Programming Embodied Interactions with a Remotely Controlled Educational Robot

ALEXANDROS MERKOURIS and KONSTANTINOS CHORIANOPOULOS,
Ionian University

Contemporary research has explored educational robotics, but it has not examined the development of computational thinking in the context of programming embodied interactions. Apart from the goal of the robot and how the robot will interact with its environment, another important aspect that should be taken into consideration is whether and how the user will physically interact with the robot. We recruited 36 middle school students to participate in a six-session robotics curriculum in an attempt to expand their learning in computational thinking. Participants were asked to develop interfaces for the remote control of a robot using diverse interaction styles from low-level to high-level embodiment, such as touch, speech, and hand and full-body gestures. We measured students' perception of computing, examined their computational practices, and assessed the development of their computational thinking skills by analyzing the sophistication of the projects they created during a problem-solving task. We found that students who programmed combinations of low embodiment interfaces or interfaces with no embodiment produced more sophisticated projects and adopted more sophisticated computational practices compared to those who programmed full-body interfaces. These findings suggest that there might be a tradeoff between the appeal and the cognitive benefit of rich embodied interaction with a remotely controlled robot. In further work, educational robotics research and competitions might be complemented with a hybrid approach that blends the traditional autonomous robot movement with student enactment.

CCS Concepts: • **Social and professional topics → Computational thinking**; **Student assessment**; **K-12 education**;

Additional Key Words and Phrases: Embodied learning, educational robotics, embodiment, children, human–robot interaction, computational thinking, assessment

ACM Reference format:
Alexandros Merkouris and Konstantinos Chorianopoulos. 2019. Programming Embodied Interactions with a Remotely Controlled Educational Robot. *ACM Trans. Comput. Educ.* 19, 4, Article 40 (July 2019), 19 pages.
https://doi.org/10.1145/3336126

1 INTRODUCTION

In recent years, researchers and educators have considered robotics as an inspiring educational tool to promote the comprehension of science, technology, engineering, and mathematics (STEM) concepts [6, 18] as well as to foster computational thinking (CT) [8] and creativity. Contemporary research has introduced robots in the classroom [40], but it has not considered the effects of alternative embodied interactions with them. In a typical educational robotics activity, children are

Author's addresses: A. Merkouris and K. Chorianopoulos, Department of Informatics, Ionian University, 7 Tsirigoti Square, Corfu, Greece; emails: {c14merk, choko}@ionio.gr.

1946-6226/2019/07-ART40 $15.00
https://doi.org/10.1145/3336126

asked to enliven the robots by creating the appropriate computer programs [7]. The programmer has to think mainly about the goal of the robot and how the robot will interact with the environment. However, there is another important aspect that should also be taken into consideration, and this is if and how the user will physically interact with the robot. In particular, Alimisis [2] points out that embodiment is an innovative approach to make robotic activities more approachable and meaningful to children. According to Dourish [17] embodiment "is the property of our engagement with the world that allows us to make it meaningful."

Additionally, with the rapid development of digital technologies, such as mobile devices, touchscreens, and computer vision, a wide gamut of interfaces is provided to users. Children can interact with digital information more naturally and physically [30], using personal devices that are appealing to them. Putting forth the notion of "embodied interaction" [17], we are moving away from the conventional keyboard and mouse input devices to touch, speech, and hand and full-body interfaces [36]. The user is allowed to act directly in the physical world, as the human body becomes the input for the interaction.

Recently, there has been a strong push to exploit these interfaces in science and computing education triggered by the views of embodied cognition researchers that physical interactions with the environment through sensorimotor modalities (touch, movement, speech, smell, and vision) are essential factors in the construction of knowledge [4, 23, 59]. We are motivated by embodied learning findings that regard a broad spectrum of human motor-perceptual skills, which reach beyond the traditional desktop metaphor and keyboard-mouse as input devices. For this reason, we set out to investigate whether various programming activities to control a robot using diverse interaction modalities, with a different level of embodiment, can affect students in exploring CT concepts. One factor that determines the level of embodiment is the degree of bodily movement activity. For instance, full-body movement entails a higher level of embodiment compared to hand micro-movements [31, 37].

The vision of Papert [49] and Kay [3] for introducing powerful ideas (math and science concepts) through programming was the main inspiration for creating the intervention. Expanding their views "beyond the screen" by targeting a robot is one aspect of our study. Another aspect concerns the dimension of embodiment and its connection to CT performance. Similarly to Kafai et al. [32], we studied and assessed the development of CT by applying Brennan and Resnick's framework [9]. Our research questions centered on these major topics:

- *Intention*: Did the robotics curriculum have any influence on students' perception of computing?
- *Comprehension*: Were there any differences in the development of students' CT skills that could be attributed to the different levels of embodiment?

Therefore, the primary contribution of our research is studying alternative types of human–robot interaction in the contexts of embodied learning and computing education. The rest of the article is structured as follows: In the next session, we present the related work; in Section 3, we describe the methodology; in Section 4, we present the results; in Section 5, we discuss the findings, implications, and limitations; and, finally, we summarize the conclusion and future work in Section 6.

2 RELATED WORK

2.1 Embodied Learning

Existential phenomenologists such as Heidegger [28], Merleau-Ponty [41], and neurologists such as Damasio [14] reject Descartes's dualism of mind and body, arguing that "thinking does not occur separately from being and acting" [17]. Based on this premise, theories of embodied cognition emphasize [4, 23, 59] the importance of perception in conceptual learning by suggesting that knowledge is intimately tied to sensorimotor actions. The mind no longer has been treated as separate from the body, and it is perceptual rich experiences that shape cognitive processes and allow individuals to construct meaning and understanding of the world [17].

An issue is how precisely perceptual rich experiences contribute to knowledge. Evidence about the mechanisms underlying embodied learning can be drawn from the theories of working memory and cognitive load [63]. It is thought that not only each sensorimotor modality (visual, auditory, and tactile) has its working memory [42] but also acts as an individual source of perceptual experiences [26]. Specifically, when multiple modalities are employed stronger memory traces are produced, and more abundant knowledge structures are created, compared to the use of a single modality. Hence, learners would be able to retrieve the multimodal knowledge representations more efficiently in the future. Second, by combining the tactile channel with the visual and auditory ones, the mental energy required to process a given amount of information is distributed across the modalities, and thus the cognitive load imposed to the learner is reduced. In sum, perceptual rich experiences not only may help individuals learn conceptual content faster and easier but also in a more in-depth manner.

Educational and developmental learning theories have also acknowledged the significance of sensory and motor actions of the human system in the learning process and the construction of knowledge [15]. For example, Maria Montessori [45] believed that through movement learners interact with the environment, and it is through these interactions that they eventually acquire even abstract ideas. From a theoretical perspective, embodied learning is also related to learning theories favoring hands-on activities and child interaction. According to Piaget [51] and Papert [48], a fundamental tenet of learning is people's actions, as they construct knowledge and form the meaning of the world by actively interacting with learning objects.

However, what distinguishes embodied learning from other hands-on learning theories is the dimension of "gestural congruency" [36]; that is, in order a perceptually rich learning experience to be effective, actions of the body need to be congruent to the mental operations and representations of the concepts to be learned [31, 46, 53]. A representative example that highlights the significance of "gesture congruency" is Johnson-Glenberg's et al. [31] study for learning about centripetal force. Specifically, having participants swing a trackable object overhead instead of using a mouse interface to control the simulation is a movement that maps coherently onto the learning domain [37] but also coincides with real-life experiences [31]. It is essential, therefore, to consider not only methods that make use of physical interactions but also how meaningful are the types of student interactions to the subject matter.

2.2 Level of Embodiment

The embodied approach [52] has been used to cover the learning of abstract materials in a wide range of topics that extend from science [26, 31, 35, 37], technology, engineering, and mathematics [1, 57] to CT [13, 21, 50].

Johnson-Glenberg et al. [31] defined four levels of embodiment, comprising three factors: sensorimotor engagement, gestural congruency, and perceived immersion. According to their proposed taxonomy, in the first level, all three factors are low, so the learner usually observes the learning material in a desktop or tablet computer with minor sensorimotor engagement (mouse-clicking or key-pressing). The next level requires a higher degree of motoric engagement such as movement of the arm or fingers. In the third level, the full-body can act as input for the interaction, but the user remains in one place, while the fourth and highest level involves full-body interaction with locomotion and a high degree of gestural congruency and immersion.

Computer simulations that make use of gestures and touch sensorimotor input [11, 26] have been considered as an innovative approach to supporting the teaching of abstract scientific concepts. Specifically, in a study conducted by Chan and Black [11] students investigated the functional relationship between the gravitational and kinetic energy through a roller coaster simulation. Participants assigned in the direct manipulation condition were asked to control the position-height of a roller coaster car and at the same observed the changes in its kinetic and potential energy. They demonstrated better recall, problem-solving and transfer abilities than the students assigned to less disembodied conditions who just watched the animation without user control. Similarly, Han and Black [26] used simulations augmented with haptic feedback to enhance elementary students' understanding of the movements of gears. Results of their study indicate that the augmented haptic simulations (force and kinesthetic and purely kinesthetic) provided richer perceptual experiences to students than the equivalent non-haptic simulation.

Activities involving a higher level of embodiment, such as full-body activity, also provide a strong foundation for fostering embodied learning as users can interact with the digital information through natural physical movement. The use of full-body interaction for learning physics principles, such as gravity force and planetary motion, through a mixed reality simulation, was the subject of research conducted by Lindgren et al. [37]. Results of the study indicate that students who used a full-body mixed reality simulation game obtained more knowledge about force and motion, showed higher levels of engagement, and more positive attitudes towards science compared to students who used the desktop version of the same simulation game. Similarly, Johnson-Glenberg et al. [31] used a mixed reality simulation to facilitate college-age participants' understanding of centripetal force. The authors found higher long-term learning gains in physics for the subjects assigned to the "high embodiment" condition (swinging a tangible trackable object overhead) compared to those assigned to the "low embodiment" condition (using a mouse as interaction tool). It is essential, therefore, to consider the role that level of embodiment plays in learning and its place in the embodied theory of education.

2.3 Computational Thinking and Embodiment within Robotics

Ever since Papert [48] introduced the concept of CT and until recently, CT [60] was regarded as a highly intellectual cognitive activity. In particular, Wing [61] described CT as "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent." Subsequently, Brennan and Resnick [9] broaden the CT term, which encompasses three dimensions: computational concepts, practices, and perspectives. Computational concepts refer to the fundamental elements that are commonly present in many programming environments such as sequences, loops, parallelism, events, conditionals, operators, and data. Computational practices are activities that may occur during the process of construction, such as experimenting and iterating, testing and debugging, reusing and remixing, and abstracting and modularizing. Brennan and Resnick [9] argued that "computational practices focus on the process of thinking and learning, moving beyond what you are learning to how you are learning." On the other hand, computational perspectives capture young learners' shifting viewpoints about themselves (expressing), their relations with others (connecting) and the world (questioning), as they engage in CT activities.

Some educators and researchers, embracing the embodied cognition view, believe that CT might be enhanced if it is channeled through rich perceptual experiences. A practical learning approach, referred to as direct embodiment, is to have students enact with their bodies the programming scripts before creating the program. For example, Fadjo [21] found that having students physically embody the actions presented in pre-defined instructional materials could positively affect the development of their CT skills as they created a video game in Scratch. Notably, Fadjo reported that the effect of physical embodiment on the development of CT skills became less pronounced as the scripts in the instructional materials became more complex. Surrogate embodiment, where learners manipulate and observe an external representative, is an alternative practice for developing programming skills [20]. For instance, Sung et al. [55] studied how direct and surrogate embodiment activities with a different level of embodiment (full vs. low embodiment) can improve students' mathematical understandings and programming skills. She found that the surrogate condition with full embodiment had the greatest impact on engagement and learning outcomes. Other scholars [13, 50] examined how embodied interactions in a virtual environment that processed students' dance movements could enhance computational learning.

Besides, using virtual environments [13] and visual programming tools such as Scratch [50, 55], a growing number of educators and researchers have considered educational robotics as a promising field for applying the embodied cognition view, mainly in the context of primary school education. Specifically, Lu et al. [38] examined how direct and surrogate bodily experiences in a robotic workshop can influence elementary students' understanding of programming concepts. Participants were asked to act out with their bodies (direct embodiment) or observe the teacher acting out (surrogate embodiment) the robot's movements and then program the robot to make the same moves. The results indicated that students assigned to the direct embodiment condition comprehended the programming concepts faster. Similarly, Sung et al. [56] and colleagues investigated how embodied experiences can affect lower elementary school students' problem-solving skills. Students, in the high embodiment condition, were asked to enact the robot's movements through full-body interaction before building and programming the robot, demonstrated better problem-solving skills than those in the low embodiment condition (using hand gestures).

Nevertheless, previous works have not considered computer programming and the development of CT in the context of embodied interaction with educational robotics. For this reason, we implemented educational robotics activities within a secondary education setting for studying the development of CT, but we adopted a different embodied learning approach compared to previous studies. Instead of asking students to enact the robots' moves before programming it, we asked them to program human–robot interfaces with a different level of embodiment. In this way, we expect to find a connection between embodiment and the development of CT skills and draw some conclusions about students' perception of computing.

3 METHODOLOGY

3.1 Subjects

We recruited 36 middle school students, aged between 14 and 15 years, with little to no prior programming experience to participate in a six-session robotics curriculum. The sample was White, relatively equally divided by sex (17 girls, 19 boys), and from lower to middle socioeconomic statuses. We randomly selected the participants from the third-level class of a middle school. Students worked in pairs in each of the activities. The criteria for matching the pairs of students were their skills and expertise and existing friendships. Thus, we created 15 same-gender and 3 mixed-gender pairs. Participants were not self-selected into the curriculum, as is the case in most after-school robotic workshops. Nevertheless, they were asked if they would like to participate in the study and their parents were informed and asked to give their permission by signing the necessary consent form.

3.2 Robotics Curriculum

The curriculum was divided into six individual sessions. In Table 1 we present what kind of applications students were asked to create and the CT concepts they explored during the sessions. The

Table 1. Overview of the Activities and the CT Concepts Introduced in Each Session of the Curriculum

Session	Activity Title	Students should create an application...	Computational Thinking Concepts
1	Touch Control	to control the robot with their fingers by touching their mobile phone screens	Events, Sequences, Data
2	Hand Control	to control the robot with hand gestures by rotating their mobile phone devices, utilizing the phone's orientation sensor	Events, Sequences, Data, Conditionals, Operators
3	Speech Control	to control the robot through speech commands, utilizing speech recognition technology	Events, Sequences, Data, Conditionals, Operators
4	Body Control	to control the robot with full-body gestures, using computer vision technology	Events, Sequences, Parallelism, Loops, Data, Conditionals, Operators
5	Line Follow	to integrate Artificial Intelligence to the robot so that it could move autonomously on the track following a black line	Events, Sequences, Data, Conditionals, Operators
6	Project	to navigate a robot on a fixed track and hit an object	Sequences, Loops, Events, Parallelism, Conditionals, Operators, Data

first five sessions followed a similar basic format: (1) Building the User Interface (UI). A template application and guiding instructions were given to students to add the necessary UI elements, (2) Programming the application's behavior, and (3) Going further by enhancing the basic application with additional features such as variable speed. In the final session, students applied the previously acquired programming knowledge to a semi-open problem-solving task. We asked them to create a program so that they could successfully navigate the robot on a fixed track and hit an object placed at a predefined spot with its robotic arm. No instructions were given to students on the final project, and they were prompted to program any interface (touch, hand or full-body gestures, speech, autonomous) they preferred. Moreover, they were allowed to "remix and reuse" [9] code from the previous sessions. Thus, the final project session followed a constructionist approach to learning and served as the condition for assessing learning outcomes of the intervention (Figure 1).

It should also be noted that each interface has certain affordances and limitations [9] and these may have influenced the balance between the activities. Nevertheless, we aimed to keep the balance between the different conditions but without controlling out their respective affordances [65].

The duration of each of the first five sessions was about 45 minutes while the final project activity lasted from 45 to 90 minutes. All activities were tested in a two-phase pilot study. Data collected during the pilot study helped us to refine the instructional material and the measuring instruments. The same researcher conducted both the preparation of the instructional material and the tutoring of the courses.

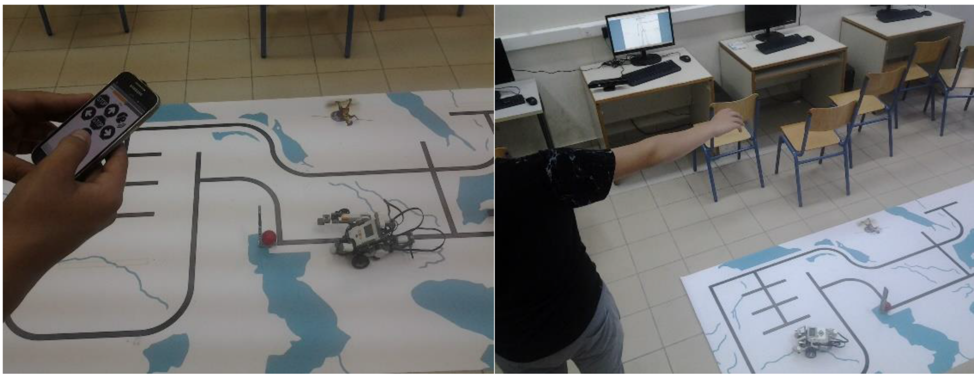


Fig. 1. Controlling a robot in the final session with touch and speech commands (left), and full-body gestures (right).

Table 2. Overview of the Interaction Modalities, Level of Embodiment, and the Development Platforms in Each Session of the Curriculum

Activities	Interaction Modalities	Level of Embodiment	Development Platform
Touch Control	Touch	First Level	App Inventor
Hand Control	Hand Gestures	Second Level	App Inventor
Speech Control	Speech	First Level	App Inventor
Body Control	Full-Body Gestures	Third Level	ScratchX
Line Follow	Artificial Intelligence	No Embodiment	App Inventor
Project	Students' Selections	Students' Selections	App Inventor or ScratchX

3.3 Materials

We employed App Inventor[1] [24] as the development platform in the sessions that involved mobile technology and students used their own mobile phones. For the session that involved full-body interaction, ScratchX[2] was employed as the development platform supported by the Kinect sensor for tracking the body [29].

The interaction modalities varied in the level of embodiment [31] (Table 2). Specifically, the modalities used in the Touch Control and Speech Control activities we can assume they belong in the first level, those in the Hand Control in the second level and those in the Body Control in the third level. The Line Follow is considered to be the activity with no embodiment.

The robots chosen for supporting the curriculum were Lego Mindstorms.[3] Both App Inventor and ScratchX programming environments have the potential to be used for programming the Lego robots, and this was the main reason for their selection. Although there were some differences in the layout (e.g., menus, tabs) of the visual blocks-based programming environments, the coding area was very similar and based on the idea of snapping blocks together.

The objective was to have students program human–robot interfaces in the desktop computer, then to carry out actions with a different level of embodiment, such as touch, speech commands, hand, and full-body gestures, and observe the consequences of their actions in the kinematics of a

[1] App Inventor: http://appinventor.mit.edu.
[2] ScratchX: http://scratchx.org/.
[3] Lego Mindstorms: https://www.lego.com/en-us/mindstorms.

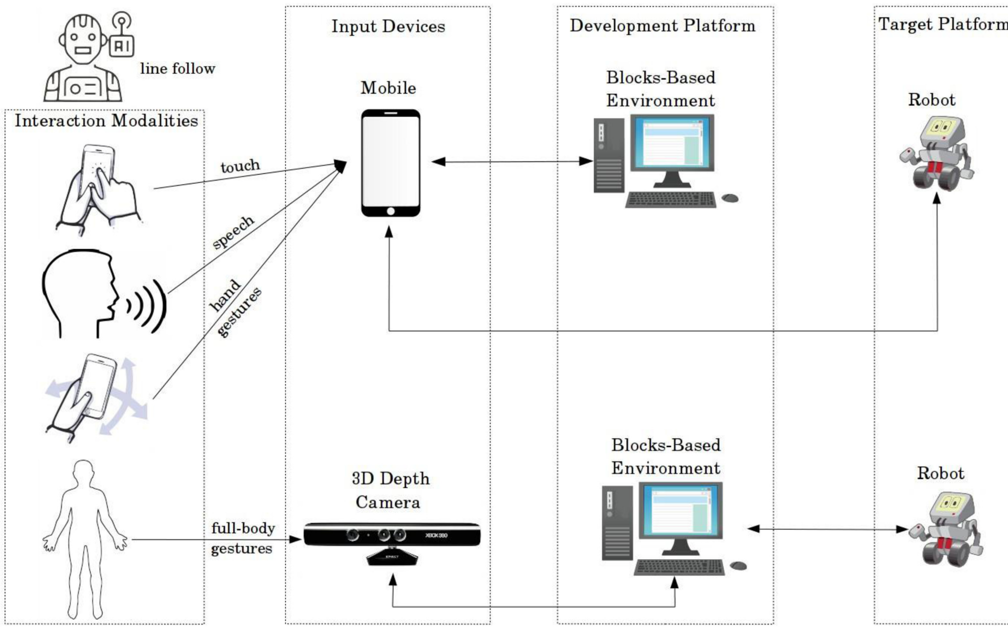


Fig. 2. Block diagram showing an overview of the interaction modalities, the input devices, the development platform, and the target platform.

robot (Figure 2). The spatial location of the output in relation to the input, referred to as mapping [39], was considered to be discrete as the actions for triggering the effect were performed either in the mobile phone or in free space (tracked by Kinect), separately from the target platform where the reaction took place. In other words, students controlled the robots through teleoperation. It should be emphasized that with the full-body interface students carried out actions directly in the physical world, as their bodies were the input for the interaction, while in the other cases they acted through another device (mobile phone).

3.4 Measuring Instruments and Data Analysis

For the study, we collected and analyzed both qualitative and quantitative data. Concerning the quantitative data, the students filled out brief pre-test and post-test questionnaires. The pre-test, before the programming activities, consisted of a five-level Likert questionnaire that recorded students' prior experience with programming, their perception of computing, robotics, and mobile development. The post-tests, after the programming activities, included a five-level Likert questionnaire that recorded a change of students' perceptions. In short, with these questionnaires, we aimed to capture the shift in participants' computational perspectives [9].

Regarding the qualitative data, we manually analyzed students' projects in the final session for assessing the development of CT. Here, we focused our analysis on the first dimension in Brennan and Resnick's [9] CT framework: computational concepts. The projects were graded based on a rubric used for grading student-made computer game projects [58]. The rubric was appropriately adjusted to fit the current intervention characteristics. According to Werner et al. [58], game programs are composed of programming constructs, pattern, and mechanics. Constructs are the elementary pieces of code that are accessible in a programming environment. When multiple programming constructs are combined programming patterns are created. Patterns are higher-level blocks that create additional program functionality. Additionally, when construct and pattern are put together, mechanics are formulated. Mechanics are defined as a variety of actions, behaviors, and control mechanisms used to support user interaction. By applying this framework in our study, we attempted to measure the correct use of programming constructs and patterns as the produced mechanics were limited to the robot navigation, the robotic arm control, and the power-speed control mechanisms. In the appendix, we provide some examples of what constitute block commands, constructs, patterns, and mechanics in the programs that the students created, since these are the primary metric of evaluation.

Given the fact that no single approach is sufficient [9] to have a more comprehensive and accurate view of students' learning outcomes we additionally employed a 30-minute plus semi-structured interview that gave participants a chance to describe not only their projects but also their experiences. They were encouraged to explain their projects and justify the selections they made, whether they would be interested in participating in programming activities in the future, what extensions did their experiences lead them to imagine, what kinds of problems did they run into, whether their ideas about computer science had changed, and whether the curriculum had influenced their future goals.

Finally, students' on-screen activity was recorded by Camtasia[4] capture to gain an overview of their computational practices, which constitute the second dimension in Brennan and Resnick's [9] CT framework. In one of the workstations, a webcam with a microphone was used for video and audio capture the participants. In the other workstation, a microphone was used to capture students' conversations. The transcripts from students' on-screen activity in the final session were coded by two coders. The first coder was one of the authors and the second was a computer science teacher with 12 years of teaching experience in middle school. An intraclass correlation coefficient was computed to assess the agreement between the coders. Results indicated that the inter-rater reliability was excellent: $ICC = 0.98$. The overall internal consistency was also excellent with a Cronbach's alpha value of $a = 0.98$. We employed Transana[5] software to transcribe, code, and analyze the interviews and the Camtasia captured data.

First, students filled out the pre-test questionnaire at their convenience and afterward, on different days, worked in pairs on each of the activities. After the completion of the curriculum, the post-test questionnaire was filled out, followed by the interview. Students answered the questionnaires individually and the questions in the interview in pairs.

4 RESULTS

We applied Brennan and Resnick's framework [9] for assessing the development of CT. Each of the three computational dimensions (concepts, practices, and perspectives) is discussed in detail in the following subsections.

4.1 Computational Concepts

First, we manually analyzed all projects in the final session by measuring the correct use of computational concepts and graded them according to the rubric described in Section 3.4.

In an attempt to evaluate the validity of the rubric we first examined the correlation coefficient between the number of total block commands, constructs, and patterns used. Table 3 shows the Pearson's correlation coefficient between the three variables. As expected, constructs were significantly correlated to the number of total blocks used, $r_b = 0.628$, $p = 0.005$. While patterns were significantly correlated to constructs, $r_b = 0.630$, $p = 0.005$, but not to the total blocks, $r_b = 0.362$,

[4]Camtasia: https://www.techsmith.com/video-editor.html.
[5]Transana: https://www.transana.com/.

Table 3. Correlation Coefficient between Total Blocks, Constructs, and Patterns

	Total Blocks	Constructs	Patterns
Total Blocks	1	0.628*** [0.149, 0.890]	0.362 ns [−0.186, 0.771]
Constructs	18	1	0.630*** [0.291, 0.836]
Patterns	18	18	1

ns = not significant ($p > 0.05$), $^*p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$. BCa bootstrap 95% CIs reported in brackets.

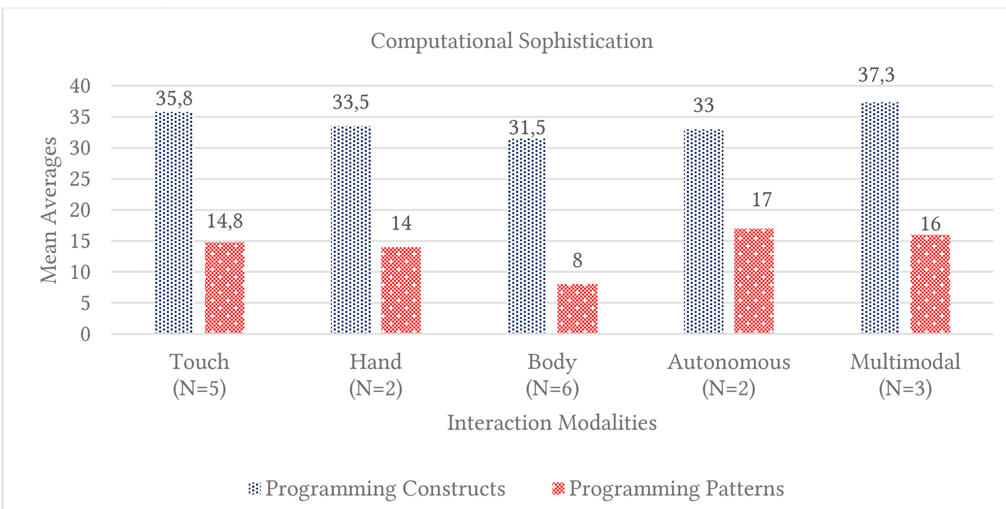


Fig. 3. Mean averages of the used programming constructs and pattern segregated according to the interaction modalities selected for controlling the heading of the robot.

$p = 0.14$. Using more block commands does not imply an increase in the numbers of patterns. The above can serve as an indication that the rubric was appropriate.

Afterward, we investigated the mean averages of the used programming constructs and patterns according to the interaction modalities selections that students made while dealing with the robot navigation programming mechanism (Figure 3). It should be noted that three groups of students chose multiple types of interactions (multimodal) for navigating the robot. The first group combined touch with hand gestures, while the other two groups combined artificial intelligence with touch.

The Kruskal–Wallis non-parametric test was used to assess statistical differences in constructs and patterns among the projects, due to the small and unequal sample size. According to the Kruskal–Wallis test, there was a statistically significant difference in patterns $H(4) = 13.15$, $p = 0.011$. Pairwise comparisons with adjusted p-values showed that the difference was significant between the projects that students used the full-body interface and the Autonomous projects ($p = 0.033$). Nevertheless, the above results should be read with caution, as the group sizes were unequal. For this reason, we also employed the Mann–Whitney non-parametric test for comparisons between the touchscreen interface projects and the full-body ones, as the sample groups were similar ($N = 5$ and $N = 6$ respectively). Constructs in the Touch projects ($Mdn = 35.80$) did

Table 4. Percentage of Time Spent in Each Computational Practice Segregated According to the Interaction Modalities

	Abstracting & Modularizing	**Coding**	**Experimenting & Iterating**	**Reusing & Remixing**	**Testing & Debugging**
Touch	25,5%	0,8%	15,6%	26,0%	32,1%
Hand	33,4%	8,6%	14,5%	22,5%	21,0%
Body	8,5%	25,1%	19,1%	16,8%	30,4%
Autonomous	25,3%	2,5%	34,7%	11,0%	26,5%
Multimodal	16,9%	16,4%	16,1%	20,2%	30,3%

not differ significantly from constructs in the Body projects ($Mdn = 31.50$), $U = 4.00$, $z = -2.04$, $p = 0.052$, $r = -0.61$. However, Patterns in the Touch projects ($Mdn = 14.80$) were significantly higher than patterns in the Body projects ($Mdn = 8.00$), $U = 0.00$, $z = -3.03$, $p = 0.004$, $r = -0.91$. In other words, higher embodiment levels led to projects with overall lower computational sophistication.

4.2 Computational Practices

We have also attempted to analyze students' computational practices by observing the problem-solving processes during the final session of the robotics curriculum. The transcribed on-screen activity of the eighteen groups was coded, by the two individual coders, according to the four computational practices listed by Brennan and Resnick [9] and the coding practice. We calculated the time spent in each computational practice, and we presented the average percentage duration of the two coders' judgments in Table 4.

The Kruskal–Wallis non-parametric test was used to assess statistical differences in the percentage of time devoted to each computational practice. According to the Kruskal–Wallis test, there was a statistically significant difference in the abstracting and modularizing practice $H(4) = 12.01,\ p = 0.017$. Pairwise comparisons showed that students who programmed the full-body interface spent significantly less time in abstracting and modularizing compare to those who programmed the touchscreen interface ($p = 0.009$), those who programmed the hand gesture interface ($p = 0.029$), and those who programmed the robot to move autonomously on the track ($p = 0.006$). Significant difference was also found in the coding practice $H(4) = 13.80,\ p = 0.008$. Pairwise comparisons with adjusted p-values showed that students who programmed the full-body interface devoted a significant amount of their time in coding compared to those who programmed the touchscreen interface ($p = 0.005$). Similarly, we employed the Mann–Whitney non-parametric test for comparisons between the touchscreen interface projects and the full-body ones. Concerning the abstracting and modularizing practice, the percentage time spent in the Touch projects ($Mdn = 25.50$) was significantly higher than the percentage time spent in the Body projects ($Mdn = 8.50$), $U = 2.00$, $z = -2.37$, $p = 0.017$, $r = -0.71$. Concerning the coding practice, the percentage time spent in the Touch projects ($Mdn = 0.80$) was significantly lower than the percentage time spent in the Body projects ($Mdn = 25.10$), $U = 20.00$, $z = 2.75$, $p = 0.004$, $r = 0.83$. In other words, higher embodiment levels led students to devote more time in coding and less time in abstracting and modularizing.

The second step in our analysis was to compare the computational practices of two groups of students. The first pair of students produced the most sophisticated project ($Constructs = 48,\ Patterns = 20$), while the second pair produced the least sophisticated ($Constructs = 32,\ Patterns = 8$). The first group selected a multimodal interaction style for navigating the robot.

Table 5. Students' Perspectives about Themselves before and after the Curriculum

Computational Perspectives	Pre-Test (N = 36)		Post-Test (N = 36)	
	Mean	**SD**	**Mean**	**SD**
How interested are you in computing education? ns	3.33	1.069	3.44	1.252
How difficult do you think computer programming is? ns	3.36	0.931	3.14	1.046
How many programming skills do you think you have?*	2.25	0.770	2.86	0.899
Would you like to learn programming in the future? ns	3.47	1.082	3.25	1.180
Would you like to create mobile applications in the future? ns	3.50	1.207	3.36	1.437
Would you like to build and program robots in the future? ns	3.22	1.333	3.19	1.191

ns = not significant ($p > 0.05$), $^*p < 0.05$.

For moving the robot forward and backward, they programmed an orientation-based interface, and for turning the robot right and left, they used a touchscreen interface. A power slider was used for changing the speed of the robot and a speech interface for triggering the movement of the robotic arm. The second group developed full-body interfaces for controlling the heading of the robot and its robotic arm, and did not program a speed control mechanism. The purpose of the comparison was to illustrate the clear differences in the two groups' strategies and the consequences of these strategies. Thus, this comparison may partly explain the observed differences in the time spent in each practice.

Specifically, both groups used extensively reusing and remixing for building their projects. However, we observed different strategies [44] between the two groups. The more competent students reused large parts of the code that was available from the previous sessions and afterward remixed them by removing the unnecessary parts. Thus, they followed an abstract way of developing their project. On the other hand, less-competent students followed an additive way as they developed their project step by step by reusing, remixing, and editing small parts of the code. They developed a little then they try it out and thus constructed their project in small steps through incremental and iterative cycles. This practice (being incremental-iterating) was not so dominant in the opposite group. A possible explanation for this phenomenon is that more competent students had a more transparent view from the start what elements needed for their projects, where they should go and what they should do. Additionally, we noticed that they organized their scripts in ways that made sense not only to them but also to others. In general, they spent more time in abstracting and modularizing strategies compared to less competent students. Finally, as noted in other studies [32] we also observed that both groups struggled with testing and debugging. However, and in this case, there were differences between the strategies [22] of the two groups. Correctly, the more competent students read their scripts thoroughly first to identify the cause of the problem and then made targeted modifications and tests to debug their projects. On the other hand, less competent students adopted less sophisticated strategies such as tinkering, making small changes in the scripts, and testing again and again until their project worked as expected.

4.3 Computational Perspectives

Finally, we measured students' perceptions of computing before and after the curriculum (Table 5). We first performed the Shapiro–Wilk test to assess normality and as the data were not normally distributed, we used the non-parametric Wilcoxon signed-ranked test to determine whether there was a significant change in students' perspectives. The results indicated that participants reported having significantly more programming skills after ($Mdn = 2.86$) the curriculum than before ($Mdn = 2.25$), $T = 238, p = 0.009, r = 0.31$. The differences in the other cases were not significant.

The qualitative data from the interviews provided additional insight into the findings that emerged from the questionnaires. As students were not self-selected to participate in the curriculum, we noticed two different approaches. On the one hand, students with limited, or no interest in computing education and programming before the curriculum reported that although after the treatment they had a more positive view, this experience was not strong enough to change their plans. For example, Athena said, "I was not interested in computing education at all before this experience, but now I like it a bit more. Nevertheless, I do not think that there is a shift in my future interests as I have already decided what I will do in the future." On the other hand, we had students with some interest in computing education who did not have the opportunity to participate in robotics activities in the past. For those participants, the curriculum had a strong effect on their motivational interest. For instance, Emanuel pointed out that "I had some thoughts to get involved with programming and this experience inspired me for my future endeavors."

Besides participants' views about themselves, we were also interested in the ideas that this embodied robotics experience led them to imagine. We asked them what kind of input devices they would like to use by creating the appropriate interfaces. Most students wanted to program a joystick-based interface, while others thought that it would be nice to use the GPS sensor or a camera embedded in the robot. "I wish I could control the robot with my mind, but I do not know if this is possible," Petros said [12]. Concerning the target platform, the majority reported that they would prefer a different robot and especially a humanoid robot that could mimic their body movements and could speak. Others noted that they would like to create mobile applications to control drones, cars or home appliances, such as TV sets. For example, Maria said, "I use my phone to stay connected with my friends. It never crossed my mind that I could use it in this way." Last, we asked them what kind of robotics challenges they would like to experience in the future. They reported that they would prefer more competitive robotics activities, such as racing with others on a multilane track, sumo-fighting [34], or football-playing [19].

5 DISCUSSION

This study sought to exploit the synergy between embodiment and educational robotics. Through a series of embodied robotics activities, we introduced computational concepts to children while at the same time examined their computational practices and perceptions of computing. We gave a problem-solving challenge to students with specific requirements: to navigate a robot on a fixed track and hit an object placed at a predefined spot with its robotic arm. Students adopted various interaction modalities, with a different level of embodiment, while building the interfaces for controlling the heading, the arm, and optionally the speed of a robot.

5.1 Embodiment and Development of Computational Thinking

Overall, although participants felt more confident about their programming skills after the robotics curriculum, the results indicate that our approach did not make a significant difference in their perceptions of computing (Table 5). This finding may be explained by the fact that participants were not self-selected in the course and the relatively short intervention. As Witherspoon et al. [62] argue it may require a considerable amount of time to cause a significant change in students' motivational interests. Nevertheless, it was encouraging to see how this experience led students to gain a new perspective about themselves and the technological world that surrounds them.

Perhaps the most significant finding that can be drawn from our study is the correlation between the level of embodiment and CT performance. Our analysis indicates that students who used combinations of low embodiment interfaces (touch and multimodal) or interfaces with no embodiment (autonomous) while developing the programming mechanisms, produced the most computationally sophisticated projects and adopted more sophisticated computational practices during the problem-solving challenge. On the other hand, students who used interfaces with a higher level of embodiment, such as full-body interfaces, not only produced the least sophisticated projects but also utilized less sophisticated computational strategies.

A possible explanation for the above finding comes from studies of embodied learning in the domain of mathematics [57]. Mainly, the use of fingers to count and calculate, defined as embodied numerosity [43], reflect an illustrative case of embodied cognition. For instance, it is quite common to observe novice learners, as they get involved in abstract numerical tasks, to utilize their fingers, thus grounding cognition in bodily experiences and reducing in this way the cognitive load imposed to them. As learner expertise increases, mental representations become more abstract and simplified [47, 52]. Thus, experts process information in a more intangible and disembodied manner.

Maybe we are witnessing a similar phenomenon in the computational thinking domain. Specifically, less competent students might have preferred a higher level of embodiment, while more competent students moved to more disembodied and abstract CT. Some researchers have suggested that higher embodiment levels are correlated with higher learning performance, while others argue that the opposite is true: Higher embodiment levels might lead to cognitive overload and thus to lower learning gains [54]. Perhaps higher levels of embodiment enabled less competent learners to offload cognition to the perceptual system by physically acting out the abstract computational concepts. Notably, as expertise increased the need to perceptually ground the computational concepts in high bodily activity diminished and CT become more intellectual and abstract. In other words, as mentality develops children move from the body dimension to the visual and finally to the symbolic one [33].

However, there is another possible explanation for the above finding. Although we have attempted to keep the balance between the conditions, each input has specific affordances, which may affect an individual's ability to perform optimally at some tasks [10, 64]. Specifically, the more embodied interface might have required additional motor-cognitive coordination and the involvement of bigger muscle groups [64] than fingers and hand gestures. This substantial effort for achieving adequate motor precision to control the robot might have added cognitive load to students [54] influencing their ability to write a more sophisticated program. In other words, interface usability might have had an impact on CT performance.

Given the equivocal nature of the finding, only further investigation will allow us to provide a conclusive explanation of why higher levels of embodiment led to lower computational sophistication.

5.2 Implications for Educational Robotics and Computational Thinking

The current study provides some significant implications for teaching abstract concepts and for applying the embodied approach to the field of educational robotics. For instance, teachers should design and incorporate embodied activities in their classes with varying levels of embodiment to provide alternative learning opportunities for all students, regardless of their mental status. Thereby, less competent students will be able to sense and feel with their bodies the concepts, while more competent students will be able to proceed to more intellectual learning stages. Additionally, the interplay between interface usability, motor effort, cognitive load, and learning performance should also be taken into consideration.

Within the domain of robotics, the current robotics curriculum also has some strong implications for how robotics can be taught in classrooms, workshops, and competitions. From the time that Papert [48] first introduced the Logo programming language and the turtle robot to the classroom, a considerable research effort has evolved in an attempt to invent new educational robotics technologies as well as to develop new innovative pedagogical methods. Nowadays, educational robots are not only meant to draw trigonometric shapes on the floor. Equipped with numerous sensors and actuators, that allow them to interact with the physical environment, they are capable to move autonomously and to perform various tasks as long as they are properly programmed. Previous well-established educational practices promote either the physical enactment of the problem-solving steps through role-playing activities (the learner enacts the robots' moves) or the manipulation of an external representative (the learner observes the teacher or another student acting out the robots' moves) before creating the program [21, 38, 56]. However, with our approach students are meant to interact with the robots, as their body movements transform directly, through programming, into commands. Thus, participants are able to enact the computational concepts as they carry out physical actions to control the robot synchronously.

We suggest that a robotics curriculum can be significantly improved regarding content, cognition, and motivational interest if the activities involve embodied experiences. In this way, educational robotics curriculums will become more approachable and meaningful to children [2].

5.3 Limitations

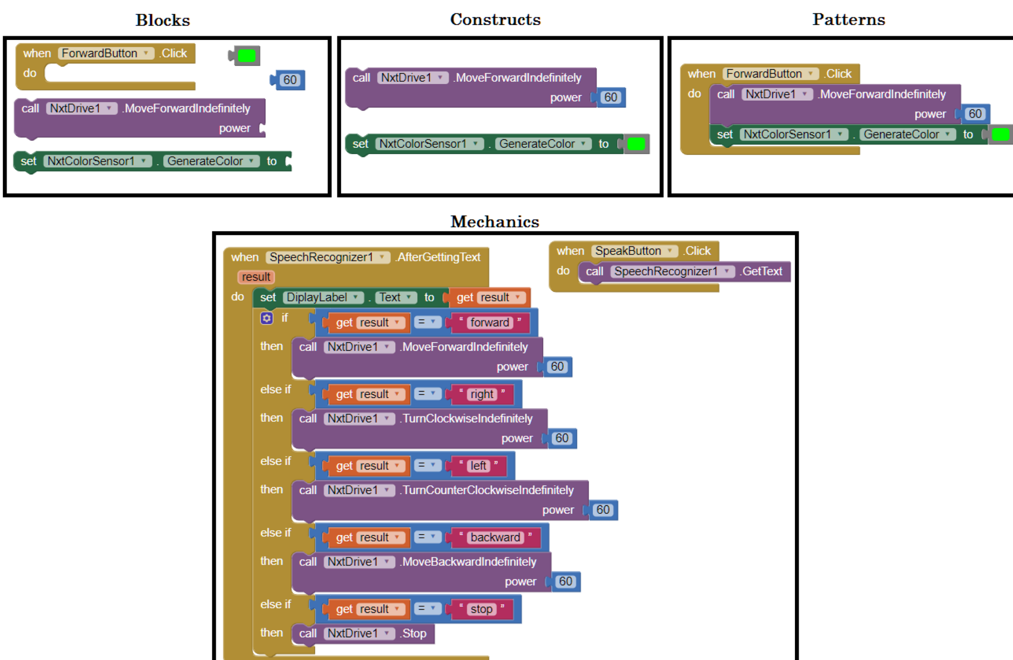
A limitation of the current study was the relatively small sample that prevented us from using parametric statistical testing; we analyzed only eighteen projects as the 36 students worked in pairs. Two reasons for the limited sample size were the duration of the robotics curriculum and the fact that the study was conducted in a formal classroom environment. Another limitation is that participants dealt with only three programming mechanics during the problem-solving task and that might have constrained their creativity. A third limitation is that we mainly employed project-based assessment for evaluating participants' fluency with particular computational concepts. Although analyzing young learners' projects is considered to be a meaningful form of performance assessment [25], this method of evaluation suffers from the criticism that students may reuse code without fully understanding its functional aspect [9, 16]. Werner et al. [58] defend this method of assessment against criticism by claiming that students' capability to remix the code they copied previously and make it function as desired in a new environment is a valid indication of their understandings. Nevertheless, as no single assessment is adequate, we also examined their computational practices to gain a more comprehensive view of their understandings. While all codes used for transcribing computational practices are clearly defined by Brennan and Resnick [9], the time-intensive nature of the coding process may be viewed as an additional limitation to apply this form of assessment in large scale.

6 CONCLUSION AND FURTHER RESEARCH

The contribution of this article is to provide additional insight into the learning impact of building human–robot interfaces with a different level of embodiment. Compared to previous studies, we exposed students to a wide range of interactive possibilities, and we examined the problem-solving strategies that arose. Our results suggest that embodiment within robotics can serve as an innovative approach to expand students' learning in CT. In this way, the established curriculum of programming an autonomous robot might be complemented with user interactions, as well as with hybrid modes that reflect the variety of human–robot interactions in research and practice [5]. We suggest that our findings might benefit teachers, assisting them in creating effective robotic interventions with an embodied learning perspective that blends the traditional autonomous robot movement with student enactment.

This form of physical enactment of computational concepts could be further applied in mixed and augmented reality programming activities where the user controls a virtual or physical agent through interactions with even higher levels of embodiment that involve locomotion such as walking in free space. A further study is needed with more significant numbers of participants and additional authentic problem-solving activities to confirm and generalize the findings of our research. Open-ended projects are necessary to promote students' creativity, giving them the opportunity to produce programs that are more complex. In this way, we will be able to identify and assess a larger number of programming constructs, patterns, and mechanics and gain a broader and more credible understanding of students' CT skills. Additionally, more embodied-oriented assessments methods should be used, measuring not only the short term but also the long-term learning gains [36] and the ability to transfer the acquired knowledge in related domains [27]. Besides CT performance, further research should concentrate on studying the effects of the embodiment on the comprehension of abstract STEM concepts, such as heading and speed. Future investigations might attempt to differentiate the input devices. For example, hand-held accelerometer-based devices (controllers such as the Nintendo Wii remote or Magic Wand), joysticks with haptic feedback, smartwatches, or wearable augmented reality devices (smart glasses such as Google Glass or Microsoft HoloLens) can be used for the interaction. Finally, future studies might also examine the use of different target platforms for the execution of code, such as humanoid robots for providing surrogate embodied experiences or drones as a means to introduce abstract concepts related to movement in three dimensions, such as orientation and gravity.

APPENDIX



Examples of block commands, constructs, patterns, and mechanics in App Inventor.

ACKNOWLEDGMENTS

The authors thank Professor Christopher Hundhausen and the three anonymous reviewers for their thoughtful comments and suggestions. We also convey special thanks to the students and teachers participating in this study.

REFERENCES

[1] D. Abrahamson. 2014. Building educational activities for understanding: An elaboration on the embodied-design framework and its epistemic grounds. *Int. J. Child-Comput. Interact.* 2, 1 (2014), 1–16.

[2] D. Alimisis. 2013. Educational robotics: Open questions and new challenges. *Themes Sci. Technol. Educ.* 6, 1 (2013), 63–71.

[3] B. J. Allen-Conn and K. Rose. 2003. *Powerful Ideas in the Classroom using Squeak to Enhance Math and Science Learning*. Viewpoints Research Institute, Inc.

[4] L. W. Barsalou. 2008. Grounded cognition. *Annu. Rev. Psychol.* 59 (Jan. 2008), 617–645.

[5] J. M. Beer, A. D. Fisk, and W. A. Rogers. 2014. Toward a framework for levels of robot autonomy in human-robot interaction. *J. Hum.-Robot Interact.* 3, 2 (2014), 74–99.

[6] F. Benitti and V. Barreto. 2012. Exploring the educational potential of robotics in schools: A systematic review. *Comput. Educ.* 58, 3 (2012), 978–988.

[7] M. U. Bers. 2010. The tangibleK robotics program: Applied computational thinking for young children. *Early Childhood Res. Pract.* 12, 2 (2010), n2.

[8] M. U. Bers, L. Flannery, E. R. Kazakoff, and A. Sullivan. 2014. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Comput. Educ.* 72 (Mar. 2014), 145–157.

[9] K. Brennan and M. Resnick. 2012. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association.* 1–25.

[10] S. K. Card, W. K. English, and B. J. Burr. 1978. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics* 21, 8 (1978), 601–613.

[11] M. S. Chan and J. B. Black. 2006. Direct-manipulation animation: Incorporating the haptic channel in the learning process to support middle school students in science learning and mental model acquisition. In *Proceedings of the 7th international Conference on Learning Sciences.* International Society of the Learning Sciences, 64–70.

[12] C. S. Crawford, C. Gardner-McCune, and J. E. Gilbert. 2018. Brain-computer interface for novice programmers. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education.* ACM, 32–37.

[13] S. B. Daily, A. E. Leonard, S. Jörg, S. Babu, and K. Gundersen. 2014. Dancing alice: Exploring embodied pedagogical strategies for learning computational thinking. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education.* ACM, 91–96.

[14] A. Damasio. 1994. Descartes' error: Emotion, reason, and the human brain GP Putnam's Sons. *The Somatic-marker Hypothesis* (1994), 165–201.

[15] B. B. de Koning and H. K. Tabbers. 2011. Facilitating understanding of movements in dynamic visualizations: An embodied perspective. *Educ. Psychol. Rev.* 23, 4 (2011), 501–521.

[16] B. Dorn, A. E. Tew, and M. Guzdial. 2007. Introductory computing construct use in an end-user programming community. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'07).* IEEE, 27–32.

[17] P. Dourish. 2004. *Where the Action Is: The Foundations of Embodied Interaction.* MIT Press.

[18] A. Eguchi. 2013. Educational robotics for promoting 21st century skills. *J. Autom. Mobile Robot. Intell. Syst.* 8, 1 (2013), 5–11.

[19] A. Eguchi. 2016. RoboCupJunior for promoting STEM education, 21st century skills, and technological advancement through robotics competition. *Robot. Auton. Syst.* 75, Part B (Jan. 2016), 692–699.

[20] C. L. Fadjo, G. Hallman Jr., R. Harris, and J. B. Black. 2009. Surrogate embodiment, mathematics instruction and video game programming. In *EdMedia: World Conference on Educational Media and Technology*. Association for the Advancement of Computing in Education (AACE), 2787–2792.

[21] C. L. Fadjo. 2012. *Developing Computational Thinking through Grounded Embodied Cognition.* Columbia University.

[22] S. Fitzgerald, G. Lewandowski, R. McCauley, L. Murphy, B. Simon, L. Thomas, and C. Zander. 2008. Debugging: Finding, fixing and flailing, a multi-institutional study of novice debuggers. *Comput. Sci. Educ.* 18, 2 (2008), 93–116.

[23] V. Gallese and G. Lakoff. 2005. The brain's concepts: The role of the sensory-motor system in conceptual knowledge. *Cogn. Neuropsychol.* 22, 3–4 (2005), 455–479.

[24] S. Grover and R. Pea. 2013. Using a discourse-intensive pedagogy and android's app inventor for introducing computational concepts to middle school students. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education.* ACM, 723–728.

[25] S. Grover. 2017. Assessing algorithmic and computational thinking in K-12: Lessons from a middle school classroom. In *Emerging Research, Practice, and Policy on Computational Thinking*. Springer, Cham, 269–288.

[26] I. Han and J. B. Black. 2011. Incorporating haptic feedback in simulation for learning physics. *Comput. Educ.* 57, 4 (2011), 2281–2290.

[27] I. Han. 2013. Embodiment: A new perspective for evaluating physicality in learning. *J. Educ. Comput. Res.* 49, 1 (2013), 41–59.

[28] M. Heidegger. 1927. *Being and Time*. Harper & Row, New York.

[29] S. Howell. 2012. Kinect2Scratch (Version 2.5).

[30] R. J. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey, and J. Zigelbaum. 2008. Reality-based interaction: A framework for post-WIMP interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 201–210.

[31] M. C. Johnson-Glenberg, C. Megowan-Romanowicz, D. A. Birchfield, and C. Savio-Ramos. 2016. Effects of embodied learning and digital platform on the retention of physics content: Centripetal force. *Front. Psychology* 7 (Nov. 2016), 1819.

[32] Y. B. Kafai, E. Lee, K. Searle, D. Fields, E. Kaplan, and D. Lui. 2014. A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)* 14, 1 (2014), 1.

[33] A. Kay. 2010. Alan Kay on Learning and Computer Science. YouTube. Retrieved April 5, 2018 from https://www.youtube.com/watch?v=Ud8WRAdihPg.

[34] S. H. Kim and J. W. Jeon. 2009. Introduction for freshmen to embedded systems using LEGO Mindstorms. *IEEE Trans. Educ.* 52, 1 (2009), 99–108.

[35] C. Kontra, D. J. Lyons, S. M. Fischer, and S. L. Beilock. 2015. Physical experience enhances science learning. *Psychol. Sci.* 26, 6 (2015), 737–749.

[36] R. Lindgren and M. Johnson-Glenberg. 2013. Emboldened by embodiment: Six precepts for research on embodied learning and mixed reality. *Educ. Res.* 42, 8 (2013), 445–452.

[37] R. Lindgren, M. Tscholl, S. Wang, and E. Johnson. 2016. Enhancing learning and engagement through embodied interaction within a mixed reality simulation. *Comput. Educ.* 95 (Apr. 2016), 174–187.

[38] C. M. Lu, S. Kang, S. C. Huang, and J. B. Black. 2011. Building student understanding and interest in science through embodied experiences with LEGO Robotics. In *Proceedings of the World Conference on Educational Media and Technology (EdMedia'11)*. Association for the Advancement of Computing in Education (AACE), 2225–2232.

[39] E. F. Melcer and K. Isbister. 2016. Bridging the physical divide: A design framework for embodied learning games and simulations. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2225–2233.

[40] A. Merkouris, K. Chorianopoulos, and A. Kameas. 2017. Teaching programming in secondary education through embodied computing platforms: Robotics and wearables. *ACM Trans. Comput. Educ.* 17, 2 (2017), 9.

[41] M. Merleau-Ponty. 1945. *The Phenomenology of Perception*. Routledge, London.

[42] S. Millar. 1999. Memory in touch. *Psicothema* 11, 4 (1999), 747–767.

[43] K. Moeller, U. Fischer, T. Link, M. Wasner, S. Huber, U. Cress, and H. C. Nuerk. 2012. Learning and development of embodied numerosity. *Cogn. Process.* 13, 1 (2012), 271–274.

[44] A. Monroy-Hernandez. 2012. *Designing for Remixing: Supporting an Online Community of Amateur Creators*. Ph.D. dissertation, Massachusetts Institute of Technology.

[45] M. Montessori. 1966. *The Secret of Childhood*. Fides, Notre Dame, IN, 20.

[46] S. Oviatt, A. Cohen, A. Miller, K. Hodge, and A. Mann. 2012. The impact of interface affordances on human ideation, problem solving, and inferential reasoning. *ACM Trans. Comput.-Hum. Interact.* 19, 3 (2012), 22.

[47] F. Paas, A. Renkl, and J. Sweller. 2004. Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture. *Instruct. Sci.* 32, 1–2 (2004), 1–8.

[48] S. Papert. 1980. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York, NY.

[49] S. Papert. 1987. March. Tomorrow's classrooms. In *New Horizons in Educational Computing*. Wiley-Interscience, 17–20.

[50] D. Parmar, J. Isaac, S. V. Babu, N. D'Souza, A. E. Leonard, S. Jörg, K. Gundersen, and S. B. Daily. 2016. Programming moves: Design and evaluation of applying embodied interaction in virtual environments to enhance computational thinking in middle school students. In *Proceedings of the 2016 IEEE Virtual Reality Conference (VR'16)*. IEEE, 131–140.

[51] J. Piaget. 2013. *The Construction of Reality in the Child*, Vol. 82. Routledge.

[52] W. T. Pouw, T. Van Gog, and F. Paas. 2014. An embedded and embodied cognition review of instructional manipulatives. *Educ. Psychol. Rev.* 26, 1 (2014), 51–72.

[53] A. Segal. 2011. *Do Gestural Interfaces Promote Thinking? Embodied Interaction: Congruent Gestures and Direct Touch Promote Performance in Math*. Ph. D. Dissertation. Teachers College.

[54] A. Skulmowski and G. D. Rey. 2018. Embodied learning: Introducing a taxonomy based on bodily engagement and task integration. *Cogn. Res.: Princip. Implic.* 3, 1 (2018), 6.

[55] W. Sung, J. Ahn, and J. B. Black. 2017. Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technol. Knowl. Learn.* 22, 3 (2017), 443–463.

[56] W. Sung, J. H. Ahn, S. M. Kai, and J. Black. 2017. Effective planning strategy in robotics education: An embodied approach. In *Society for Information Technology & Teacher Education International Conference*. Association for the Advancement of Computing in Education (AACE), 1065–1071.

[57] C. Tran, B. Smith, and M. Buschkuehl. 2017. Support of mathematical thinking through embodied cognition: Nondigital and digital approaches. *Cogn. Res.: Princ. Implic.* 2, 1 (2017), 16.
[58] L. Werner, J. Denner, and S. Campe. 2015. Children programming games: A strategy for measuring computational learning. *ACM Trans. Comput. Educ.* 14, 4 (2015), 24.
[59] M. Wilson. 2002. Six views of embodied cognition. *Psychon. Bull. Rev.* 9, 4 (2002), 625–636.
[60] J. M. Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.
[61] J. M. Wing. 2008. Computational thinking and thinking about computing. *Philos. Trans. Roy. Soc. Lond. A* 366, 1881 (2008), 3717–3725.
[62] E. B. Witherspoon, C. D. Schunn, R. M. Higashi, and R. Shoop. 2018. Attending to structural programming features predicts differences in learning and motivation. *J. Comput. Assist. Learn* 34, 2 (Apr. 2018), 115–128.
[63] Z. C. Zacharia, E. Loizou, and M. Papaevripidou. 2012. Is physicality an important aspect of learning through science experimentation among kindergarten students? *Early Childhood Res. Quart.* 27, 3 (2012), 447–457.
[64] S. Zhai, P. Milgram, and W. Buxton. 1996. The influence of muscle groups on performance of multiple degree-of-freedom input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 308–315.
[65] O. Zuckerman and A. Gal-Oz. 2013. To TUI or not to TUI: Evaluating performance and preference in tangible vs. graphical user interfaces. *Int. J. Hum.-Comput. Stud.* 71, 7–8 (2013), 803–820.

Received July 2018; revised January 2019; accepted April 2019