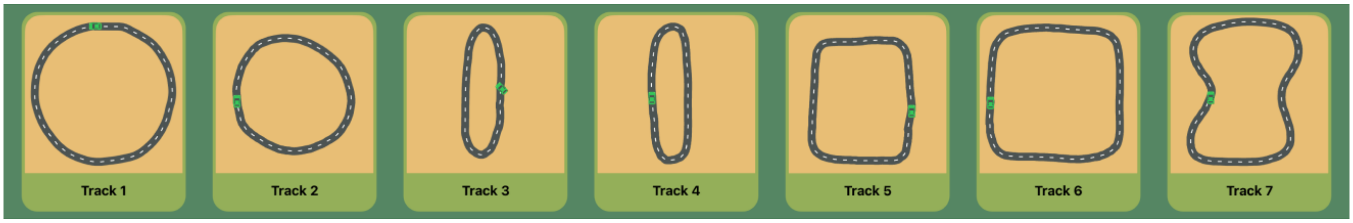


**Figure 3: The seven rapid training tracks a participant could choose from during the study. Visually similar tracks (e.g., Track 1 and Track 2) go in opposite directions.**

*4.3.2 Reinforcement Learning Model Training.* Upon opening the RLbot model training app, the experimenter briefly described the starting state of the robot and the constraints for the study. Specifically, the starting robot was partially pretrained but still not very good at navigating, and the participant was permitted to test their robot (i.e., run a simulation with the current model without training it) as many times as they wanted, but could only train the robot three times.

The experimenter then provided the participant a series of tasks to complete, with necessary prompting along the way to keep the participant on track:

(1) Recreate the bus route track from the one shown in the objective (see right side of Figure 2)
(2) Test your robot on the bus route track and identify any problems it may still have
(3) Create a training track to help your robot gain experience to address that problem
(4) Train your robot on that track
(5) Describe what your robot learned (or didn't learn) based on that training

Participants repeated steps 2–5 three times, completing this process once for each of their training opportunities. They were permitted to ask questions at any time.

Critically, to address the long (i.e., multi-hour) duration of RL model training in conjunction with the constrained timespan of the study, when participants initiated training at step 4, they were prompted to select a track from a list of seven existing "rapid training" tracks (see Figure 3) and asked to explain the reasoning behind their decision. The first six of these tracks were selected because they each target a specific kind of turn the participant might want to direct training toward, whereas Track 7 serves as a more general-purpose training track. We told participants we could train very quickly on these seven tracks and would train just the last ten episodes on the track they had created. In reality, we had pre-trained models for all permutations of these seven tracks; we would load up the appropriate model and run 10 simulations to display as training replays using this next model. In this way, participants could see the progression of training a model even within the time of the user study session and we could gain a deeper understanding of what specific features participants were hoping to train their models on based on their rapid training track choice and subsequent discussion about that choice. Of course, in practice we imagine a user could switch back and forth between model development and coding while training occurred, or set training to run overnight or between class periods.

*4.3.3 Event Handling Code.* After three rounds of training, participants moved over to the ARtonomous programming interface to write event handling code. Here we loaded up the target Bus Route track and its requisite bus stops, reminded participants of their objective, and pointed out the function library before allowing them to jump into programming. The experimenter was able to answer any questions the participant asked and could prompt them to re-read the objective or sections thereof, but did not otherwise assist with the coding.

*4.3.4 Reflection and Feedback.* Participants concluded the study with about 10 minutes of reflection and feedback, described in detail in section 4.4.

## 4.4 Metrics and Collected Data

Through both qualitative description and quantitative metrics, we evaluated participants' developing understanding of RL, their engagement with the ARtonomous experience, and their curiosity to learn more. We collected saved track and model data for all participants, recorded video and audio of the study sessions, and transcribed answers to the questions asked in the introduction and reflection portions of the study. We asked all questions to participants verbally, while also screensharing a slide with the written question for the participant to read and refer to. Three researchers collaboratively developed the coding schema used to analyze these quotes and together scored 20% of participants, discussing and resolving differences. The first author then independently scored the remaining participants according to the agreed upon scoring system.

*4.4.1 RQ1-Developing Understanding: What is RL.* To examine if participants developed an understanding of reinforcement learning through this experience, we evaluated their knowledge at the end of the session. Similar to evaluative methodology in prior work on educational technology [17], we asked participants to describe what reinforcement learning is, how they used reinforcement learning during the session, and to provide a novel example of reinforcement learning. We looked for key response elements for each answer, and we report percentages of participants who included each element alongside representative quotes to lend further insight.

*4.4.2 RQ2-Developing Understanding: New Track Scenario.* To see if and how participants developed an ability to identify and solve issues with individual models, as part of the reflection and feedback section we presented them with a novel track scenario and a failing robot. This robot successfully followed a wide right turn before failing to turn when that wide right turn transitioned into a very tight left turn. We asked the participants what kind of training track they would create to train this robot; we were looking for participants to specify a tight turn, a left turn, or a transition from a right turn to a left turn. We report on the content of these responses in terms of percentage of participants that included each correct element.

*4.4.3 RQ3-Developing Understanding: RL or Code Scenarios.* To probe participants' understandings of the strengths and limitations of RL, we provided them with a series of eight scenarios in robotics (e.g., stop at a stop sign or navigate around an unexpected obstacle) and asked them if the scenario is better solved with RL or programming. We scored their responses against our answer key; four of the scenarios were intended to be solved with RL and four with code.

*4.4.4 RQ4-Engagement.* To measure participants' engagement with the ARtonomous experience, we used the Giggle Gauge, a self-report engagement metric developed for children [18]. This metric was originally created for use with a bifurcated 4-point scale appropriate to the cognitive limits of children as young as four years old. However, given our older participants, we used the same prompts with a 7-point Likert-type response to allow for more gradation in answers.

*4.4.5 RQ5-Curiosity: Content Analysis of Questions.* To qualitatively describe participants' curiosity about RL after this experience, we showed them a short video of a self-driving car driving along surface streets and a highway. We then asked, "What questions would you like to ask the creator of this self-driving car?" We report a brief content analysis of these questions to understand where participants' curiosities lie.

*4.4.6 RQ5-Curiosity: Desire to Learn More.* We quantitatively measured participants' curiosity about RL after this experience by asking them to respond on a 7-point scale the extent to which they would like to learn more about RL.

## 4.5 Limitations in Study Design

We note and explain one deliberate choice that contributes to a limitation in this study design: the constrained time. We chose to limit our study to 90 minutes because we imagine long term usage of this system would accompany more complex design and programming tasks; students might kick off model training and then switch over to programming while waiting for that training to finish. That is, we imagine a child using this system in a classroom setting might interact directly with the model creation interface for short spurts, interspersed with other learning activities (e.g., using the programming interface). Our research questions, though, did not focus heavily on the coding portions of this app. Therefore, we chose to forego a complex coding task that would have further limited our recruitment to students with greater familiarity with Swift. Furthermore, implementing rapid training and condensing these separate interactions into a single experience simplified recruitment and remote study participation for both researchers and for participating families (e.g., for parents who had to give up a work computer for their child to call into the session).

# 5 RESULTS

We report findings regarding understanding of RL, engagement, and curiosity across all 15 participants. All participants completed all three rounds of model training and wrote code that met the objective.

## 5.1 RQ1-What is RL

Participants reported a definition of RL, the way they used RL in the experience, and a novel example of RL. While only 13.3% ($n$ = 2) of participants explained that RL was a form of ML (perhaps perceived as an out-of-scope response given the context of the question), 60.0% ($n$ = 9) described an agent getting an input, 73.3% ($n$ = 11) described an agent choosing an action and receiving rewards or penalties, and 93.3% ($n$ = 14) discussed an agent that learned from rewards, penalties, or experiences for future new situations. Participant 11 was one of 6 participants who included all of inputs, actions, rewards, and improvements in their response:

> "It's when you have some something do something. And it takes your input, and it does an action, and then you give it like—like points, or not points or whatever, um, so that it knows when it did something wrong and when it did something right. So that it can try to do the right thing next time." *–P11, age 12*

When considering how they used RL in the activity, 86.7% ($n$ = 13) of participants explained that they used it to train a robot to navigate, but only 40.0% ($n$ = 6) spoke about creating targeted training tracks to help it learn and just 20% ($n$ = 3) discussed identifying problems in the models, the robot receiving rewards or penalties, or the robot's improvement based on the training. Interestingly, though, 26.7% ($n$ = 4) of the participants mentioned their own learning as a form of reinforcement learning. For instance, they described that during the process of coding they could run their code to see when the robot did "good things" or "bad things" and use that to adjust their code for the next run:

> "I had to look at the track to see if it was doing what I was supposed to do in the objective. And then I used what it did on the track the first time...to change or keep the run in the second time. And then in the end I ended up doing what the objective wanted me to do." *–P5, age 13*

Finally, when describing a new example of reinforcement learning, 93.3% ($n$ = 14) participants provided a novel example about learning some kind of behavior or material, 66.7% ($n$ = 10) explicitly discussed a reward or penalty, and 73.3% ($n$ = 11) explained how gaining experiences might change future outcomes.

> "If you...don't do your homework one day then your teacher punishes you, then the next day you're going to do your homework because the teacher punished you for not doing your homework." *–P15, age 13*

## 5.2 RQ2-New Track Scenario

All participants described one of the three reasonable training tracks for the failing robot, with 86.7% ($n = 13$) describing a tight turn and 26.7% ($n = 4$) suggesting a left turn track (note: some participants included both descriptions in their answer). Additionally 13.3% ($n = 2$) specified that the track should go from one turn into another to specifically target the failing scenario of not transitioning from a right turn to a left turn.

## 5.3 RQ3-RL or Code Scenarios

Participants were successfully able to distinguish which robotics scenarios might be best solved with reinforcement learning and which with code. Out of eight scenario prompts, participants on average answered 85% correctly (SD=18%). A single-tailed t-test confirms these scores are significantly above chance, $t(15) = 7.90, p < 0.001$.

## 5.4 RQ4-Engagement

We evaluated participant engagement with the ARtonomous experience using a modification of the Giggle Gauge [18] with a 7-point Likert-type response. Overall we found reasonably high levels of engagement ($M = 5.99, SD = 1.02$), with all items scoring above 6 on average except aesthetics ($M = 5.63, SD = 1.23$) and challenge ($M = 5.53, SD = 1.19$). Notably, if we drop data from one participant who experienced an early bug that led to a multi-second UI delays, we see average engagement increases to 6.17 ($SD = 0.77$).

## 5.5 RQ5-Inspired Curiosity

*5.5.1 Content Analysis of Questions.* We categorize the questions participants would want to ask the the creator of a self-driving car into questions about RL, questions about broader machine learning, and non-ML related questions. Most questions fall into the first two categories, indicating a curiosity about machine learning and its applications. Specifically, 40% ($n = 6$) of the participants asked about reinforcement learning (e.g., How many episodes did it take to drive so smoothly?) and another 40% ($n = 6$) about machine learning. Half of machine learning questions related to computer vision or object detection (e.g., How do you program your camera to recognize objects nearby?) and half related to machine decision making (e.g., How does the car deal with something unexpected on the road, like if there's a big rock?). Of the remaining participants, 6.7% ($n = 1$) asked about user experiences/design (i.e., Why is there a steering wheel if the car drives itself?), 6.7% ($n = 1$) listed categories they might discuss (i.e., the complexity of real life compared to simulation) without directly stating a question, and 6.7% ($n = 1$) stated that they had no questions.

*5.5.2 Desire to Learn More.* When asked on a scale from one to seven if they would like to learn more about reinforcement learning, participants unanimously reported that they would. All participants responded with a score of 5 or higher, for a mean score of 6.47 ($SD = 0.74$).

# 6 DISCUSSION

We began this project with concerns about how the gap between state-of-the-art robotics and educational robotics widens with advancements in machine learning. Further, with educational robotics placing a heavy emphasis on mechanical engineering, the high cost of robotics kits designed for youth can present barriers to participation. By addressing cost and constraints with virtual robots and by building upon new machine learning based conceptual infrastructures, ARtonomous is a more financially and physically accessible educational robotics tool reflective of the types of data-driven approaches used in professional robotics today.

Our evaluation demonstrates that participants were able to develop an understanding of reinforcement learning through the ARtonomous experience (i.e., what it is, how it fails, and when it applies), that they were highly engaged with the application, and that they were curious to learn more about reinforcement learning and its real-word uses. All participants could suggest reasonable training tracks for a novel track scenario and were able to—at least in part—explain reinforcement learning, its usage, and an example thereof. Participants reported high levels of engagement with the app and an even higher desire to learn more about the subject material. Collectively, these results demonstrate ARtonomous' success in addressing our key research objectives: 1) creating low cost, high engagement robotics education tools that help learners explore reinforcement learning and 2) bridging the gap between educational and professional robotics to develop student understanding of reinforcement learning, support positive engagement with the content, and build student curiosity toward machine learning more broadly.

## 6.1 Low-Cost and High Engagement with Virtual Robotics

Surveying the landscape of educational robotics tools, the cost of specialized hardware can inhibit broad reach. In the United States in particular, economic disadvantage is strongly correlated with race and ethnicity. Consequently, expensive tools can contribute to the maintenance of racist barriers to educational equality.

These concerns validate core premises of the ARtonomous vision: that we can use general purpose computing devices to simulate robotic systems and visualize interaction between those systems and the surrounding environment. Our research illustrates that training and programming simulated robots is engaging and educational for participants. This result is consistent with prior work on virtual robotics [8, 21, 56], with the additional inclusion of ML model training.

Looking ahead, we foresee that augmented reality (AR) functionality has the potential to take this engagement and learning even further [16], still without requiring specialized (and potentially cost-prohibitive) hardware. We built prototype functionality of AR integration that allows robot training—including replays, test simulations, and overlay visualizations—to appear in the physical-virtual space (see Figure 4). Although we chose to forego testing of the AR components within this tool, instead prioritizing the geographic and economic diversity of our participant pool, a large body of prior work shows that AR educational experiences can improve learning and engagement [16, 44]. We look forward to future work exploring AR's potential in robotics education contexts.

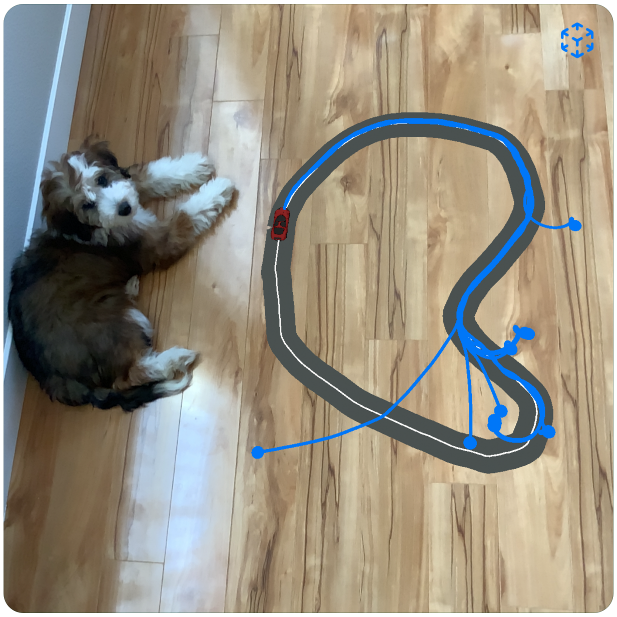


Figure 4: Prototype augmented reality features allow users to view their track, robot, and training overlay details in physical space.

## 6.2 Bridging the Gap Between Educational and Professional Robotics

All fields of disciplined human endeavor involve representational infrastructures—tools that participants in those fields use to frame, communicate about, and accomplish work [29, 49]. Learning a field, whether robotics, archaeology, or policing [28], is fundamentally about learning to see and do as those already within that field do. As we discuss above, the rapid advancement of ML in professional robotics combined with the relative absence of ML in educational robotics is progressively relegating educational robotics into a less and less authentic enterprise. The representational infrastructure of educational robotics is generally lacking methods of ML, including RL, and norms that emphasize the contemporary importance of autonomy. Our findings show 1) that RL (and ML) can be a part of how students learn robotics and 2) that the representational infrastructure gap between robotics and educational robotics is closable via new systems like the one we present here.

Our research illustrates how new systems can foster approaches to teaching and learning that place more emphasis on the computer science of ML and programming. The iPad-based ARtonomous experience we evaluated involves several representational practices that are lacking or uncommon in today's most common educational robotics platforms and experiences, including creating idealized contexts (i.e., training and simulation environments) for robot operation, identifying specific examples of problematic or successful robotic behavior from simulations within those contexts, noting patterns in aggregated replays of multiple simulations, drawing focused training scenarios to address problematic patterns, evaluating ML model improvement over time, juxtaposing code with simulations of RL-based model execution, and observing and debugging the execution of programs that are combinations of RL models and user-defined code.

Some of these representational practices have analogues in today's physical robotics education experiences. For example, participants might sketch arenas that they wish for their robots to execute within and then draw the trajectories that they want their robots to follow within those arenas. They might video record their robots in action and replay those later in order to share their successes or document opportunities for improvement. Virtual robotics platforms, such as the Virtual Robotics Toolkit [15], also afford the ability to capture and replay simulations.

However, the essence of this work—iteratively training an ML-based navigation model using a series of training environments designed in light of past model performance and then integrating the ML model with user-authored code—involves a set of practices that are not available in other tools but are reflective of broader AI education objectives [36, 52]. Moreover, ML could enable young people to create robotic systems that are currently difficult, if not impossible, for them to do with programming alone (e.g., create autonomous vehicles that navigate model cities without pre-programmed routes). We further envision future RL education platforms for middle school learners that introduce additional aspects of model training (e.g., observation space, exploration rate, or reward function) in similarly engaging and accessible ways. The approachability of ML practices, as evidenced by our findings, raises an essential question for robotics education: What are the most important learning goals for robotics education in a world where authentic robotics practice involves a data-driven synthesis of ML and programming in order to create autonomous systems? This work suggests that the skills of creating, assessing, and integrating ML models should and could be among those goals.

# 7 CONCLUSION

We presented ARtonomous, a tablet-based application that introduces middle school students to reinforcement learning through virtual autonomous robotics. We address cost constraints in educational robotics while still maintaining high engagement by leveraging a virtual robot approach. Further, we aim to close the gap between educational and professional robotics through the introduction of machine learning techniques. Through a 90-minute evaluation study with middle school aged students (ages 11–14) we demonstrated the efficacy of ARtonomous in achieving these design goals. Specifically, ARtonomous supported participants in developing an understanding of reinforcement learning, led to high-levels of self-reported engagement, and inspired curiosity among participants for further learning about RL. Having demonstrated that RL can be made accessible and approachable for middle school students and that it can integrate into robotics education, our work highlights opportunities for robotics educators, HCI and education researchers, and product developers to reassess the most important learning goals in educational robotics and to consider making space for RL or ML via virtual simulation methods.

# 8 SELECTION AND PARTICIPATION OF CHILDREN

Youth were recruited from a school for students with reading-related learning differences, a hardware and robotics camp in Colorado, a middle school in Southern California serving a lower-income community, and via an email list to employees at a large technology company. We chose these sites to recruit participants with a range of geographic, economic, and personal backgrounds. All youth provided verbal assent to participate in the study and to be video recorded, and youth and parents signed consent forms prior to video and audio data collection. Families were told they could choose to end the study at any time and still receive a gift card; families received a $25 gift card in exchange for participating.

# REFERENCES

[1] Adam Agassi, Hadas Erel, Iddo Yehoshua Wald, and Oren Zuckerman. 2019. Scratch nodes ML: A playful system for children to create gesture recognition classifiers. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems.* 1–6.

[2] Batu Akan, Afshin Ameri, Baran Cürüklü, and Lars Asplund. 2011. Intuitive industrial robot programming through incremental multimodal language and augmented reality. In *2011 IEEE International Conference on Robotics and Automation.* IEEE, 3934–3939.

[3] Safinah Ali, Daniella DiPaola, Irene Lee, Jenna Hong, and Cynthia Breazeal. 2021. Exploring Generative Models with Middle School Students. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* 1–13.

[4] Francisco Bellas, Martin Naya, Gervasio Varela, Luis Llamas, Abraham Prieto, Juan Carlos Becerra, Moises Bautista, Andres Faiña, and Richard Duro. 2017. The Robobo project: Bringing educational robotics closer to real-world applications. In *International Conference on Robotics and Education RiE 2017.* Springer, 226–237.

[5] Erin Beneteau, Yini Guan, Olivia K Richards, Mingrui Ray Zhang, Julie A Kientz, Jason Yip, and Alexis Hiniker. 2020. Assumptions Checked: How Families Learn About and Use the Echo Dot. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–23.

[6] Erin Beneteau, Olivia K Richards, Mingrui Zhang, Julie A Kientz, Jason Yip, and Alexis Hiniker. 2019. Communication breakdowns between families and Alexa. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.* 1–13.

[7] Fabiane Barreto Vavassori Benitti. 2012. Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education* 58, 3 (2012), 978–988.

[8] Matthew Berland and Uri Wilensky. 2015. Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology* 24, 5 (2015), 628–647.

[9] Paulo Blikstein. 2008. Travels in Troy with Freire: Technology as an agent of emancipation. In *Social Justice Education for Teachers.* Brill Sense, 205–235.

[10] Paulo Blikstein et al. 2015. Computationally Enhanced Toolkits for Children: Historical Review and a Framework for Future Design. *Found. Trends Hum. Comput. Interact.* 9, 1 (2015), 1–68.

[11] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).

[12] Yuanzhi Cao, Zhuangying Xu, Fan Li, Wentao Zhong, Ke Huo, and Karthik Ramani. 2019. V. Ra: An in-situ visual authoring system for robot-IoT task planning with augmented reality. In *Proceedings of the 2019 on Designing Interactive Systems Conference.* 1059–1070.

[13] Mark Cheli, Jivko Sinapov, Ethan E Danahy, and Chris Rogers. 2018. 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).*

[14] Yu-Hui Ching, Yu-Chang Hsu, and Sally Baldwin. 2018. Developing computational thinking with educational technologies for young learners. *TechTrends* 62, 6 (2018), 563–573.

[15] Cogmation. [n.d.]. Virtual Robotics Toolkit. https://www.virtualroboticstoolkit.com

[16] Chris Dede. 2009. Immersive interfaces for engagement and learning. *science* 323, 5910 (2009), 66–69.

[17] Griffin Dietz, Jimmy K Le, Nadin Tamer, Jenny Han, Hyowon Gweon, Elizabeth L Murnane, and James A Landay. 2021. StoryCoder: Teaching Computational Thinking Concepts Through Storytelling in a Voice-Guided App for Children. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* 1–15.

[18] Griffin Dietz, Zachary Pease, Brenna McNally, and Elizabeth Foss. 2020. Giggle gauge: a self-report instrument for evaluating children's engagement with technology. In *Proceedings of the Interaction Design and Children Conference.* 614–623.

[19] DJI. [n.d.]. RoboMaster S1. https://www.dji.com/robomaster-s1

[20] Stefania Druga. 2018. *Growing up with AI: Cognimates: from coding to teaching machines.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[21] Amy Eguchi and Jiayao Shen. 2012. Student learning experience through CoSpace educational robotics. In *Society for Information Technology & Teacher Education International Conference.* Association for the Advancement of Computing in Education (AACE), 19–24.

[22] Motahhare Eslami, Karrie Karahalios, Christian Sandvig, Kristen Vaccaro, Aimee Rickman, Kevin Hamilton, and Alex Kirlik. 2016. First I "like" it, then I hide it: Folk Theories of Social Feeds. In *Proceedings of the 2016 cHI conference on human factors in computing systems.* 2371–2382.

[23] Motahhare Eslami, Aimee Rickman, Kristen Vaccaro, Amirhossein Aleyasen, Andy Vuong, Karrie Karahalios, Kevin Hamilton, and Christian Sandvig. 2015. "I always assumed that I wasn't really that close to [her]": Reasoning about Invisible Algorithms in News Feeds. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems.* 153–162.

[24] David Fernandes, Francisco Pinheiro, André Dias, Alfredo Martins, Jose Almeida, and Eduardo Silva. 2019. Teaching robotics with a simulator environment developed for the autonomous driving competition. In *International Conference on Robotics in Education (RiE).* Springer, 387–399.

[25] FIRST. 2021. Coach's Playbook. https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/coachs-playbook.pdf

[26] National Center for Education Statistics. 2018. Average class size in public schools, by class type and state: 2017–18. https://nces.ed.gov/surveys/ntps/tables/ntps1718_fltable06_t1s.asp

[27] For Inspiration and Recognition of Science and Technology. [n.d.]. FIRST Robotics. https://www.firstinspires.org/

[28] Charles Goodwin. 1994. Professional vision. *American Anthropologist* 96, 3 (1994), 606.

[29] Rogers Hall, Reed Stevens, and Tony Torralba. 2002. Disrupting representational infrastructure in conversations across disciplines. *Mind, Culture, and Activity* 9, 3 (2002), 179–210.

[30] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. 2018. Stable Baselines. https://github.com/hill-a/stable-baselines.

[31] Innovation Center of St. Vrain Valley Schools. [n.d.]. Robotics. https://innovation.svvsd.org/focus-areas/robotics/

[32] Golnaz Arastoopour Irgens, Knight Simon, Alyssa Wise, Thomas Philip, Maria C Olivares, Sarah Van Wart, Sepehr Vakil, Jessica Marshall, Tapan S Parikh, M Lisette Lopez, et al. 2020. Data literacies and social justice: Exploring critical data literacies through sociocultural perspectives. (2020).

[33] Ken Kahn and Niall Winters. 2017. Child-friendly programming interfaces to AI cloud services. In *European Conference on Technology Enhanced Learning.* Springer, 566–570.

[34] Betul Keles, Niall McCrae, and Annmarie Grealish. 2020. A systematic review: the influence of social media on depression, anxiety and psychological distress in adolescents. *International Journal of Adolescence and Youth* 25, 1 (2020), 79–93.

[35] Matthew V Law, Zhilong Li, Amit Rajesh, Nikhil Dhawan, Amritansh Kwatra, and Guy Hoffman. 2021. Hammers for Robots: Designing Tools for Reinforcement Learning Agents. In *Designing Interactive Systems Conference 2021.* 1638–1653.

[36] Duri Long and Brian Magerko. 2020. What is AI literacy? Competencies and design considerations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.* 1–16.

[37] Makeblock. [n.d.]. mBot. https://www.makeblock.com/steam-kits/mbot

[38] Jane Margolis, Rachel Estrella, Joanna Goode, Jennifer Jellison Holme, and Kim Nao. 2017. *Stuck in the shallow end: Education, race, and computing.* MIT press.

[39] Ángel Martínez-Tenor, Ana Cruz-Martín, and Juan-Antonio Fernández-Madrigal. 2019. Teaching machine learning in robotics interactively: The case of reinforcement learning with Lego® Mindstorms. *Interactive Learning Environments* 27, 3 (2019), 293–306.

[40] MIT App Inventor. [n.d.]. Artificial Intelligence with MIT App Inventor. https://appinventor.mit.edu/explore/ai-with-mit-app-inventor

[41] Nintendo. 2020. Mario Kart Live: Home Circuit. https://mklive.nintendo.com

[42] William Christopher Payne, Yoav Bergner, Mary Etta West, Carlie Charp, R Benjamin Benjamin Shapiro, Danielle Albers Szafir, Edd V Taylor, and Kayla DesPortes. 2021. danceON: Culturally Responsive Creative Computing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* 1–16.

[43] Roy D Pea. 2004. The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *The journal of the learning sciences* 13, 3 (2004), 423–451.

[44] Iulian Radu. 2012. Why should my students use AR? A comparative review of the educational impacts of augmented-reality. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 313–314.

[45] Martin Riedmiller, Thomas Gabel, Roland Hafner, and Sascha Lange. 2009. Reinforcement learning for robot soccer. *Autonomous Robots* 27, 1 (2009), 55–73.

[46] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. 2017. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging* 2017, 19 (2017), 70–76.

[47] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* 362, 6419 (2018), 1140–1144.

[48] Sphero, Inc. [n.d.]. Sphero. https://www.sphero.com

[49] Lucy A Suchman. 1988. Representing Practice in Cognitive Science. *Human Studies* (1988), 305–325.

[50] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction.* MIT press.

[51] The LEGO Group. [n.d.]. LEGO Mindstorms. https://www.lego.com/en-us/themes/mindstorms/about

[52] David Touretzky, Christina Gardner-McCune, Fred Martin, and Deborah Seehorn. 2019. Envisioning AI for k-12: What should every child know about AI?. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9795–9799.

[53] Ana M Villanueva, Ziyi Liu, Zhengzhe Zhu, Xin Du, Joey Huang, Kylie A Peppler, and Karthik Ramani. 2021. RobotAR: An Augmented Reality Compatible Teleconsulting Robotics Toolkit for Augmented Makerspace Experiences. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.

[54] Michael Walker, Hooman Hedayati, Jennifer Lee, and Daniel Szafir. 2018. Communicating robot motion intent with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 316–324.

[55] Xiaoyu Wan, Xiaofei Zhou, Zaiqiao Ye, Chase K Mortensen, and Zhen Bai. 2020. SmileyCluster: supporting accessible machine learning in K-12 scientific discovery. In *Proceedings of the Interaction Design and Children Conference*. 23–35.

[56] Eben B Witherspoon, Ross M Higashi, Christian D Schunn, Emily C Baehr, and Robin Shoop. 2017. Developing computational thinking through a virtual robotics programming curriculum. *ACM Transactions on Computing Education (TOCE)* 18, 1 (2017), 1–20.

[57] Wonder Workshop. [n.d.]. Cue Robot. https://www.makewonder.com

[58] Mike Woodcock. 2019. Solving CarRacing with PPO. https://notanymike.github.io/Solving-CarRacing/

[59] Ziyi Zhang, Samuel Micah Akai-Nettey, Adonai Addo, Chris Rogers, and Jivko Sinapov. 2021. An Augmented Reality Platform for Introducing Reinforcement Learning to K-12 Students with Robots. *arXiv preprint arXiv:2110.04697* (2021).

[60] Ziyi Zhang, Sara Willner-Giwerc, Jivko Sinapov, Jennifer Cross, and Chris Rogers. 2021. An Interactive Robot Platform for Introducing Reinforcement Learning to K-12 Students. In *International Conference on Robotics in Education (RiE)*. Springer, 288–301.

[61] Abigail Zimmermann-Niefield, Shawn Polson, Celeste Moreno, and R Benjamin Shapiro. 2020. Youth making machine learning models for gesture-controlled interactive media. In *Proceedings of the Interaction Design and Children Conference*. 63–74.

[62] Abigail Zimmermann-Niefield, Makenna Turner, Bridget Murphy, Shaun K Kane, and R Benjamin Shapiro. 2019. Youth learning machine learning through building models of athletic moves. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*. 121–132.