Programming Touch and Full-body Interaction with a Remotely Controlled Robot in a Secondary Education STEM Course

Alexandros Merkouris
Department of Informatics
Ionian University
Corfu, Greece
c14merk@ionio.gr

Konstantinos Chorianopoulos
Department of Informatics
Ionian University
Corfu, Greece
choko@ionio.gr

ABSTRACT

Contemporary research has introduced educational robotics in the classroom, but there are few studies about the effects of alternative embodied interaction modalities on computational thinking and science education. Twenty-six middle school students were asked to program interfaces for controlling the heading and speed of a robot using two types of embodied interaction modalities. We compared touch and full-body gestures to autonomous control, which does not require any embodied interaction. We assessed the development of their computational thinking skills by analyzing the projects they created during a problem-solving task and examined their understandings of science concepts related to kinematics. We found that novice students preferred full-body interfaces, while advanced students moved to more disembodied and abstract computational thinking. These findings might be applied to focus computing and science education activities to the right age and abilities groups of students.

CCS CONCEPTS

• Social and professional topics → Computer science education; K-12 education;

KEYWORDS

Embodied learning, Educational robotics, Children, Human-robot interaction, Computational thinking, Assessment

ACM Reference format:

Alexandros Merkouris and Konstantinos Chorianopoulos. 2018. Programming Touch and Full-body Interaction with a Remotely Controlled Robot in a Secondary Education STEM Course. In *Proceedings of 22nd Pan-Hellenic Conference on Informatics (PCI'18). ACM, New York, NY, USA, 5 pages.* https://doi.org/10.1145/3291533.3291537

PCI '18, November 29-December 1, 2018, Athens, Greece

ACM ISBN 978-1-4503-6610-6/18/11$15.00
https://doi.org/10.1145/3291533.3291537

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 [5] as well as to foster computational thinking (CT) [6]. In a typical educational robotics activity, children are asked to enliven the robots by creating the appropriate programs. The programmer has to think mainly about the goal of the robot and how the robot will interact with the environment. However, another important aspect is whether and how the user will physically interact with the robot.

With the rapid development of digital technologies, such as mobile devices, touch screens, and computer vision, a wide spectrum of interfaces is provided to users. Children can interact with digital information more naturally and physically [11], using touch and full-body gestures as input for the interaction besides the conventional keyboard and mouse interfaces. There has been a strong push to exploit these interfaces in science and computing education triggered by the views of embodied cognition researchers [4, 23] that physical interactions with learning objects through sensorimotor modalities (touch, movement, speech, smell and vision) are essential factors in the construction of knowledge.

The embodied approach has been used to cover the learning of abstract materials in a wide range of topics that extend from science [10, 12, 14], technology, engineering and mathematics [1] to CT [8, 19]. Recently, a growing number of educators and researchers have considered educational robotics as a promising field for applying the embodied cognition view. In particular, Alimisis [2] points out that embodiment is an innovative approach for making robotic activities more approachable and meaningful to children. Direct embodiment, where students enact with their bodies the robots' moves before creating the program, and surrogate embodiment, where learners manipulate and observe an external representative, seem a useful approach for learning abstract learning abstract STEM and computational concepts [15, 20].

This small sample of embodied studies highlights the need to explore further the positive learning effects of embodiment within robotics. In response to this necessity, we implemented educational robotics activities for studying the development of

CT, but we adopted a different embodied learning approach. Instead of asking students to enact the robots' moves, we asked them to program human-robot interfaces with a different type of embodiment. In this way, we could find a connection between interactivity and the development of CT skills. The ideas of Papert [18] 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. Our research questions centered on these major topics:

- *Interface affordances for scientific exploration*: What kind of interaction modalities did students select for controlling the heading and speed of a robot and what were the criteria for their selections?
- *Comprehension*: Were there any differences in the development of students' CT skills that could be attributed to the different types of embodiment?

2 METHODOLOGY

2.1 Subjects

We recruited twenty-six middle school students (13 girls, 13 boys), aged between fourteen and fifteen years, with little to no prior programming experience to participate in a four-session robotic curriculum. We randomly seleceted the participants from the third-level class of a middle school. The decision for selecting this specific age group was guided by the fact that none of the students had previously received teaching in computer programming as part of previous formal education. Students worked in pairs in each of the activities. Thus, ten same-gender and three mixed-gender pairs were created.

2.2 Robotic Curriculum

The curriculum was divided into four individual sessions. In Table 1, we present what kind of applications students were asked to create, and which CT and scientific concepts they explored during the sessions. A guided approach was adopted for the first three sessions, while followed a similar basic format: 1) Building the User Interface, 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 were able to apply the previously acquired programming knowledge to a problem-solving task. They were asked 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 (Fig. 1). No instructions were given to students on the final session, and they were prompted to program any interface (touch, full-body, artificial intelligence) they preferred. Moreover, they were allowed to "remix and reuse" [7] code from the previous sessions. The final session followed a constructionist approach to learning and served as the condition for assessing learning outcomes of the intervention.

Activity Title	Students should create an application...	Computational Thinking Concepts	Scientific Concepts
Touch Control	to control the robot with their fingers by touching their mobile phone screens	Events, Sequences, Data	Kinematics (heading and speed)
Body Control	to control the robot with full-body gestures, using computer vision technology	Events, Sequences, Parallelism, Loops, Data, Conditionals, Operators	Kinematics (heading and speed)
Line Follow	to integrate Artificial Intelligence to the robot so that it could move autonomously on the track	Events, Sequences, Data, Conditionals, Operators	Kinematics (heading and speed)
Project	to navigate a robot on a fixed track and hit an object	Sequences, Loops, Events, Parallelism, Conditionals, Operators, Data	Kinematics (heading and speed)

Table 1: Overview of the activities, the CT and scientific concepts introduced in each session.

The duration of each of the first three sessions was about 45 minutes while the final project activity lasted between 45 to 90 minutes. Thus, an adequate amount of time was given students to develop their programs.

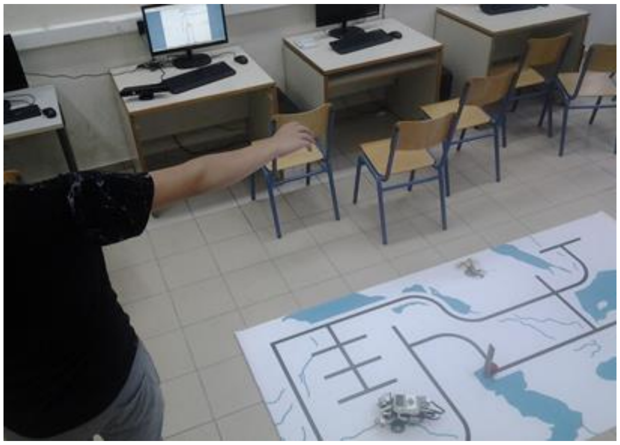

Figure 1: Controlling a robot in the final session with full-body gestures.

2.3 Powerful Ideas Explored

In line with the Papert's and Kay's views [3, 18], our robotic curriculum intended to provide context for exploring powerful ideas through programming. Although we introduced a wide range of abstract STEM concepts in each activity, in this work, we focused on the kinematics domain and specifically we were mainly interested in the scientific concepts of heading and speed (Table 1). The concepts of heading and speed were investigated in multiple activities as participants created applications, where users are carrying out actions with a different type of embodiment, and then they investigate and understand the effect of the actions in the kinematics of a robot. They investigate "by doing and by discovery" and "understand just enough to get going" as Papert [18] suggests.

2.4 Materials

We employed App Inventor[1] [9] as the development platform for the sessions that involved mobile technology and students used their own mobile phone devices. For the session that involved full-body interaction, ScratchX [2] was used as the development platform and was supported by the Kinect sensor for tracking the body. The interaction modalities (touch and full-body gestures) varied in the amount of kinesthetic and gestural congruency [12]. The robots chosen for supporting the curriculum were Lego Mindstorms[3]. Both App Inventor and ScratchX 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 programming environments, the coding area was very similar and based on the idea of snapping blocks together.

2.5 Measuring Instruments and Data Analysis

For the study, both qualitative and quantitative data were collected and analyzed. Concerning the quantitative data, the students filled out a brief "Again Again table" [20] questionnaire for evaluating each of the activities. Regarding the qualitative data, students' projects in the final session were manually analyzed for assessing the development of CT. The projects were graded based on a rubric used for grading student-made computer game projects [22]. The rubric was appropriately adjusted to fit the current intervention characteristics. According to Werner [22] programs are composed of programming constructs, pattern, and mechanics. 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 speed control mechanisms. Additionally, students' on-screen activity was recorded by Camtasia[4] capture to gain an overview of their practices during the final session.

3 RESULTS

3.1 Activities Evaluation

First, we evaluated the four individual activities using the "Again Again table" [20] questionnaire (Fig. 2). Because the assumption of normality has been violated, we applied the non-parametric Friedman's ANOVA test to verify whether there were significant statistical differences among the activities. The analysis indicated that there was indeed a significant difference, $x^2(3) = 15.09$, $p = .002$. Wilcoxon tests were used to follow up this finding. It appears that the Project activity was more engaging than the Line Follow activity, $T = 100.5, r = .43$. Similarly, the Body Control activity was more favored to be repeated than the Line Follow activity, $T = 4.5, r = -.39$.

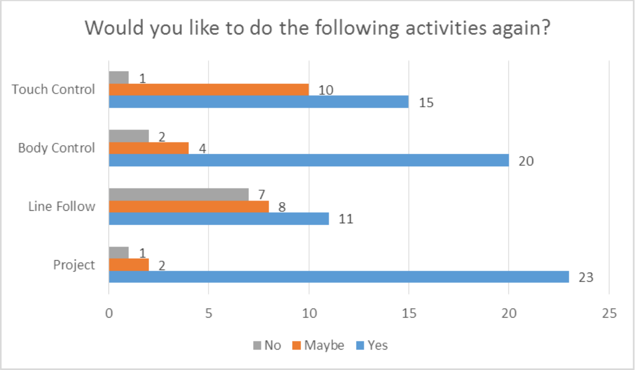

Figure 2: "Again Again table" for evaluating the activities.

3.2 Building Interfaces for Heading and Speed

To complete the problem-solving task given to them in the final session students programmed interfaces where the users produced actions with a different type of embodiment to handle the heading and speed of the robot (Fig. 3).

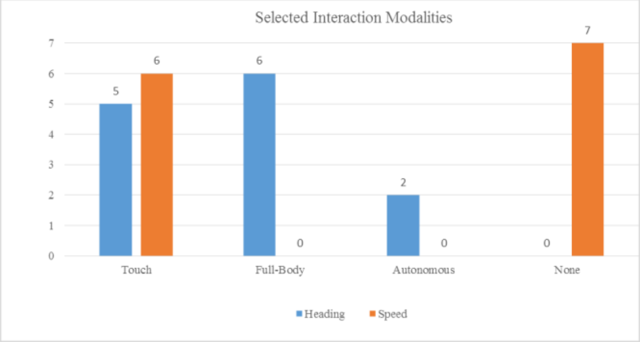

Figure 3: Selected interaction modalities for heading and speed.

Concerning the concept of heading, in most cases, full-body gestures and touch sensorimotor were selected as input for the interaction. Notably, for the full-body interface the absolute or relative position between two different body joints in the Cartesian coordinate system was used to control the heading of the robot. For example, when the user lifted his right hand above his right shoulder, the robot would move forward. While in the case of the touchscreen interface, students explored the concept of heading by manipulating a traditional touchscreen button interface. For navigating the robot with accuracy on the track, the

---

[1] App Inventor: http://appinventor.mit.edu

[2] ScratchX: http://scratchx.org/

[3] Lego Mindstorms: https://www.lego.com/en-us/mindstorms

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

program must respond immediately to the users' actions. Finally, two groups of students implemented a feedback loop so that the robot could move autonomously. The robot continuously tracked with its color sensor the black line and adjusted its heading automatically.

For the speed control mechanism, six groups of students preferred to program an interface that allowed them to adjust speed with touch sensorimotor, by manipulating a power slider. Thus, the position of the power slider was mapped to the speed, and any change in its position changed the robot's speed. Surprisingly, none of the students implemented a full-body interface even though they were previously given instructions in the Body Control session how they could vary the speed according to the distance between two body joints (right and left knee) in the Cartesian coordinate system. Finally, seven group of students did not implement a mechanism for adjusting the speed of the robot.

3.3 Assessing Computational Thinking

We applied Brennan's and Resnick's framework [7] for assessing the development of CT. Within their framework, three dimensions are defined: computational concepts; computational practices; and computational perspectives. Nevertheless, we concentrated our analysis on the first two computational dimensions.

3.3.1 Computational Concepts. An important aspect of the proposed framework is computational concepts, such as sequences, loops, parallelism, events, conditionals, operators, and data. 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 2.5. We calculated the mean averages of the used programming constructs and patterns (Fig. 4).

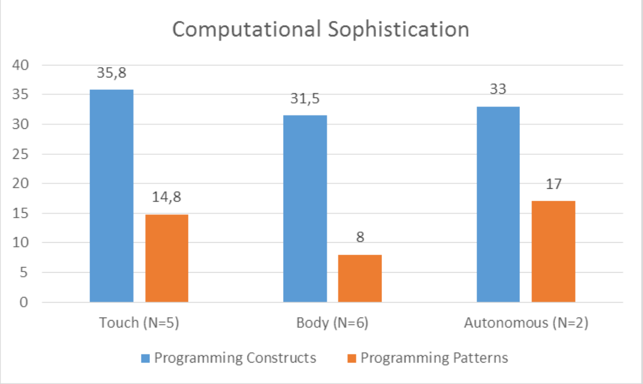

Figure 4: Mean averages of the used programming constructs and pattern according to the selected interaction modalities.

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 test, there was statistically significant difference in patterns $H(4) = 13.15, p = .011$. Pairwise comparisons with adjust p-values showed that the difference was significant between the projects that students used the full-body interface and those that the robot was programmed to move autonomously on the track ($p = .033$). 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 with the full-body ones, as the sample groups were similar ($N = 5$ and $N = 6$ respectively). Constructs in the Touch projects ($Mdn = 36.00$) did not differ significantly from constructs in the Body projects ($Mdn = 32.00$), $U = 4.00, z = -2.04, p = .052, r = -.61$. However, Patterns in the Touch projects ($Mdn = 16.00$) were significant higher than patterns in the Body projects ($Mdn = 8.00$), $U = 0.00, z = -3.03, p = .004, r = -.91$. In other words, full-body interfaces led to projects with overall lower computational sophistication.

3.3.2 Computational Practices. Computational practices are an additional key dimension of CT. Brennan and Resnick [7] defined four practices: being incremental and iterative, testing and debugging, reusing and remixing, and abstracting and modularizing.

Here, we have attempted to analyze computational practices by observing the on-screen problem-solving activity during the final session. Mainly, we examined the computational practices of two groups of students. The first pair of students produced the most sophisticated project ($Constructs = 36, Patterns = 17$), while the second pair produced the least sophisticated ($Constructs = 32, Patterns = 8$). The first group (*advanced*) controlled the heading of the robot and the movement of the robotic arm autonomously. A power slider was used for changing the speed of the robot. The second group (*novice*) used full-body interfaces for heading the robot and controlling the robotic arm, and did not program a speed control mechanism.

We noticed differences in the strategies that the two groups followed in each computational practice. Specifically, both groups used extensively reusing and remixing for building their projects. However, we observed that more advanced students reused large parts of the code that was available from the previous session and afterward remixed them by removing the unnecessary parts. On the other hand, students in the novice group developed step by step their project by reusing, remixing and editing small parts of the code. They developed a little then they try it out, thus constructed their project in small steps through incremental and iterative cycles. This practice (being incremental-iterating) was not noticed in the advanced group. A possible explanation for this phenomenon is that students in the advances group had a more transparent view from the start what elements needed for their projects, where they should go and what they should do. In general, they spent more time in abstracting and modularizing strategies compared to students in the novice group. Finally, as noted in other studies [13] we also observed that both groups struggled with testing and debugging. Correctly, the more advanced students first read their scripts thoroughly to identify the cause of the problem and then made targeted modifications and tests to debug their projects. On the contrary, novices 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 Discussion and Further Research

This study sought to exploit the synergy between embodied interaction and educational robotics. Through a series of robotics activities, we introduced abstract STEM and computational concepts to children while at the same time examined the development of their CT skills. Students adopted various interaction modalities while building the interfaces for controlling the heading, arm, and speed of the robot. Six groups created full-body interfaces for controlling the heading of the robot and its robotic arm. Surprisingly, none of the groups that used the body interfaces implemented the speed control mechanism as students struggled to program a concurrent body gesture. For the robot navigation, touch sensorimotor was also extensively used, as it allowed users to guide the robot more accurately. It seems that the participants not only chose interfaces that were attractive to them but also interfaces that their affordances matched to the specific programming tasks [17].

Perhaps the most significant finding is the correlation between the types of embodiment and CT performance. Our analyses indicate that students who used touchscreen interfaces or interfaces with no embodiment (autonomous) produced the most computational sophisticated projects. On the other hand, students who used full-body interfaces produced the least sophisticated projects. A possible explanation is that embodiment enabled novice learners to offload cognition to the perceptual system by physically acting out abstract computational concepts. Notably, as expertise increased the need to perceptually ground computational concepts in high bodily activity diminished and CT become more intellectual and disembodied.

Our results suggest that embodiment within robotics can serve as an innovative approach to expand students' learning in CT and STEM. We believe that the findings of our study might benefit teachers, assisting them in creating effective robotic interventions with an embodied learning perspective.

Besides CT performance, further research should concentrate on studying the effects of the embodiment on the comprehension of abstract STEM concepts, such as heading, speed. Finally, future studies might also examine the use of different target platforms [16] for the execution of code, such as wearables, 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.

REFERENCES

[1] Abrahamson, D., 2014. Building educational activities for understanding: an elaboration on the embodied-design framework and its epistemic grounds. *International Journal of Child-Computer Interaction*, *2*(1), pp.1-16.

[2] Alimisis, D., 2013. Educational robotics: Open questions and new challenges. *Themes in Science and Technology Education*, *6*(1), pp.63-71.

[3] Allen-Conn, B.J. and Rose, K., 2003. *Powerful ideas in the classroom using squeak to enhance math and science learning*. Viewpoints Research Institute, Inc.

[4] Barsalou, L.W., 2008. Grounded cognition. *Annu. Rev. Psychol.*, *59*, pp.617-645.

[5] Benitti, F. and Barreto, V., 2012. Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, *58*(3), pp.978-988.

[6] Bers, M.U., Flannery, L., Kazakoff, E.R. and Sullivan, A., 2014. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, pp.145-157.

[7] Brennan, K. and Resnick, M., 2012, April. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada* (pp. 1-25).

[8] Fadjo, C.L., 2012. *Developing computational thinking through grounded embodied cognition.* Columbia University.

[9] Grover, S. and Pea, R., 2013, March. Using a discourse-intensive pedagogy and android's app inventor for introducing computational concepts to middle school students. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 723-728). ACM.

[10] Han, I. and Black, J.B., 2011. Incorporating haptic feedback in simulation for learning physics. *Computers & Education*, *57*(4), pp.2281-2290.

[11] Jacob, R.J., Girouard, A., Hirshfield, L.M., Horn, M.S., Shaer, O., Solovey, E.T. and Zigelbaum, J., 2008, April. Reality-based interaction: a framework for post-WIMP interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 201-210). ACM.

[12] Johnson-Glenberg, M.C., Megowan-Romanowicz, C., Birchfield, D.A. and Savio-Ramos, C., 2016. Effects of embodied learning and digital platform on the retention of physics content: Centripetal force. *Frontiers in psychology*, *7*, p.1819.

[13] Kafai, Y.B., Lee, E., Searle, K., Fields, D., Kaplan, E. and Lui, D., 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), p.1.

[14] Lindgren, R., Tscholl, M., Wang, S. and Johnson, E., 2016. Enhancing learning and engagement through embodied interaction within a mixed reality simulation. *Computers & Education*, *95*, pp.174-187.

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

[16] Merkouris, A., Chorianopoulos, K. and Kameas, A., 2017. Teaching programming in secondary education through embodied computing platforms: Robotics and wearables. *ACM Transactions on Computing Education (TOCE)*, *17*(2), p.9.

[17] Oviatt, S., Cohen, A., Miller, A., Hodge, K. and Mann, A., 2012. The impact of interface affordances on human ideation, problem solving, and inferential reasoning. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *19*(3), p.22.

[18] Papert, S., 1987, March. Tomorrow's classrooms. In *New horizons in educational computing* (pp. 17-20). Wiley-Interscience.

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

[20] Read, J.C., 2008. Validating the Fun Toolkit: an instrument for measuring children's opinions of technology. *Cognition, Technology & Work*, *10*(2), pp.119-128.

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

[22] Werner, L., Denner, J. and Campe, S., 2015. Children programming games: a strategy for measuring computational learning. *ACM Transactions on Computing Education (TOCE)*, *14*(4), p.24.

[23] Wilson, M., 2002. Six views of embodied cognition. *Psychonomic bulletin & review*, *9*(4), pp.625-636.