# MoveToCode: An Embodied Augmented Reality Visual Programming Language with an Autonomous Robot Tutor for Promoting Student Programming Curiosity

Thomas R Groechel, Ipek Goktan, Karen Ly, Anna-Maria Velentza, and Maja J Matarić

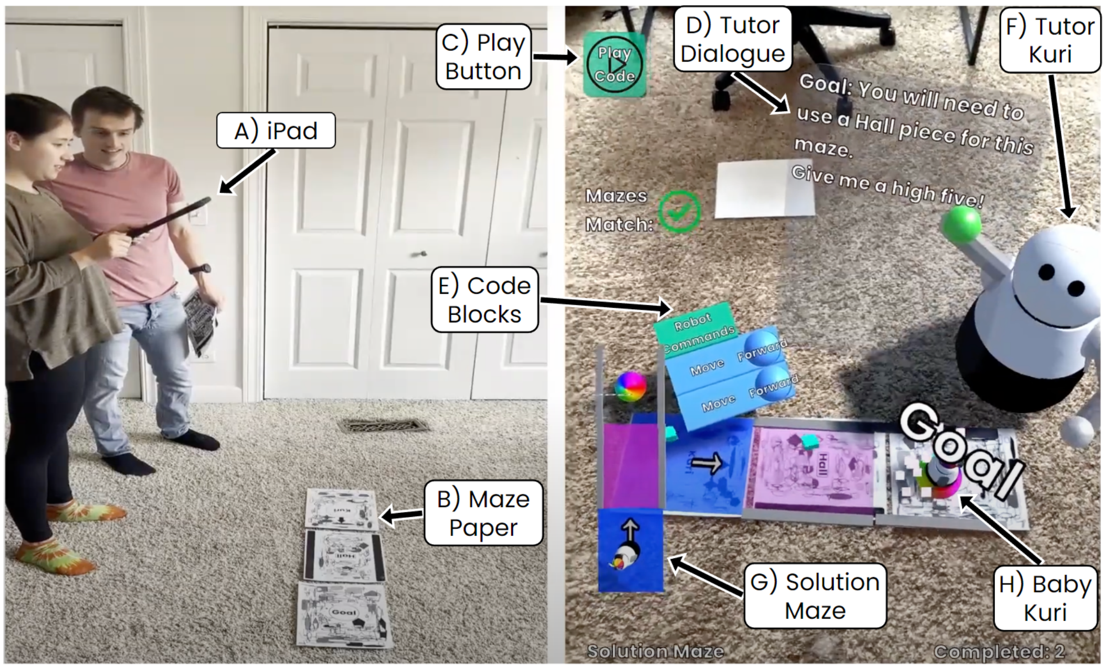


Fig. 1: MoveToCode pair-programming exercise. Left) Programmers. Right) MoveToCode activity view. A) Vertically-held mobile tablet; B) tangible maze paper tracked by the tablet; C) code play button; D) virtual tutor dialogue; E) code blocks that control the miniature virtual robot through the maze; F) autonomous, augmented reality robot tutor Kuri poised for a high five; G) goal maze configuration; & H) miniature virtual robot starting tile and programmed to reach the goal tile.

***Abstract*— Virtual, augmented, and mixed reality for human-robot interaction (VAM-HRI) is a new and rapidly growing field of research. The field of socially assistive robot (SAR) has made impactful advances in educational settings, but has not yet benefited from VAM-HRI advances. We developed MoveToCode - an open-source, embodied (i.e., kinesthetic) learning visual programming language that aims to increase student (ages 8-12) curiosity during programming. MoveToCode uses an augmented reality (AR) autonomous robot tutor named Kuri that models the students' kinesthetic curiosity and acts to promote their curiosity in programming. MoveToCode design was informed by pilot studies and tested in Los Angeles elementary classrooms ($n = 21$). Results from main study validated our design decisions compared to the pilot study which was conducted in a real elementary school classroom environment ($n = 15$), showing an improvement in perceived robot helpfulness (median $+\Delta 1.25$ out of 5) and number of completed exercises (median $+\Delta 1$, maximum of 11). While no significant changes were found in pre/post student curiosity or intention to program later in life, students wrote more open-ended questions post-study on topics related to robots, programming, research, and if they would like to do the activity again. This work demonstrates the potential of using VAM-HRI in a kinesthetic context for SAR tutors, and highlights the existing conventions and new design considerations for creating AR applications for SAR.**

This work was supported by the NSF NRI 2.0 grant for "Communicate, Share, Adapt: A Mixed Reality Framework for Facilitating Robot Integration and Customization", NSF IIS-1925083. Anna-Maria Velentza was funded by the Fulbright Foundation, Greece.

Thomas Groechel, Ipek Goktan, Karen Ly, Anna-Maria Velentza, and Maja J Matarić are all with the Interaction Lab, Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA {groechel, ipekg, karenly, av_281, mataric}@usc.edu

## I. INTRODUCTION

Socially assistive robot (SAR) tutors have been demonstrated to be effective in supporting student learning of both cognitive and socio-emotional skills (e.g., curiosity). Even a brief interaction can lead humans to develop a bond with a SAR [1], and encourage a receptive approach towards technology and innovative problem-solving, resulting in greater levels of success, strengthening positive emotions [2]. The development of dynamic learning systems with SAR, for a variety of topics but especially for computer programming, has been repeatedly proposed for primary education [3].

Programming education usually covers fundamental programming language syntax, its implementation, and associated practical assignments. Nevertheless, teachers often devote significant time debugging students' programs rather than teaching the material, impeding the progress of programming instruction and demotivating students.

Virtual, Augmented, and Mixed Reality for Human-Robot Interaction (VAM-HRI) is a new area of research that uses 3D virtual imagery to enhance human-robot interaction. The field has seen rapid growth in recent years, in part due to the increasing availability of VAM technology, such as commercial augmented and virtual reality devices [4], as well as the VAM-HRI community workshops that have brought together researchers and practitioners [5]. VAM-HRI has been applied in a variety of domains, including manufacturing [6], training [7], and research [8].

Leveraging VAM-HRI in SAR for teaching programming represents a particularly promising new area of research to address some of the above-mentioned challenges. Toward that end, we developed MoveToCode, an open-source, embodied (i.e., kinesthetic) learning visual programming language that aims to increase student (ages 8-12) curiosity during programming, using an AR software application supported by a virtual SAR robot tutor. This paper describes the iterative MoveToCode design and evaluation process.

## II. Background and Related Work

Combining hands-on work and coding increases student interest and involvement in programming practices and principles [9], particularly when involving social and creative learning [10]. Implementing a computer game is a learning activity that inspires motivation, enthusiasm, and engagement with the material [11]. Visual programming environments are an effective way to introduce programming concepts; students find them fun and practical, leading to high motivation and positive attitudes toward coding [12]. Additionally, combining hands-on work with coding can improve student understanding of programming principles such as loops, conditionals, and events, as well as practices such as remixing, testing, and debugging [9], [13].

Past work measured children's curiosity after interacting with a tablet story-maker app and a SAR, both displaying a curious personality. There was no significant difference in curiosity measures between the two conditions, and both performed better than the non-curious condition in exploration tasks and question generation [14]. Similarly, the presence of a SAR in an elementary school encouraged children to ask science-related questions, and increased curiosity about science by children who asked questions, but did not impact the overall science curiosity of the class [15].

Student interaction in collaborative AR learning environments has promise for enhancing learning [16]. The combination of AR learning environments with intelligent agents encourages kinesthetic learning, creative engagement, and adaptability to the student's needs [17]. We created MoveToCode to explore design conventions and potential benefits of kinesthetic interactions using VAM and SAR.

## III. Technical Approach

MoveToCode was designed through a series of university ethics board-approved pilot studies, culminating in the main study, as described in Sec. IV. The studies included transitioning MoveToCode from costly AR headsets to more affordable tablets, redesigning the learning exercise to support tangible pair-programming (Fig. 2), and revising the actions and policy of the robot tutor. Throughout this process, we blended known design conventions with new considerations for the embodied learning AR activity with a robot tutor. MoveToCode is freely distributed at `https://github.com/interaction-lab/MoveToCode`.

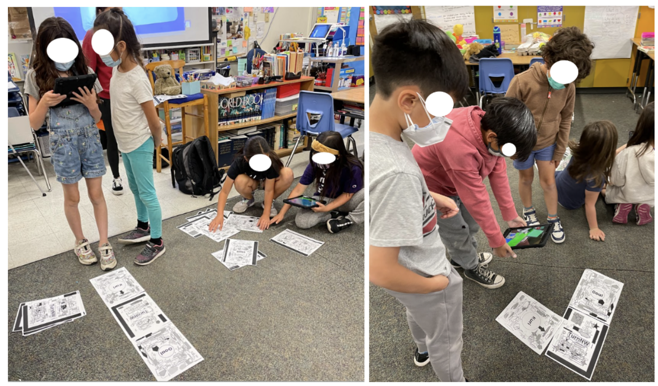

Fig. 2: Images from a MoveToCode pilot study with 8-12 year old students in Los Angeles.

### A. MoveToCode Implementation and Exercises

MoveToCode was originally designed for teaching traditional computing concepts such as print statements, math equations, if statements, variables, and looping through console-based exercises. The design did not have any physical connection to the real world [18].

We changed the MoveToCode design to involve programming a miniature virtual robot "baby Kuri" (Fig. 1.H), with the help of a normal-sized autonomous robot tutor called "tutor Kuri" (Fig. 1.F). Both robots were 3D models of the Mayfield Kuri; tutor Kuri also had socially expressive arms from Groechel et al. [19]. The MoveToCode design task was to program baby Kuri's path through a maze created by the student using physical pieces of paper. Gamified programming approaches have been shown to be effective for increasing computation thinking skills in a variety of systems and age groups [20].

Our MoveToCode exercises used two modes (Fig. 3). In mode 1, students rearrange physical pieces of paper to create a maze identical to the "Solution Maze" (Fig. 1.G). Maze pieces had connector pieces that lined up and highlight when they were connected (Fig. 3.B). When the maze was identical to the solution, the student holding the iPad could press a "Lock Maze" button, locking the connectors and transitioning the app to mode 2. In mode 2, the student uses 3D code blocks (Fig. 1.E) to program baby Kuri to navigate the designed maze.

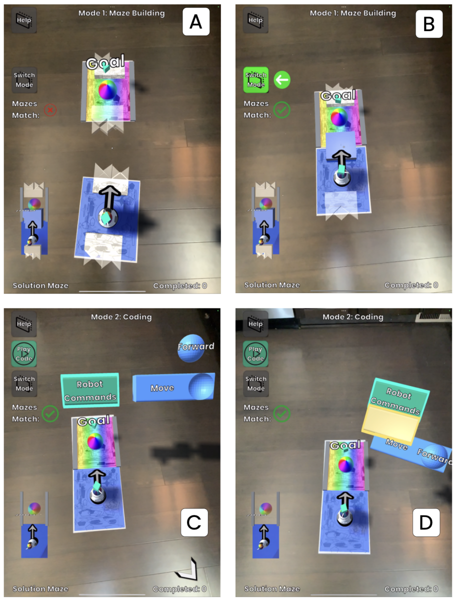


Fig. 3: The two modes of our MoveToCode exercises: in mode 1 (A & B) the user connects maze pieces to match a solution maze, and in mode 2 (C & D) the user programs baby Kuri to complete that maze.

The exercises covered computational thinking concepts including sequencing, looping, and using different blocks to solve the same problem (e.g., `Move(Forward)` $\approx$ `Turn(Left)`$\rightarrow$`Turn(Left)`$\rightarrow$`Move(Backward)`). Ten exercises were created and ordered based on the time taken to complete them in a pilot study. All necessary code blocks for each exercise were provided. As the exercises become more complex, erroneous blocks were also added that were either 1) not part of a correct solution or 2) `Set Color + Color` blocks that allowed students to change baby Kuri'as color but had no effect on the desired solution.

### B. Kinesthetic Curiosity Habituation & Tutor Action Policy

Tutor Kuri's action selection was based on the student's *kinesthetic curiosity* ($KC^S$) [21] – a multimodal measure we developed that combines the student's movement and curiosity measures into a single, personalized measure. At a given time $t$, $KC^S_t$ is defined as:

$$movement^S_t = \sum_{n=t-tw+1}^{t} dist(pose_n, pose_{n-1}) \tag{1}$$

$$curiosity^S_t = \sum_{n=t-tw}^{t} [ISA^S_n \neq NULL] \tag{2}$$

$$KC^S_t = w_0 * z(movement^S_t) + w_1 * z(curiosity^S_t) \tag{3}$$

where $movement^S_t$ (1) is measured with accumulated head pose change over a sliding time window $tw$, and $curiosity^S_t$ (2) is measured as the sum of information seeking actions (ISAs) over $tw$. Customizable weights $w_0$ and $w_1$ were both set to 0.5 but could be adjusted to favor student movement or curiosity. ISA scores were defined relative to the domain and action space of the student. Improving upon the original definition, we introduced *habituation saliency* [22] for a given human action, defined as:

$$KDS = 1 - \frac{ActionCounts[A_t]}{max(ActionCounts)} \tag{4}$$

$$TS = min(\frac{(t - lastRecordedTime(A_t))^2}{60^2}, 1) \tag{5}$$

$$ISA^S_t = \begin{cases} 2 & \text{if } A_t \text{ has never been done} \\ 0 & \text{if } A_t \text{ is recorded in last } tw \\ KDS + TS & \text{otherwise} \end{cases} \tag{6}$$

where $A_t$ is the human action taken at time $t$ (in seconds), $ActionCounts$ is a map of unique actions to the total number of times each action was completed, $KDS$ represents knowledge-driven saliency, and $TS$ represents temporal saliency. $KDS$ is discounted as an action is completed, normalized by the max number of times any action is completed. $TS$ grows at a quadratic rate by squaring the difference of time $t$ and the last recorded time of $A_t$, followed by normalization via squaring a max time constant of 60 seconds. Maxiumum $TS$ is set to 1, denoting any action done $\geq$ 1 minute ago.

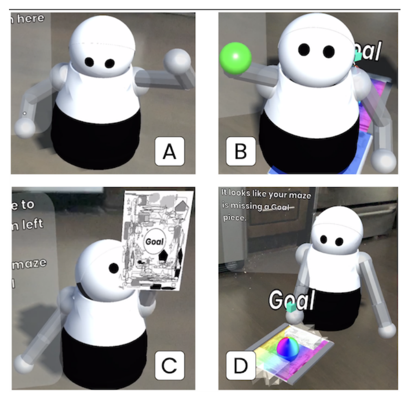


Fig. 4: A subset of tutor Kuri actions: A) wave, B) high five, C) showing a type of missing paper, and D) moving to and pointing at a misaligned maze piece.

As shown in Fig. 4, tutor Kuri actions included *context-dependent helpful actions* indicated below with *:

- Idle and look around
- Wave to user
- Interactive high-five
- Move out of the user's way
- Dialogue (Fig. 1.H)
  - Exercise goal
  - Congratulatory phrases
  - * Encouragement phrases
  - * Referencing a maze piece or code block
- * Showing a type of maze paper not yet used
- * Moving and pointing to a misaligned maze piece
- * Moving and pointing to a misaligned code block

Tutor Kuri's action policy was designed to select *context-dependent helpful actions* whenever $KC^S_t < 0.5$ and the last time Kuri performed an action was $> tw$. The 0.5 threshold was chosen because it had been shown to produce higher short-term $KC_t^S$ scores compared to a lower threshold [21]. Tutor Kuri gave the exercise goal at the start of a new exercise and offered a high-five with accompanying congratulatory dialogue upon exercise completion. It moved out of the user's way whenever it was not performing an action, collided with virtual objects (often maze pieces), or was $< 0.75m$ from the user. The target position was calculated in the horizontal plane (i.e., $y = 0$ for a 3D $\{x, y, z\}$ vector where $y$ was defined as up) as follows: a vector from the collided object to the user was calculated and normalized as $\hat{V_C}$. A vector $\hat{V_{avg}}$ was calculated as the vector between $\hat{V_C}$ and the user's unit forward vector. $\hat{V_{avg}}$ was added to the user's position to denote the target destination for tutor Kuri. Tutor Kuri only waved to the user when it first arrived and when it was about to leave, as defined by the within-subjects study design conditions (Sec. IV).

### C. Design Considerations Implemented from Pilot Studies

MoveToCode was designed through a series of pilot studies that allowed for adjustments based on user feedback. The design process began with pilot study I ($n = 10$) with college students and the original language headset design [21]. Next we performed pilot study II ($n = 5$) with Ph.D. and Master's students from our research lab, who tested a console-based version of MoveToCode adapted for tablets, with the AR tutor Kuri added. The exercises were then redesigned to focus on programming baby Kuri through a maze, using tangible pieces of paper to ground the experience. A final pilot study (pilot study III) was conducted with Los Angeles elementary school students ($n = 15$) aged 8-12 who were part of an after school robotics club. The final design was a combination of lessons learned from these pilot studies and existing design conventions drawn from Google's Augmented Reality User Experience Design Guidelines [23]. The designs were then tested in our main study, which was conducted in Los Angeles elementary school classrooms, as described in Sec. IV.

In pilot study III, 8 students identified their gender as female and 7 as male. Their age ranged from 8-12 years old ($\bar{X} = 9.7, \sigma = 1.1$). Identified ethnicity of the students was Asian + White : 3, White: 3, Hispanic origin + Asian: 2, Asian: 2, Hispanic origin + White: 1, Black/African American + Asian + White: 1, and preferred not to specify: 3. Prior coding experience included Scratch : 9, Scratch + BotBall (robotics): 4, and Scratch + Code.org : 2.

*1) Existing Design Conventions:* We identified AR-specific design conventions used in other AR applications, and observations and feedback from our pilot studies supported those conventions.

One important design consideration was the difficulty users had in understanding how far away a virtual object was from them, i.e., the z-depth of virtual objects. To address this issue, we added shader-based shadows to objects (Fig. 3) to communicate depth [24]. Another useful design convention was the use of context-aware arrows to reference off-screen objects of interest (Fig. 3.C), such as code blocks and tutor Kuri. The arrows functioned as non-anthropomorphic deictic gestures, aimed at quickly drawing the user's attention to an object [25], [26]. Finally, we found that users struggled with manual rotation of 3D objects, such as code blocks. To address this issue, we implemented a feature that locked the rotation of objects to the user when being manipulated. This feature, based on guidelines from Google's Augmented Reality User Experience Design Guidelines [23], helped to improve system usability.

*2) New Design Considerations:* New design solutions emerged in the pilot studies, focused on two categories: 1) the use of physical pieces of paper and 2) the 3D code blocks.

The use of a physical medium that connected learning between the virtual and physical worlds is related to tangible programming languages [18]. We chose paper over custom-made or 3D printed objects to improve accessibility for real-world classroom studies. The physical paper provided a defined role for a second student, eliminated the need for one student to hold the mobile device while the other watched, and provided physical anchoring points for virtual content. Further, the paper anchors served as spawning reference points for virtual objects, such as code blocks and virtual maze pieces, and naturally defined the AR play area for any physical environment. AR experiences need to account for many possible physical domains [23], from a large convention center to a cramped room. This means a play area needs to account for different environments and to adjust the virtual content to avoid object clipping and unreachable objects. The physical pieces of paper naturally restricted the play area as they could only be placed in real world locations.

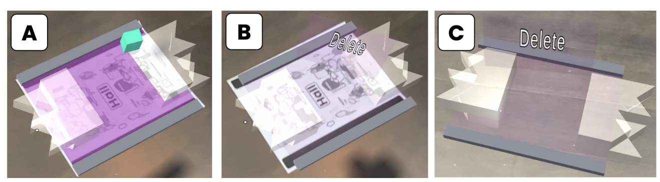


Fig. 5: All possible states of tracking a piece of maze paper. **A) Tracking & In View**: virtual analog is overlaid with spinning tracking indicator cube; **B) Not Tracking & In View**: virtual analog persists having higher transparency, removing the spinning indicator cube, and adding a delete button; and **C) Not Tracking & Not In View**: virtual analog is identical to B.

Tracking pieces of paper presented a new challenge (Fig. 5): when the maze paper was tracked by the mobile device, a virtual analog was positioned exactly where the paper was. However, when the paper stopped being tracked, the virtual analog persisted, confusing users. The persistence was necessary for mazes that required a large number of pieces, as the mobile device might not be able to track them all at once even if the physical pieces were all in the camera frame. To address user confusion, we first added spinning tracking indicators (Fig. 5.A) to the virtual analogs when the paper was being tracked, and made them disappear when tracking ended. However, users remained confused. We then made the virtual analogs heavily translucent ($\alpha \approx 11.7\%$) when compared to the tracking ($\alpha \approx 54.9\%$). This change resulted in no more participant confusion (as measured by the number of participant questions about tracking) in the full study described in Sec. IV.

We designed 3D code blocks using tangible and virtual programming. The benefits of tangible programming language interfaces, such as those used in AR settings, are well-established [27]. These interfaces allow users to interact with code in a tactile, spatial, and persistent manner, as the physical pieces can be placed in the environment without taking up space on the mobile device interface. In contrast, 2D block-based coding interfaces offer unlimited supplies of blocks and ease of rearrangement, but take up a large portion of mobile device screen real estate and are tied to the device reference frame. We created 3D code blocks for MoveToCode that offered the benefits of both virtual blocks (i.e., ease of spawning, dynamic expansion, and repositioning) and tangibles (i.e., persistence in the 3D environment and spatial grounding).

Connecting code blocks presented a new challenge, as users struggled with z-depth manipulation. To address this, we implemented a system that cast a ray from the user's grab point to the object, and treated the object as part of the code block's collision detection (i.e., "hit box"), effectively extending it along an infinite z plane relative to the user. Combined with the design choices outlined in Section III-C.1, this approach simplified user interactions with the system.

The final design considerations involved robot tutor actions (listed in Sec. III-B). Pilot testing identified that users wished to interact with the robot (leading to the interactive high-five) but also that users got frustrated if the robot was in the way and interacted with them too often. For example, in pilot study III, conducted in classroom, students did not always had a positive opinion about Kuri, by using statements such as:"get out of the way Kuri" and similar phrases. When interviewed, they mentioned wanting to "move Kuri out of the way", have Kuri "go above the ceiling", and have Kuri "go outside." This led us to introduce the "move out of the way of the user" action.

Finally, we also found that holding mobile devices vertically allowed for longer use compared to holding them horizontally, informing our final, vertical design.

## IV. Study Design

### A. Hypotheses

**H1:** Comparing the pilot study III with the main study, the design decisions described in Sec. III-C will increase:
- **A:** ***Perceived Robot Helpfulness***
- **B:** Number of completed exercises

**H2:** The presence of a virtual robot tutor, compared to no virtual robot tutor, increases the amount of time the user looks at the tutor robot or tutor dialogue.

**H3:** Comparing post-interaction to pre-interaction, students will indicate an increase in:
- **A:** ***Interest in Programming***
- **B:** ***Future Intention to Program***

### B. Recruitment and Participants

This study was approved by the University's Institutional Review Board (IRB #UP-20-00030). A recruitment flyer was sent to a list of local public and private schools. Inclusion criteria for the study were students 7-13 years of age proficient in English. Two teachers from two different schools responded and scheduled 1-hour study sessions.

Twenty-one students participated. Before each study, legal guardian consent and child assent were obtained for all participating students.

Fifteen identified their gender as male, 5 female, and 1 preferred not to specify. Their ages ranged from 9 to 10 years old ($\bar{X} = 9.5, \sigma = 0.5$). Students' self-identified ethnicities were: of Hispanic origin : 13, Black/African American + Asian: 2, Black/African American + Hispanic origin: 1, Hispanic origin + White: 1, Black/African American: 1, Asian + Middle Eastern or North African: 1, and preferred not to specify: 2. Prior coding experiences included Code.org: 9, Scratch + Code.org: 3, Scratch: 2, Scratch + Code + Roblox: 1, Robotics: 1, Other (not specified): 2, and None: 3.

*1) Measurements:* Quantitative Questions – To measure student curiosity, we used the standard question generation task [28] in which the students are prompted to ask as many questions about a topic without providing answers. This task has been used in relevant research, including measuring child curiosity after interacting with a social robot [14]. We instructed the students to write down as many questions as they could after the briefing section to avoid any biases from the task or the questionnaires. As a pre-test they generated questions before the task, and as a post test, to measure if they increased their curiosity about programming, after the task.

Qualitative Questionnaire – We constructed a questionnaire based on existing validated and reliable questionnaires and evaluated it for age appropriateness by two independent teachers. The pre-test questionnaire had two parts. The first part surveyed the students' attitudes, separated into two major thematic areas: i) ***Interest in Programming*** construct which had 13 items and evaluated the students' interest in programming, based on [29] questionnaire, and ii) ***Future Intention to Program***, which had 3 items and evaluated students' intention and motivation to follow a future career in programming, subset of the STIMEY Horizon Project questionnaire [30]. Students evaluated them on a Likert scale from "Strongly Disagree" to "Strongly Agree". The second part consisted of four additional demographic questions regarding students' age, ethnicity, gender identity, and prior experience with programming. The post-test questionnaire had the same questions as the pre-test and one more thematic area, ***Perceived Robot Helpfulness***, consisting of four items based on the age-appropriate usability scale [31].

The validity of the questionnaire was tested by a multidisciplinary group of engineers and psychologists with Lawshe's subject-matter expert rating methodology (SMEs).

The Content Validity Ratio (CVR critical) of the questionnaire was acceptable for five experts at .99, with two-tailed $p$=.01 [32]. Finally, we used age-appropriate fonts and characters in the graphical user interface.

### C. Procedure

Students were first given the pre-survey that included demographic questions, curiosity in programming, and intention to program (described in Sec. IV-B.1). At the beginning, students were shown a video of MoveToCode (`https://youtu.be/6CMuADWboD8`). However, some students did not fully understand the procedure, and therefore we replaced the video with a live demonstration

After the demo phase, students were given 5 minutes to write all questions they had. The teacher assigned the pairs of students, while there was a group consisted by 3 students. Working in dyads in a supporting environment enhances student problem-solving skills, teaches them to manage difficulties, and communicating with their peers increases their belief in their own capabilities [33], [34]. Although collaboration can be challenging depending on student social and cognitive skills [35], working in a supportive environment enhances engagement in STEM activities [36]. We asked the teachers to pair up the students so that they members of each dyad had matched capabilities as much as was possible.

Each dyad was given a tablet and maze papers. We used the $9^{th}$ generation Apple 10.2-inch iPads as Apple has 52% of the United States market share for tablets as of 2022 [37]. Each group was assigned a work area (e.g., Fig. 2). Half of the groups were randomly assigned to start the MoveToCode activity with the AR robot tutor Kuri model being visible (Condition A), or not visible (Condition B), for counterbalance purposes.

The activity, described in Sec. III-A, lasted 20 minutes and 22 seconds, consisting of the following:

- 6 seconds allowing the iPad to scan the room geometry
- 10 minutes 8 seconds condition A
- 10 minutes 8 seconds condition B

The tutor Kuri dialoM2Ce box (Fig. 1.D) was visible in both conditions. The conditions were randomly counterbalanced. The action policy described in Sec. III-B was used in both conditions, with the only difference being the visibility of the AR robot tutor's body and arms. At the beginning of each condition, tutor Kuri either waved and said "hello" or waved and said "goodbye" for 8 seconds at the end.

The MoveToCode application running on the tablet automatically closed after the end of the activity. Although all students were instructed that this would happen, all groups nonetheless restarted the app. They were then instructed to stop and returned to their seats. Students were then given a post-survey and then asked to write down questions for 5 minutes.

### D. Data Analysis

To evaluate the effectiveness of the design decisions outlined in Section III-A, we compared the pilot study III ($n =$ 21) with the main classroom study ($n = 21$). We measured the students' scores on ***Perceived Robot Helpfulness*** by conducting a two-sided Mann-Whitney U test. Additionally, we analyzed behavioral data collected at a rate of 50Hz, i.e., every 0.02 seconds, and included the number of exercises reached by both the main study groups and pilot study III groups. Further, we compared the amount of time spent looking at tutor Kuri or the tutor dialogue box within the main study groups, as shown in Figure 1.D. To analyze ***Interest in Programming*** and ***Future Intention to Program*** we compared the pre- and post-interaction survey data using a two-sided Wilcoxon signed-rank test. We report Cliff's delta ($\delta$) for effect size.

We qualitatively coded student written open-ended questions by first reading through all questions, creating categories, and then categorizing each question. The question categories were: 1) ***Robots*** – curious about the robot; 2) ***Programming*** – curious about programming; 3) ***Research*** – questions pertaining to the researchers; and 4) ***Repetition*** – asking about being able to do the activity again. A subset of questions qualified and counted as multiple categories (e.g., "How do you code robots?").

## V. Results

### A. Perceived Robot Helpfulness

Robot helpfulness scores were compared between the pilot study III ($n = 15$) to the main study ($n = 21$) with individual scores plotted in Fig. 6. Two-sided Mann-Whitney tests indicated a significant increase in ***Perceived Robot Helpfulness*** between the main study classroom ($Mdn = 4.25$) and pilot study III ($Mdn = 3.0$) conditions ($U = 283.0, p < .001, \delta =$ $.797$). This supports **H1.A** indicating design changes from pilot study III to the main study described in Sec. III-C.

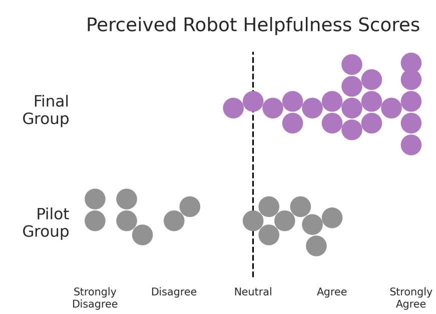


Fig. 6: Perceived robot helpfulness of the pilot study III plotted with the main classroom study. The shown scores are an average of 4 perceived robot helpfulness items described in Sec. IV-B.1.

### B. Behavioral Data

The number of exercises reached by each pair from the pilot study III group ($Mdn = 5$) to the main study ($Mdn =$ 6) groups is plotted in Fig. 7. The pilot group size ($n = 7$ pairs) was too small for statistical tests to evaluate **H1.B**.

We measured the total amount of time spent looking at tutor Kuri or the tutor dialogue (Fig. 8), measured by

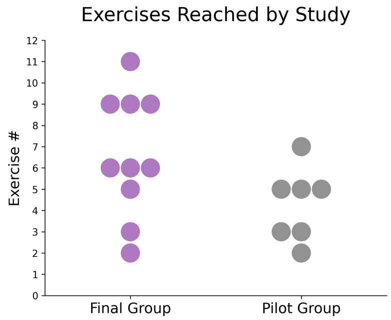


Fig. 7: Number of exercises reached by each study group at the end of the exercises.

casting a ray each time step from the center of the iPad with the first colliding object recorded. Collision boxes remained active with meshes turned off in the no robot condition. Ray collisions with the boxes were counted to not heavily favor the visible robot condition by merely having a larger target area. Seven groups looked at the robot or dialogue box when the robot mesh was visible and two groups looked more when the robot mesh was invisible. One group recorded less than 1 second of looking at tutor Kuri and the dialogue box. The sample size is too small to conduct statistical tests to evaluate **H2**.

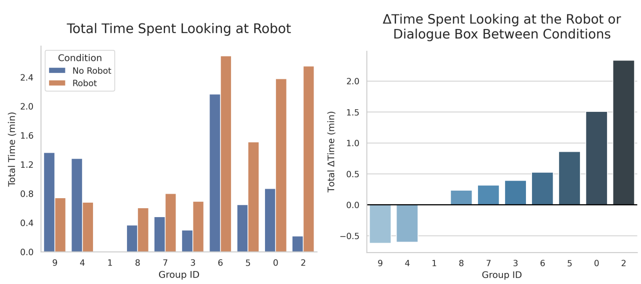


Fig. 8: **Left**:Time spent looking at the robot or dialogue box between conditions, sorted by difference in time of the Robot vs. No Robot conditions. **Right**: The sorted difference in time spent looking at the robot or dialogue box in the robot condition and the robot or dialogue box when the robot was not visible.

### C. Interest and Future Intention to Program

Interest in programming and future intention to program were compared between pre- and post-interaction survey responses. Two-sided Wilcoxon signed-rank tests indicated no significant increases in ***Interest in Programming*** between post-interaction ($Mdn = 4.31$) and pre-interaction ($Mdn = 4.07$) surveys ($z = 128, p = .390$). A Wilcoxon signed-rank test indicated no significant increases in ***Future Intention to Program*** between post-interaction ($Mdn = 3.67$) and pre-interaction ($Mdn = 3.33$) surveys ($z = 49.5, p = .849$). Thus neither **H3.A** nor **H3.B** are supported.

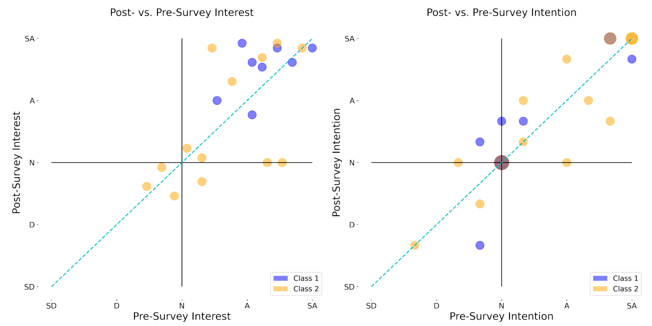


Fig. 9: **Left**: Curiosity in programming. **Right**: Intention to pursue programming further. Axes of each graph are from "Strongly Disagree" to "Strongly Agree". Scores above the diagonal line indicate higher post scores when compared to pre. Dot size is relative to the number of score occurrences.

### D. Pre-Post Student Questions

Question categories and counts are shown in Table I. Nine of the 21 students generated questions during the pre-test phase; the total number of written questions was 22. During the post-test phase, both the number of students who wrote questions and the total number of questions increased: 15 students generated 36 questions.

| **Category** | **Pre Total** | **Post Total** | **% of Pre** | **% of Post** |
|---|---|---|---|---|
| Robot | 9 | 12 | 40.9% | 33.3% |
| Programming | 6 | 20 | 27.3% | 55.6% |
| Research | 9 | 9 | 40.9% | 25.0% |
| Repetition | 1 | 11 | 4.5% | 30.6% |

TABLE I: Students' question generation per category for the pre-interaction (22) and post-interaction (36) question writing sessions. The percentages calculated relative to the total questions asked within that session (e.g., $\frac{9}{22} = 40.9\%$).

Example questions asked include the following:

- "How did the robot move?"
- "Does the robot have emotions?"
- "How old are you, do you code for a job?"
- "Do you like this career?"
- "Can we do more coding?"
- "Can we expect more of this in the future?"
- "Will you have a different program if we see you again?"

## VI. Discussion

We developed MoveToCode, an open-source, embodied learning visual programming language to increase young students' curiosity in programming. Our goal was to leverage advances in VAM-HRI to support SAR tutors. MoveToCode utilized an augmented reality autonomous robot tutor named Kuri, which responded to students' kinesthetic curiosity and promoted their interest in programming, as showed by the number of questions they generated regarding programming after the end of the task. Our design decisions were informed by a series of pilot studies and validated in local Los Angeles elementary school classrooms.

Although there were no significant changes in pre-post student curiosity or intention to program later in life, the student participants wrote more open-ended questions post-study and showed an improvement in perceived robot helpfulness and the number of completed exercises. Student participants generated almost twice as many questions after using MoveToCode, demonstrating a significant increase in their curiosity. The students asked questions about research and academic life after meeting the researchers and completing the programming task, indicating that the interaction raised their curiosity about STEM and research.

This work demonstrates the potential of using VAM-HRI in a kinesthetic context for SAR tutors, highlighting the existing conventions and new design considerations for creating AR applications for SAR.

## References


[1] L. Robert, "Personality in the human robot interaction literature: A review and brief critique," in *Robert, LP (2018). Personality in the Human Robot Interaction Literature: A Review and Brief Critique, Proceedings of the 24th Americas Conference on Information Systems, Aug*, pp. 16–18, 2018.

[2] G. Brajnik and S. Gabrielli, "A review of online advertising effects on the user experience," *International Journal of Human-Computer Interaction*, vol. 26, no. 10, pp. 971–997, 2010.

[3] K. Yoshino and S. Zhang, "Evaluation of teaching assistant robot for programming classes," *Int. J. Inf. Educ. Technol.*, vol. 10, no. 5, pp. 327–334, 2020.

[4] M. Walker, T. Phung, T. Chakraborti, T. Williams, and D. Szafir, "Virtual, augmented, and mixed reality for human-robot interaction: A survey and virtual design element taxonomy," *arXiv preprint arXiv:2202.11249*, 2022.

[5] T. Groechel, M. Walker, C. T. Chang, E. Rosen, and J. Forde, "A tool for organizing key characteristics of virtual, augmented, and mixed reality for human-robot interaction systems," *IEEE Robotics & Automation Magazine*, 2022.

[6] A. Y. Nee and S.-K. Ong, "Virtual and augmented reality applications in manufacturing," *IFAC proceedings volumes*, vol. 46, no. 9, pp. 15–26, 2013.

[7] J. D. Bric, D. C. Lumbard, M. J. Frelich, and J. C. Gould, "Current state of virtual reality simulation in robotic surgery training: a review," *Surgical endoscopy*, vol. 30, no. 6, pp. 2169–2178, 2016.

[8] B. Ikeda and D. Szafir, "Advancing the design of visual debugging tools for roboticists," in *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 195–204, IEEE, 2022.

[9] Y. B. Kafai and V. Vasudevan, "Constructionist gaming beyond the screen: Middle school students' crafting and computing of touchpads, board games, and controllers," in *Proceedings of the workshop in primary and secondary computing education*, pp. 49–54, 2015.

[10] M. N. Giannakos and L. Jaccheri, "From players to makers: An empirical examination of factors that affect creative game development," *International Journal of Child-Computer Interaction*, vol. 18, 2018.

[11] J. Robertson and C. Howells, "Computer game design: Opportunities for successful learning," *Computers & Education*, vol. 50, no. 2, pp. 559–578, 2008.

[12] J.-M. Sáez-López, M. Román-González, and E. Vázquez-Cano, "Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools," *Computers & Education*, vol. 97, pp. 129–141, 2016.

[13] J. Denner, L. Werner, and E. Ortiz, "Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?," *Computers & Education*, vol. 58, no. 1, pp. 240–249, 2012.

[14] G. Gordon, C. Breazeal, and S. Engel, "Can children catch curiosity from a social robot?," in *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 91–98, IEEE, 2015.

[15] M. Shiomi, T. Kanda, I. Howley, K. Hayashi, and N. Hagita, "Can a social robot stimulate science curiosity in classrooms?," *International Journal of Social Robotics*, vol. 7, pp. 641–652, 2015.

[16] C. Pasalidou and N. Fachantidis, "Designing a shared workspace for learning using augmented reality and social robots," in *HCI International 2022 Posters: 24th International Conference on Human-Computer Interaction, HCII 2022, Virtual Event, June 26–July 1, 2022, Proceedings, Part III*, pp. 80–87, Springer, 2022.

[17] A. Sousa Calepso, N. Hube, N. Berenguel Senn, V. Brandt, and M. Sedlmair, "cardlearner: Using expressive virtual agents when learning vocabulary in augmented reality," in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 2022.

[18] T. Sapounidis and S. Demetriadis, "Educational robots driven by tangible programming languages: A review on the field," in *International Conference EduRobotics 2016*, pp. 205–214, Springer, 2017.

[19] T. Groechel, Z. Shi, R. Pakkar, and M. J. Matarić, "Using socially expressive mixed reality arms for enhancing low-expressivity robots," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1–8, IEEE, 2019.

[20] M. Guenaga, A. Eguíluz, P. Garaizar, and J. Gibaja, "How do students develop computational thinking? assessing early programmers in a maze-based online game," *Computer Science Education*, vol. 31, no. 2, pp. 259–289, 2021.

[21] T. Groechel, R. Pakkar, R. Dasgupta, C. Kuo, H. Lee, J. Cordero, K. Mahajan, and M. J. Matarić, "Kinesthetic curiosity: Towards personalized embodied learning with a robot tutor teaching programming in mixed reality," in *International Symposium on Experimental Robotics*, pp. 245–252, Springer, 2021.

[22] Q. Wu and C. Miao, "Curiosity: From psychology to computation," *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, pp. 1–26, 2013.

[23] "Augmented reality design guidelines - google developers."

[24] C. Diaz, M. Walker, D. A. Szafir, and D. Szafir, "Designing for depth perceptions in augmented reality," in *2017 IEEE international symposium on mixed and augmented reality (ISMAR)*, pp. 111–122, IEEE, 2017.

[25] I. Goktan, K. Ly, T. R. Groechel, and M. J. Mataric, "Augmented reality appendages for robots: Design considerations and recommendations for maximizing social and functional perception," *arXiv preprint arXiv:2205.06747*, 2022.

[26] L. Brown, J. Hamilton, Z. Han, A. Phan, T. Phung, E. Hansen, N. Tran, and T. Williams, "Best of both worlds? combining different forms of mixed reality deictic gestures," *ACM Transactions on Human-Robot Interaction*, 2022.

[27] K. Hattori and T. Hirai, "An intuitive and educational programming tool with tangible blocks and ar," in *ACM SIGGRAPH*, 2019.

[28] P. L. Harris, *Trusting what you're told: How children learn from others*. Harvard University Press, 2012.

[29] D. Gul, I. Cetin, and M. Y. Ozden, "A scale for measuring middle school students' attitudes toward programming," *Computer Applications in Engineering Education*, vol. 30, no. 1, pp. 251–258, 2022.

[30] A.-M. Velentza, S. Ioannidis, and N. Fachantidis, "Service robot teaching assistant in school class-room," pp. 12115–12117, 2020.

[31] C. Putnam, M. Puthenmadom, M. A. Cuerdo, W. Wang, and N. Paul, "Adaptation of the system usability scale for user testing with children," in *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, pp. 1–7, 2020.

[32] F. R. Wilson, W. Pan, and D. A. Schumsky, "Recalculation of the critical values for lawshe's content validity ratio," *Measurement and evaluation in counseling and development*, vol. 45, no. 3, 2012.

[33] N. A. Çakır, A. Gass, A. Foster, and F. J. Lee, "Development of a game-design workshop to promote young girls' interest towards computing through identity exploration," *Computers & Education*, vol. 108, pp. 115–130, 2017.

[34] D. C. Carvajal-Ayala and R. A. Avendaño-Franco, "Implementing lesson plans for collaborative learning with children in an efl context.," *GIST Education and Learning Research Journal*, vol. 22, 2021.

[35] J. Laru, S. Järvelä, and R. B. Clariana, "Supporting collaborative inquiry during a biology field trip with mobile peer-to-peer tools for learning: a case study with k-12 learners," *Interactive Learning Environments*, vol. 20, no. 2, pp. 103–117, 2012.

[36] A. Puvirajah, G. Verma, and T. Campbell, "Advancing minoritized learners' stem oriented communication competency through a science center-based summer program," *Journal of Museum Education*, vol. 45, no. 4, pp. 437–449, 2020.

[37] "Tablet vendor market share united states of america."