Teaching Programming in Secondary Education Through Embodied Computing Platforms: Robotics and Wearables

ALEXANDROS MERKOURIS, Ionian University
KONSTANTINOS CHORIANOPOULOS, Hellenic Open University
ACHILLES KAMEAS, Hellenic Open University

Pedagogy has emphasized that physical representations and tangible interactive objects benefit learning especially for young students. There are many tangible hardware platforms for introducing computer programming to children, but there is limited comparative evaluation of them in the context of a formal classroom. In this work, we explore the benefits of learning to code for tangible computers, such as robots and wearable computers, in comparison to programming for the desktop computer. For this purpose, 36 students participated in a within-groups study that involved three types of target computer platform tangibility: (1) desktop, (2) wearable, and (3) robotic. We employed similar blocks-based visual programming environments, and we measured emotional engagement, attitudes, and computer programming performance. We found that students were more engaged by and had a higher intention of learning programming with the robotic rather than the desktop computer. Furthermore, tangible computing platforms, either robot or wearable, did not affect the students' performance in learning basic computational concepts (e.g., sequence, repeat, and decision). Our findings suggest that computer programming should be introduced through multiple target platforms (e.g., robots, smartphones, wearables) to engage children.

CCS Concepts: • **Social and professional topics** → **Computer science education**; **K-12 education**; *Student assessment*; • **Computer systems organization** → *Robotics*

Additional Key Words and Phrases: Ubiquitous computing, embodiment, robot, wearable, learning, experiment, children

ACM Reference Format:
Alexandros Merkouris, Konstantinos Chorianopoulos, and Achilles Kameas. 2017. Teaching programming in secondary education through embodied computing platforms: Robotics and wearables. ACM Trans. Comput. Educ. 17, 2, Article 9 (May 2017), 22 pages.
DOI: http://dx.doi.org/10.1145/3025013

9

1. INTRODUCTION

The importance of computer programming education has been growing along with the diffusion of computing applications in many aspects of work and everyday life. At the same time, students have become familiar with the use of information technology (e.g., desktop computer, smartphone and tablet, video-game console, etc.), but do not have skills in computer programming [Resnick et al. 2009]. In particular, computing education research has highlighted the need to motivate children to learn the basics of computer programming [Kelleher and Pausch 2005]. The benefits of learning to program the computer extend beyond the deeper understanding of science and math

Authors' addresses: A. Merkouris, Department of Informatics, Ionian University, Tsirigoti Square 7, GR-49100, Corfu, Greece; email: c14merk@ionio.gr; K. Chorianopoulos and A. Kameas, School of Science and Technology, Hellenic Open University, Parodos Aristotelous 18, GR-26335, Patras, Greece; emails: choko@acm.org, kameas@eap.gr.

DOI: http://dx.doi.org/10.1145/3025013

into creativity and the use of language [Haugland 1992]. For this reason, students should also be provided with the appropriate stimuli that motivate them to learn computer programming.

Previous works in computer programming education for children have taken into consideration many parameters, such as visual programming environments (e.g., Scratch [Resnick et al. 2009]), gender issues (e.g., Alice [Kelleher et al. 2007]), and pedagogy [Cooper et al. 2010]. Nevertheless, there is limited consideration of the embodied dimension of learning, because most of the approaches are focused only on the cognitive aspect. Indeed, the desktop computer has been employed in most cases of computer education both as a programming tool and as the target for computer program execution. Although computer programming is a highly abstract cognitive activity, the learning of computer programming might be enhanced if it is channeled through embodied mediums, such as robotics and wearables.

Research in educational robotics is based to a great extent on Papert's [1980] hands-on, experiential theory (constructionism). Robotic computing platforms have been proposed as a means to engage students with a particular focus on the Science, Technology, Engineering, and Mathematics (STEM) curriculum [Barreto and Benitti 2012; Eguchi 2013; Mubin et al. 2013; Nugent et al. 2009]. Besides the use of educational robotics, another application of constructionism in education targeting mostly girls is that of computational textiles. E-textiles have been effectively used to introduce STEM sciences to students in a more appealing way [Buechley et al. 2007; Buechley et al. 2008; Lau et al. 2009; Kafai et al. 2014; Katterfeldt et al. 2009; Qiu et al. 2013]. However, limited research has been conducted on the effectiveness of target computing platforms. Although there is previous research [Kafai et al. 2014; Katterfeldt et al. 2009; Qiu et al. 2013] on writing computer code through tangibles and on the effectiveness of using these platforms in order to acquire new skills [Marshall 2007], there is limited evidence on students' attitudes and their level of engagement in the process of writing code for tangible computing platforms.

In this work, we explore the benefits of teaching computer programming through embodied platforms such as robotics and wearable computers. Our research questions consider the following issues:

1. Which target platform can develop students' programming skills most effectively?
2. Are there differences between boys and girls with regard to the preference of a tangible platform?
3. Is tangible computing more engaging than desktop computing in learning computer programming?

Therefore, we focus on the experimental evaluation of robotic and wearable computing platforms as motivators for learning to program in a real-world classroom.

2. RELATED WORK

According to the constructionist learning theory, children are better learners when they construct knowledge voluntarily, for a personally significant purpose, engaged in designing and creating visible objects such as computer programs, animations, robots, and e-textiles [Papert 1980; Resnick et al. 1996]. For this purpose, researchers and educators have developed computer programming environments and pedagogic strategies that favor the construction of knowledge through playing with real-world metaphors or tangible objects, such as the turtle in the Logo programming environment or the interactive robots of Lego Mindstorms.

Exploring the benefits of teaching programming through robotics is one aspect of our study. Robotic computing has been proposed as an inspiring framework for getting students involved with STEM disciplines [Barreto and Benitti 2012; Nugent et al. 2009;

Mubin et al. 2013; Eguchi 2013], as well as with computer programming [Powers et al. 2006]. Indeed, there are many robotic technologies, products, and teaching approaches (e.g., robotic competitions with students [Eguchi 2016]). According to Mataric [2004] robotics has "the potential to significantly impact the nature of engineering and science education at all levels, from K-12 to graduate school." Barreto's and Benitti's [2012] systematic review of the research, conducted on the use of educational robotics in schools, indicates that through educational robotics learners developed skills such as (i) thinking skills, (ii) science process skills/problem-solving approaches, and (iii) social interaction/teamwork skills. Short-term robotics interventions presented in Nugent et al. [2010] also seem to be highly effective in shaping student STEM attitudes positively and motivating students about robotics. Specifically, robotics toolkits, such as LEGO Mindstorms, manage to merge the physical and virtual worlds, offering a haptic experience that particularly complies with constructionist ideas [Przybylla and Romeike 2014]. The Mindstorms platform has been widely used at schools for teaching programming and engineering concepts. More specifically, research [Dagdilelis et al. 2005] carried out with high school students reported positive effects on students' interest besides the achieved educational goals of the course. There are, of course, cases where the research findings are inconclusive concerning the educational benefits [Daniel and Cliburn 2006; McNally et al. 2006]. We hypothesized that in comparison to desktop computing, students learning programming with robotics, especially boys, would be more engaged, report more positive emotions, and be able to develop their programming skills more effectively.

An additional issue of our research is studying the positive aspects of learning programming with the use of wearables. Computational textiles (e-textiles) toolkits, such as Buechley's LilyPad Arduino [Buechley et al. 2008], although similar in many functional aspects to robotics construction kits, make use of soft materials instead of motors and gears and incorporate crafting techniques such as sewing. E-textiles educational activities introduce other forms of expression that historically have a more feminine orientation, therefore attracting a different population of students in engineering, programming, and computer science [Buechley et al. 2008; Buechley and Hill 2010; Lau et al. 2009]. According to Buechley et al. [2007], high school students working with LilyPad and e-textiles increased basic circuit and programming knowledge. Qiu et al. [2013] confirmed that using LilyPad to construct e-textiles can both draw attention to a diverse population and increase students' comfort, enjoyment, and interest in working with electronics and programming. Moreover, through the EduWear projects e-textile construction activities, participants became more self-confident in dealing with technology and were able to link their own creations and technologies present in their environment [Katterfeldt et al. 2009]. Respectively, we made the assumption that, in comparison to desktop programming, wearable computing would provoke more positive emotions to students, especially girls, and inspire them to acquire more programming skills.

Another important research issue concerns the attitudes of the students. Attitudes and perceptions of expected behavior determine how a person is likely to act in different situations [Willis and Gerontol 1992] such as learning computer programming. Therefore, positive attitudes toward computers can increase computer use and understanding of emergent skills in young and older users [Charters et al. 2014]. According to Beisser [2006], prior technological experiences affect attitudes towards computing. It appears that girls do not have as much confidence as boys with regard to technology. Moreover, Gürer and Camp [2002] argue that girls, more often than boys, underestimate their computer ability. Although this might just be a culturally imposed stereotype or a self-fulfilling prophesy, Beisser [2006] has found that the technological confidence of girls has benefited by visual programming environments. With respect to confidence, multiple studies have also found that girls' comfort level increases with experience

Table I. We Compare the Benefits of Learning Computer Programming on Different Types of Tangibility of the Target Platform

Tangibility	Target platform	Programming environment
Generic	Desktop computer	Scratch 2.0
Robotic	Lego Mindstorms	Enchanting
Wearable	Arduino LilyPad	Modkit

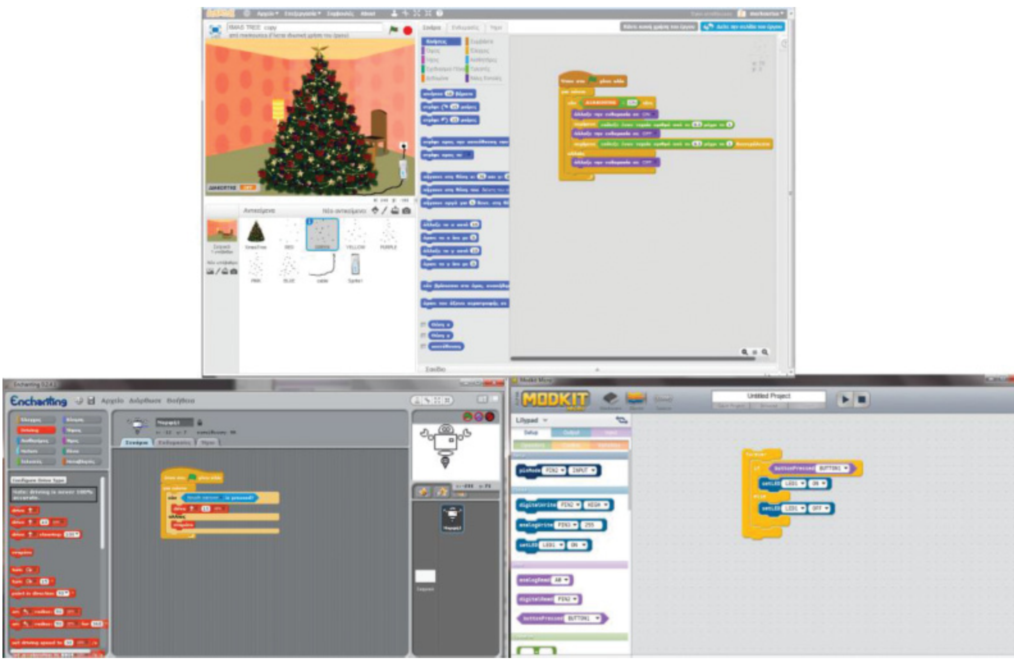


Fig. 1. The Enchanting (left) and the Modkit (right) visual programming environments are based on snapping together blocks, just like the MIT Scratch one (top).

[Snyder 2014]. According to Kelleher and Pausch [2005], computational thinking performance and interest towards programming depend highly on previous programming experiences and time spent in programming activities rather on than gender. Sullivan and Bers [2012] suggested that introducing robotics and programming in early childhood can raise girls' interest and abilities in engineering. Moreover, Nourbakhsh et al. [2004] study on robotics revealed that girls were more likely to have struggled with programming than boys and that they entered the course with less confidence than boys. However, it was found that, by the end of the course, girls' confidence increased more than the boys'. Therefore, it is important to evaluate computer programming systems in terms of student attitudes, views, and intention of learning programming in the future.

3. METHODOLOGY

The goal of our research was to experimentally evaluate the comparative benefits of wearable and robotic computing as target platforms for learning to program (Table I). In addition to the wearable and robotic treatments, we have employed the desktop computing target platform as a point of reference. In each case, the visual programming environment was blocks based (Figure 1). The attitudes, views, intentions, and emotions of the students were measured with questionnaires before and after the use of the computing platforms. Furthermore, computational thinking tests were applied for the assessment of students' programming skills [Lewis 2010].

3.1. Materials

The desktop computer was employed in all cases as the development platform, but the program execution was performed on a different target platform to reveal benefits that can be attributed to the type of tangibility (generic, textile, and robotic, respectively).

First, as a point of reference, we used the desktop computer as a target platform, and students developed with Scratch[1] (Figure 1). For the wearable computing target platform, we employed the Arduino LilyPad system, which was programmed with the Modkit [Millner and Baafi 2011] visual programming environment. Finally, in the case of the robotic treatment, the students developed their programs with the Enchanting visual programming environment. Both the Modkit[2] and the Enchanting[3] visual programming environment are similar to Scratch. 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, which is inspired by Legos. Students are not permitted to make syntactic errors, as the shape of the command blocks determines the ways that blocks can fit together while the drag-and-drop system refuses connections that do not make sense. Moreover, the command blocks, in the selected programming environment, are categorized according to color, helping students to find the appropriate blocks faster. It should be also mentioned that Modkit, compared to Scratch, does not support Tinkerability [Resnick et al. 2009]. Student's programming experience might be hampered in the case of the wearable treatment as they were not allowed, in the Modkit environment, to test a block by just clicking on it.

For each target platform, we had to select hardware and to create the respective instructional material. The target platform was selected according to the type of tangible treatment. In the case of the generic type of target platform, we used the same desktop computer that was used for the development. The robotic treatment was supported by Lego Mindstorms, and the wearable one was supported by Arduino LilyPad. We chose to work with the Lego Mindstorms platform because it was available in the school. The Arduino LilyPad was selected because it was cheap to obtain and readily available from online shops. In both cases, the main motivation for selecting the above hardware was the availability of a blocks-based visual programming environment on the desktop computer (Enchanting and Modkit, respectively). The creation of the instructional material was guided by the need to represent the same computational concepts (e.g., repeat) and a time constraint of completing the scenario in 45 minutes for each target platform. The instructional material consisted of two parts.

In the first part, the students were asked to assemble an object (Figure 2) on the respective target platform (desktop, robotic, wearable), which was a virtual Christmas tree, a moving robot, and a messenger bag with LEDs (Light Emitting Diodes). In each case, the students were provided with the basic components of the object and instructions for construction. For example, in the case of the robotic platform, the students were provided with the NXT microcontroller, two motors, a touch sensor, and cables. Students were asked to connect the actuators and the sensor with the NXT microcontroller to make the robot capable of moving and sensing the environment. In the case of the wearable platform, the students were provided with the LilyPad microcontroller, LEDs, a LilyPad slide switch, and conductive thread. The microcontroller and most of the LEDs had already been sewed onto the bag to save time. Students went on to

[1] Scratch: http://scratch.mit.edu.
[2] Modkit: http://www.modkit.com.
[3] Enchanting: http://enchanting.robotclub.ab.ca.

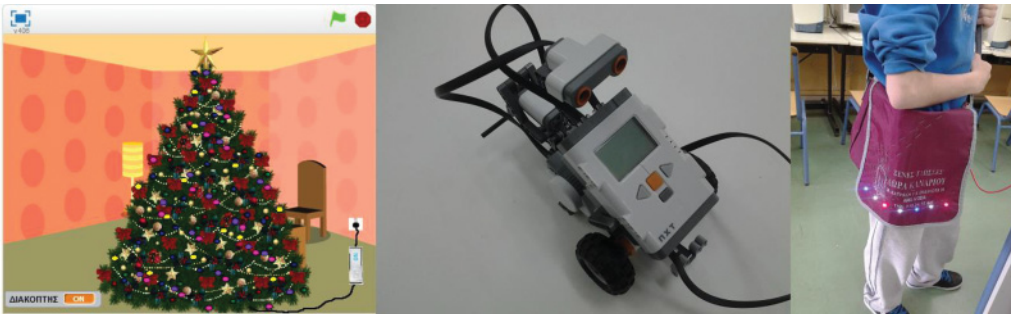


Fig. 2. The respective treatments for each type of tangibility were (1) generic (left): a virtual Christmas tree; (2) robotic (middle): a two wheeled robot; and (3) wearable (right): a bag with LEDs.

connect one more LED and the slide switch. Although we are mainly interested in computer programming education, the instructional material also included the construction of an object to reinforce a sense of ownership to the student. The duration of the first part was about 10 to 15 min for each target platform.

In the second part of the instructional materials, the students were asked to write a computer program for the object they had put together in the first part. We focused on three basic computational concepts as follows [Lewis 2010]: (1) sequence, (2) repeat, and (3) decision (if-else), and we asked the students to use the above programming notions in order to bring more life into their creations from the first part. Moreover, the instructional material included code examples that demonstrated the use of the computational concepts. Regardless of the type of tangibility, the students were asked to create very similar computer programs, at least in terms of code. For example, in the case of the desktop treatment, the students were asked to make the lights of the Christmas tree blink when a virtual switch button was pressed. In the case of the robotic treatment, the students were asked to make the robot move when the touch sensor was pressed. Finally, in the case of the wearable treatment, the students were asked to blink the LED lights when the slide switch was turned on. Although the virtual lights and switch button differ from the motors, the real LED lights, and the real slide switch, in terms of computer programming, all cases involved the decision (if-else) structure, so we regarded them as equivalent for the purposes of our experiment. The second part lasted about 30 min for each treatment.

The preparation of the instructional material was guided by a 2-month pilot study that explored feasible student activities for learning computer programming with tangible components. Moreover, the pilot study refined the activities to make them as alike as possible in terms of visual programming, despite the fact that the three target platforms differ significantly. In the first phase of the pilot study, four students (two boys and two girls) participated, with high achievements in STEM courses. In this stage, which was conducted after school, the main programming activities to be learned were explored and selected. Moreover, student's feedback helped us refine the instructional material and the pre-test and post-test measuring instruments. In the second phase of the pilot study, the activities were tested in a real classroom environment by 12 students, who were randomly selected from the first level of a middle school class (between 12 and 13 years old). Final rectification of the instructional materials and measuring tools was performed during this phase. Both the preparation of the instrumental material and the tutoring of the courses were conducted by the same researcher.

In summary, we have made a deliberate attempt at establishing the face validity of equivalence among the treatments, but the respective platforms have inherent qualities that cannot be compared or be considered as exactly similar.

Table II. Order of Treatment for Each Subgroup

Treatment	Generic First ($N = 12$)	Wearable First ($N = 12$)	Robotic First ($N = 12$)
First	Desktop	Wearable	Robotic
Second	Wearable	Robotic	Desktop
Third	Robotic	Desktop	Wearable

3.2. Subjects

The participants of the study were 36 students (18 girls and 18 boys), who were randomly selected from the first-level class (between 12 and 13 years old) of a middle school. Three subgroups, with 6 boys and 6 girls each, were created: the *Generic-First*, the *Wearable-First,* and the *Robotic-First*. The order with which each subgroup dealt iteratively with each target platform can be found in Table II.

No student had received teaching in computer programming as part of previous formal education, but we also employed a demographic questionnaire to record previous computing experiences. The total number of the sample was limited by the duration required to do a within-subjects study with three types of treatment (desktop, robotic, and wearable).

3.3. Measuring Instruments and Data Analysis

The pre-tests before the computer programming activities consisted of a four-level Likert questionnaire that recorded their previous experience, views, and intentions towards computers, coding, robotics, and electronics. Taub et al. [2012] define *views* as "how students perceive something like CS (Computer Science), regardless of their evaluations of these perceptions," *attitudes* as "representing evaluations towards an 'attitude object' in dimensions such as good/bad, harmful/beneficial, pleasant/unpleasant, and likable/unlikable," while *intentions* include "the motivational factors that influence a behavior."

The post-tests after performing the computer programming activities included: (1) a five-level Likert questionnaire based on the following semantically differential emotions: happy-sad, confused-confident, bored-interested, disappointed-satisfied, and undetermined-determined, and (2) a four-level Likert questionnaire that recorded a change of views and intentions towards computing.

The post-tests also included computational thinking examination, with nine multiple choice and three gap-filling questions, on the programming concepts investigated during the study. The first three questions of the examination concern "sequence," questions 4–6 concern "repeat," questions 7–9 concern "decision," and the three last questions concern "extended program." Only the computational thinking examination for the wearable treatment is displayed in the appendix. The examinations for the robotic and desktop treatments have the same level of difficulty and similar content and are not presented in the appendix. The data were collected through online questionnaires and tests and then analyzed with SPSS (Statistical Package for the Social Sciences). The pre-tests and post-tests can be found in the appendix.

3.4. Procedure

First, the students filled in the pre-tests that recorded their demographics and views at their own convenience. In the beginning of the experiment, the students were informed that they were going to participate in a voluntary activity about computing and that the results of the activity do not count towards the grade of their normal course of study. Students worked in same-gender pairs on each of the activities but answered the questions of the post-tests individually. The emotion post-test was filled in first, followed by the view-intentions questionnaire, and, finally, by the students' programming skills assessment. On different days within the same week, the students

Table III. Emotions' Mean Average Segregated According to Emotions and Target Platforms

Students' Emotions	Target Platform	Mean[a] ($N = 36$)	Std. Deviation
Happy*	Generic	4.14	.867
	Wearable	4.36	.593
	Robotic	4.64	.639
Confident	Generic	3.50	1.056
	Wearable	3.58	.937
	Robotic	3.78	1.017
Interested*	Generic	4.17	1.056
	Wearable	4.33	.828
	Robotic	4.69	.525
Satisfied*	Generic	4.28	.815
	Wearable	4.31	.668
	Robotic	4.61	.645
Determined*	Generic	3.44	.809
	Wearable	3.69	.856
	Robotic	4.08	.692

[a]Means are of five.

followed the same procedure for the second and the third treatment of experiment. The order of assigning activities to students was randomized to minimize the learning effects of the within-groups design.

Although there might be some concerns about the ecological validity with regard to our constructionist motivation, we consider the proposed positivist approach as a necessary first step that provides feasible and direct comparative insights.

4. RESULTS

Since the participants were randomly selected from one grade level, it was expected that the three subgroups would be equivalent before any treatment. The one-way within-subjects' analysis of variances (ANOVA), which was applied on the answers' given in the pre-tests, verified this assumption, as no significant statistical difference was found in the demographics among the students' treatments subgroup.

To determine whether there were differences between boys' and girls' subgroups, an independent t-test was applied. The results indicated significant statistical difference between boys and girls in the following cases: how difficult computers are in use ($t\,(34) = 3.280$, $p = .002$ with a high-sized effect, $d = 1.07$), how comfortable they feel when typing on a computer ($t\,(34) = -3.255$, $p = .003$ with a high-sized effect, $d = 0.94$), programming skills they consider to have ($t\,(34) = 2.832$, $p = .008$ with a high-sized effect, $d = 0.83$), robotic experience ($t\,(34) = 2.364$, $p = .030$ with a medium-sized effect, $d = .56$), and knowledge ($t\,(34) = 3.45$, $p = .002$ with a high-sized effect, $d = .86$) they assume to have. Boys, therefore, consider computers more difficult to use than girls, while girls reported feeling more comfortable when typing on a computer. Moreover, boys felt more confident in programming, since they believed that they had more programming skills than girls. The same applied in robotics' experience and knowledge. The fact that boys consider themselves technologically superior to girls in programming and robotics coincides with the views of Beisser [2006] and Nourbakhsh et al. [2004].

4.1. Tangible Computing and Student's Emotions

The one-way within-subjects' ANOVA was applied to verify whether there was a significant statistical difference in the students' emotions (Table III) when dealing with the three computing technologies. The results indicated that there was indeed a significant difference in four of five emotional categories, more specifically in the following categories: happy ($F(2, 70) = 5.49$, $p = .026$, $\omega^2 = .064$), interested ($F(2, 70) = 4.22$, $p = .019$, $\omega^2 = .05$), satisfied ($F(2, 70) = 3.47$, $p = .037$, $\omega^2 = .05$), and determined ($F(1.66, 57.97) = 7.02$, $p = .002$, $\omega^2 = .09$).

The application of Mauchly's test indicated that the assumption of sphericity had not been violated in categories happy ($x^2(2) = .48$, $p = .79$), interested ($x^2(2) = 2.79$, $p = .25$), and satisfied ($x^2(2) = .49$, $p = .78$). The assumption of sphericity had been violated in the case of the category determined ($x^2(2) = 7.91$, $p = .02$), and therefore Greenhouse-Geisser corrected tests are reported ($\varepsilon = .83$).

In the post hoc analysis, three paired sampled t-tests were conducted in all the above-mentioned categories of emotions. The purpose was to define the target platforms among which emotional differences appeared. In particular, the paired sampled t-test indicated that students felt happier with robotic computing (M = 4.64, $SD = .639$) than with desktop computing (M = 4.14, $SD = .867$). This difference, −0.50, BCa 95% CI[−0.80, −0.20], was significant $t(35) = -3.416$, $p = .002 < .017$, and represented a medium-sized effect, $d = .58$. They found robotics more interesting (M = 4.69, $SD = .525$) than desktop programming (M = 4.17, $SD = 1.056$). This difference, −0.528, BCa 95% CI[−0.89, −0.16], was significant $t(35) = -2.927$, $p = .006 < .017$, and represented a medium-sized effect, $d = .49$. Students felt more satisfied with robotic computing (M = 4.61, $SD = 0.645$) than desktop (M = 4.28, $SD = .815$). This difference, −0.333, BCa 95% CI[−0.60, −0.07], was significant $t(35) = -2.523$, $p = .016 < .017$, and represented a medium-sized effect, $d = .40$. Finally, they felt more determined when programming robots (M = 4.08, $SD = .692$) instead of using the classic method of programming (M = 3.44, $SD = .809$). This difference, −0.639, BCa 95% CI[−0.90, −0.38], was significant $t(35) = -5.033$, $p = .000 < .017$, and represented a large-sized effect, $d = .79$. The conducted paired sampled t-test verified the prevalence of robotic computing over desktop computing regarding the analysis of emotions.

4.2. Tangible Computing and Students' Views and Intentions

Students' views and intentions towards programming are presented in Table IV according to the post-test's mean averages.

The ANOVA test, applied in the post-test, indicated that in the students' intention of engaging in programming activities during leisure time, the assumption of sphericity had not been violated ($x^2(2) = 1.969, p = .374$), and there was significant difference $F(2, 70) = 3.290$, $p = .043$, $\omega^2 = .03$. The post hoc analysis showed a difference between robotic computing (M = 3.53, $SD = .56$) and desktop computing (M = 3.19, $SD = .749$). This difference, −0.33, BCa 95% CI[−0.59, −0.08], was significant $t(35) = -2.456$, $p = .012 < .017$, and represented a medium-sized effect, $d = .45$. A significant difference was also found in the students' assertiveness regarding their knowledge of computing, $F(1.70, 59.55) = 4.885$, $p = .015$, $\omega^2 = .04$. Mauchly's test indicated that the assumption of sphericity had been violated, $x^2(2) = 6.557$, $p = .038$, therefore Greenhouse-Geisser–corrected tests are reported ($\varepsilon = .85$). The paired sampled t-test revealed prevalence of robotic (M = 2.94, $SD = .630$) over desktop computing (M = 2.64, $SD = .543$). The difference, −0.31, BCa 95% CI[−0.48, −0.13], was significant $t(35) = -3.494$, $p = .001 < .017$, and represented a medium-sized effect, $d = .55$.

In conclusion, the results indicated that students in the robotic group are more likely to engage in computing during leisure time and they believe that they have more programming skills than students in the generic group.

4.3. Tangible Computing and Students' Performance

Inductive statistical analysis showed no significant difference in students' performance (Figure 3) in all computational concepts regardless of the target platform. It can be

Table IV. Views' and Intentions' Mean Average Segregated According to the Target Platforms

Students' Views and Intentions	**Target Platform**	**Mean[a] (*N* = 36)**	**Std. Deviation**
Would do you like to learn programming in the future?	Generic	3.14	.639
	Wearable	3.31	.577
	Robotic	3.08	.692
Do you have the intention of attending computing lessons in the future?	Generic	3.06	.630
	Wearable	3.00	.632
	Robotic	3.28	.615
Do you have the intention of engaging in programming activities during leisure time?*	Generic	3.19	.749
	Wearable	3.22	.760
	Robotic	3.53	.560
How improved are your programming skills after this activity?	Generic	2.89	.622
	Wearable	3.03	.560
	Robotic	3.06	.583
How difficult do you think computer programming is?	Generic	2.36	.723
	Wearable	2.14	.723
	Robotic	2.17	.775
How many programming skills do you think you have?*	Generic	2.64	.543
	Wearable	2.72	.566
	Robotic	2.94	.630
How good do you think you are at programming?	Generic	2.50	.737
	Wearable	2.58	.649
	Robotic	2.64	.798
How interested are you in computing because of your experience in these programming activities?	Generic	2.83	.737
	Wearable	2.97	.810
	Robotic	3.22	.722

[a]Means are of four.

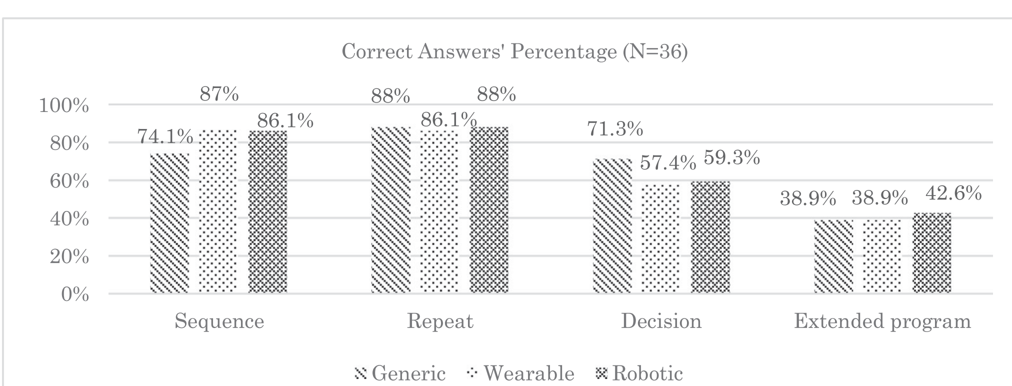


Fig. 3. Correct answers' percentage segregated according to computational concepts and the target platforms.

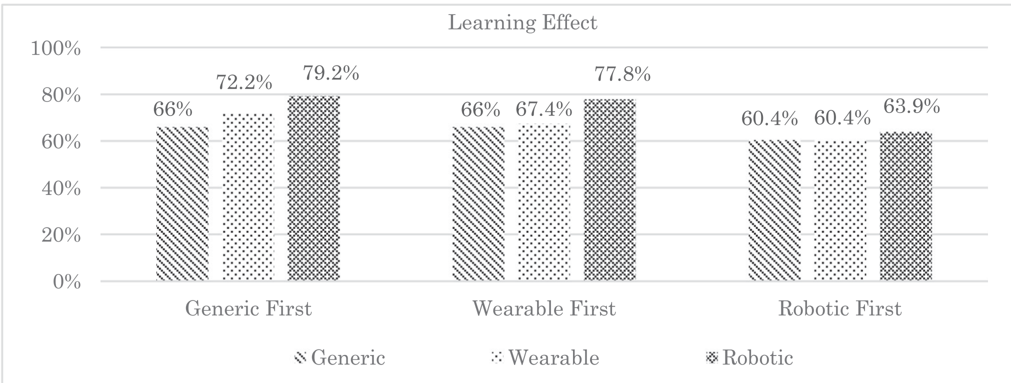


Fig. 4. Correct answers' percentage segregated according to the order of treatment and the target platform.

therefore inferred that the tangible computing platforms employed in this survey did not dramatically affect the student's performance in programming.

Moreover, since a within-groups experiment was applied in our research, students' programming skills improved after each programming activity, as expected. According to Figure 4, the *Generic-First* and *Wearable-First* students' subgroups had the highest and smoothest improvement in their performance. Surprisingly, the *Robotic-First* students' subgroup showed minor improvement on their programming skills, although a similar smooth increase in the performance was anticipated.

4.4. Gender and Tangible Computing

With regards to the emotions boys reported to have experienced, an ANOVA test revealed significant difference in categories: confident ($F(2, 34) = 3.968$, $p = .028$, $\omega^2 = .05$) and determined ($F(1.51, 25.65) = 6.599$, $p = .009$, $\omega^2 = .14$). Mauchly's test indicated that the assumption of sphericity had not been violated in the category confident, $x^2(2) = 1.085$, $p = .581$, while in the category determined it had been violated, $x^2(2) = 6.301$, $p = .043$, and therefore Greenhouse-Geisser–corrected tests are reported ($\varepsilon = .75$). In the case of girls, the difference was found in the following category: happy $(1.49, 25.31) = 4.239$, $p = .036$, $\omega^2 = .10$). Mauchly's test indicated that the assumption of sphericity had been violated, $x^2(2) = 6.73$, $p = .035$, and therefore Greenhouse-Geisser–corrected tests are reported ($\varepsilon = .74$). The post hoc analysis showed that more positive emotions were reported in robotic computing than in the desktop for both boys and girls.

Concerning boys' responses in the post-test, there was a significant statistical difference in their intention to engage in programming activities during their leisure time ($F(1.47, 25.04) = 6.731$, $p = .008$, $\omega^2 = .13$). Mauchly's test indicated that the assumption of sphericity had been violated, $x^2(2) = 7.082$, $p = .029$, and therefore Greenhouse-Geisser–corrected tests are reported ($\varepsilon = .74$). The post hoc analysis indicated that boys have a greater intention of engaging in programming activities with robots (M = 3.67, $SD = .485$) than in desktop programming activities (M = 3.22, $SD = .732$) during leisure time. The difference, −0.444, BCa 95% CI[−0.75, −0.14], was significant $t(17) = -3.063$, $p = .007 < .017$, and represented a medium-sized effect, $d = .62$.

When an ANOVA test was applied on the girls' responses, the difference was identified in programming activities that increased their interest in computing ($F(2, 34) = 3.907$, $p = .03$, $\omega^2 = .03$). Mauchly's test indicated that the assumption of sphericity

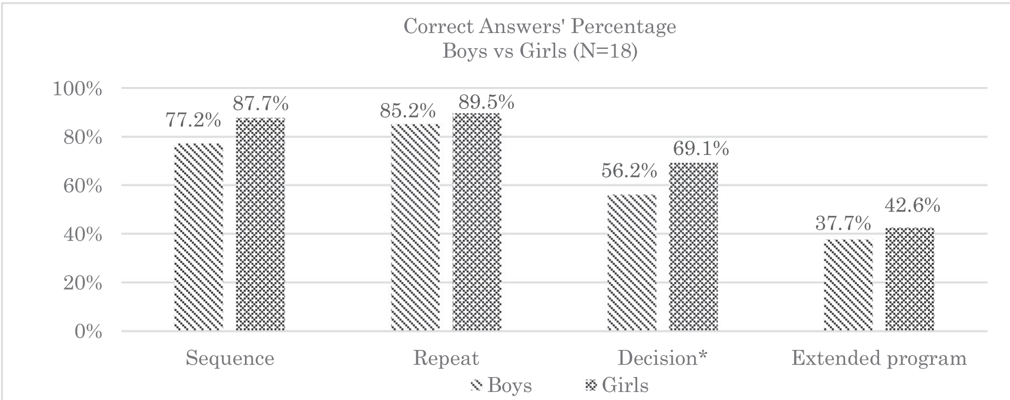


Fig. 5. Correct answers' percentages of boys vs. girls segregated according to computational concepts and the target platforms.

had not been violated, $x^2(2) = 1.393$, $p = .498$. The paired sampled t-test showed that girls were expressing heightened interest in computing during robotic programming activities ($M = 3.11$, $SD = .832$) compared to desktop programming activities ($M = 2.72$, $SD = .752$). The difference, -0.389, BCa 95% CI[-0.64, -0.14], was significant $t(17) = -3.289$, $p = .004 < .017$, and represented a medium-sized effect, $d = .52$.

According to the students' responses analysis in the computational thinking tests, girls performed better in all programming concept categories in this study (Figure 5). Statistical difference between boys' ($M = 1.69$, $SD = .987$) and girls' ($M = 2.07$, $SD = .929$) performance was confirmed by the independent samples t-test in the decision (if-else) programming notion, $t(106) = -2.109$, $p = .0037 < .017$, and represented a medium-sized effect, $d = .39$. Unexpectedly, these results demonstrated that boys, although they self-reported in the pre-tests to have more programming skills than girls, did not eventually acquire more programming skills than girls.

5. DISCUSSION

Embodied target platforms such as robotics and wearables are assumed to be great motivators for children in learning computer programming [Dagdilelis et al. 2005; Qiu et al. 2013]. Yet there is a limited number of empirical studies in real-world classroom environments to confirm these assumptions. In this article, we explore the benefits of learning to code for ubiquitous computers, such as robots and wearable computers, in comparison to programming for the desktop computer. We measured engagement in terms of students' emotions and attitudes. Additionally, computer programming performance was measured and gender differences were examined. Moreover, a within-subjects approach was adopted to ensure a directly comprehensive comparison between the different treatments [Zuckerman and Gal-Oz 2013].

Our results on emotional engagement suggest that students in the robotic group were more engaged than students in the generic group, as they reported more positive emotions. Although we expected that the wearable treatment would also stimulate the students to report more positive emotions than those in the generic treatment, the quantitative analysis revealed that there is no significant difference between the two cases. With regard to attitudes, the robotic group students were more likely to engage in computing during leisure time and believed to have more programming skills than those in the generic group. Respectively, no significant difference was observed between the wearable treatment and the generic one. One possible explanation is that the wearable computing treatment in this experiment is based on the Arduino LilyPad platform, which is not as refined as Lego Mindstorms. Therefore, further research should consider the wearable computing treatment with a more refined implementation of a wearable target platform.

Contrary to our expectations, the tangible target platforms did not affect students' computational skills, as there were no significant differences in learning programming among the three technologies. In addition, the use of robots as the introductory target platform in our within-group study had a neutral learning effect on students. Finally, with respect to gender, the girls' performance in the computational thinking tests in the decision computational concept was better than the boys', although they felt less confident than boys. Moreover, no significant difference in performance was observed in sequence, repeat, and extended program (Figure 3). This finding is in line with the views of Gürer and Camp [2002] and Kelleher and Pausch [2005]. Of course, we have to take into consideration that each computing intervention lasted only 1h, and therefore successful perception of the provided programming knowledge could be difficult for children in such a short period. Consequently, a further longitudinal investigation to verify the above findings is required.

Notably, the results of the study revealed that there is no gender difference in the interest toward the type of the ubiquitous computing platform. Previous research has promoted wearable computing platforms as more engaging for girls (e.g., Buechley et al. [2007], Lau et al. [2009], and Qiu et al. [2013]); however, the current findings indicate that girls are as much emotionally engaged in robots as boys. Most studies on wearables and robots usually take place on weekends or after school hours, and, thus, students have sufficient time to design and/or create their own e-textiles or functional robots. One possible explanation for the above findings and drawback of our study could be that time restriction was imposed on each activity due to the fact that our research was conducted during the regular school time. As a result, the instructor guided students all along the process of design, construction, and programming of the materials, so this factor might have restricted students' creativity and full involvement.

Moreover, the "treatment" [Papert 1987] method of research was employed in our study to experimentally evaluate the comparative benefits of wearable and robotic computing. Therefore, our research has employed the Technocentric approach [Papert 1987], which, despite some qualitative limitations, provides a comparative quantitative measure of alternative technological approaches proposed by previous research. Nevertheless, we acknowledge that the target platforms are intimately tied to the programmable media that are used to enact them. Further research should account for these differences in expressiveness and attempt to amplify them by asking participants to create artifacts that are compatible with each target platform. Instead of homogenizing students' activity across the different platforms, we intend to differentiate participants' experiences to maximally reflect the potential of the different tools.

The contribution of this article is to provide an additional comparative insight on the effects of learning programming with physical and non-physical objects. Therefore, our empirical study could be placed in the "Effects of Physicality" perspective as identified by Marshall [2007] in his analytic framework. Moreover, the above findings can benefit teachers, assisting them in creating effective programming activities targeting physical objects. The main purpose is to maintain students' interest towards computing, regardless of their gender. Of course, whether technologies such as Mindstorms and LilyPad will be successfully integrated into the educational system depends also on the amount of money that needs to be spent for their acquisition. As we have seen from the results of the study, the Lego Mindstorms robotics platform gave the most promising signs; however, the high purchase cost of the equipment (350 $ for the basic robotics package) is its main limitation. On the other hand, the LilyPad wearable technology is not as refined as Mindstorms but is more cost effective (50 $ for the basic LilyPad development platform). This advantage considered in addition to the availability of student-friendly graphical IDEs (Integrated Development Environments), such as Modkit, facilitate its larger-scale use in computer science education.

6. CONCLUSION AND FURTHER RESEARCH

All three programming technologies considered in the study are important tools in the effort to attract students to computer programming in a pleasant and innovative way. According to the findings of the investigation, programming robots through Lego Mindstorms demonstrated the most promising signs with respect to the students' feelings and interest in programming.

In further longitudinal research, we intend to repeat the experiment with other groups of students and additional activities following the student initiative [Resnick 2006] to confirm the findings of our research. Moreover, we need to study the effects of creating computer programs that execute on other platforms beyond the desktop computer, such as smartphones. For example, the visual programming environment of MIT App Inventor targets Android-based mobile devices, and it is blocks based, so it shares the metaphor of programming through snapping together pieces of virtual Lego blocks that represent functions, statements, and variables. Indeed, Wolber [2011] has suggested that programming for a user device that feels personal might increase the engagement and the interest of the students. Thus, it is worthwhile to do an experiment that compares target devices according to the dimension of ownership. For example, a personal set of target devices might include a tablet or a smartphone (with MIT App Inventor), while a public set of devices might include a smart TV (through RaspberryPi). In addition to the type of tangibility explored in this research, we need to build a theory on the type of ownership (e.g., personal, group, and public).

Besides the type of target device ownership, further research should also study the particular effects of user interaction styles with diverse tangible computing platforms in more detail. Indeed, the input and output modalities (e.g., keyboard, mouse, touchscreen, LEDs, motors, environmental sensors, etc.) between the different target devices are widely diverse. For example, we need to study the benefit of motor output (e.g., moving robot) in comparison to visual output (e.g., blinking LED, robot) to have a more nuanced understanding of the attributes that facilitate particular learning styles. In this way, we will be able to build a fine-grained theory for the deployment of effective programming exercises for varying learning styles.

Finally, we should explore alternative types of tangible computing platforms as programming environments. There has been research comparing the tangible and desktop computers in an informal setting [Horn et al. 2009], but there is limited formal evaluation in the classroom [Sapounidis et al. 2015] and comparison with emerging programming platforms, such as tablets (e.g., Microsoft Research TouchDevelop). Although the desktop programming environment is very powerful and flexible, it lacks some of the benefits of the tangible computers, such as the affordance for reality-based interaction and collaboration. In this way, we will be able to provide each student with the right mix between tangible and generic for the following basic parameters in computer education: (1) programming environment, (2) target platform, and (3) interaction modalities of the executable code.

APPENDIX

Pre-tests

Experience at using computers	**Not at all (1)**	**A little (2)**	**Quite (3)**	**A lot (4)**
How good are you at using computers?				
How difficult do you find computers in use?				
How comfortable are you when using the mouse?				
How comfortable are you when typing on a computer?				

Programming experience	**Not at all (1)**	**A little (2)**	**Quite (3)**	**A lot (4)**
Are you interested in computer science?				
Do you have any programming experience?				
How difficult do you think computer programming is?				
How many programming skills do you think you have?				

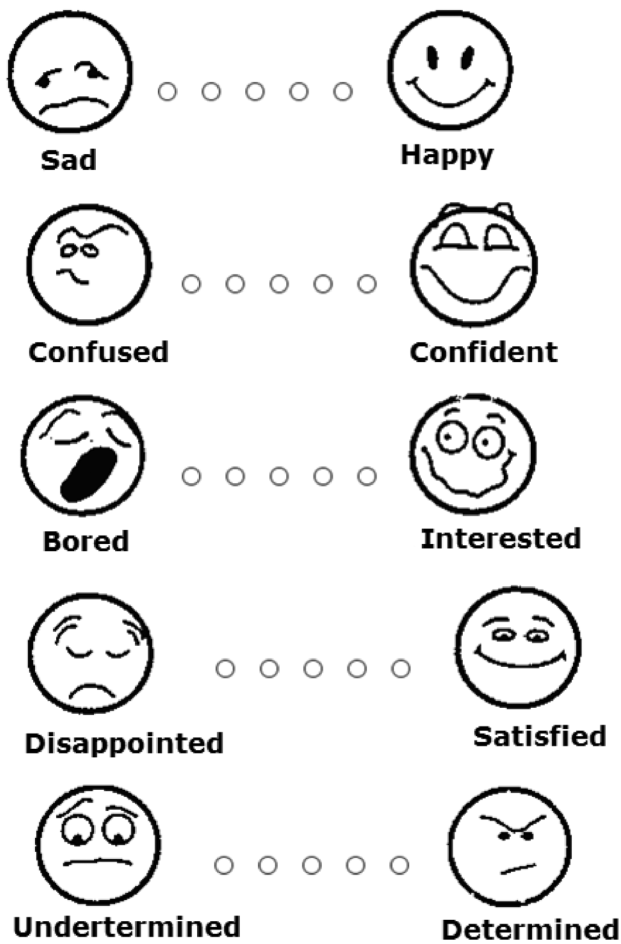
Experience in electronics-electric circuits	**Not at all (1)**	**A little (2)**	**Quite (3)**	**A lot (4)**
Are you interested in electronics-building electric circuits?				
Do you have any experience in electronics-building electric circuits?				
How difficult do you think building electric circuits is?				
How many skills in electronics do you think you have?				

Experience in robotics	**Not at all (1)**	**A little (2)**	**Quite (3)**	**A lot (4)**
Are you interested in robotics?				
Do you have any experience in robotics?				
How difficult do you think engaging in robotic activities is?				
How many skills in robotics do you think you have?				

Post-tests

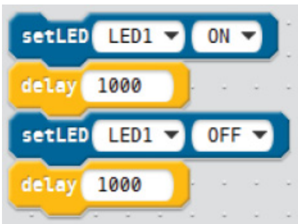
Students' Emotions

How did you feel during the activity?

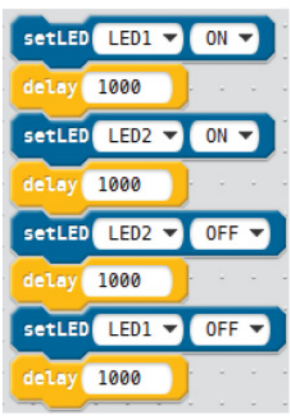


Students' Views and Intentions	**Not at all (1)**	**A little (2)**	**Quite (3)**	**A lot (4)**
Would do you like to learn programming in the future?				
Do you have the intention of attending computing lessons in the future?				
Do you have the intention of engaging in programming activities during leisure time?				
How improved are your programming skills after this activity?				
How difficult do you think computer programming is?				
How many programming skills do you think you have?				
How good do you think you are at programming?				
How interested are you in computing because of your experience in these programming activities?				

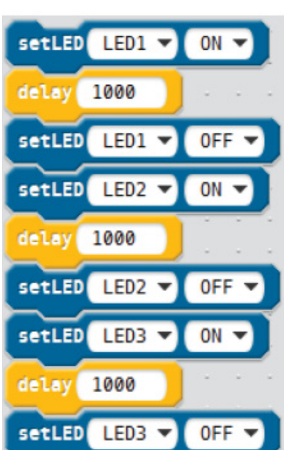
Computational Thinking Examination

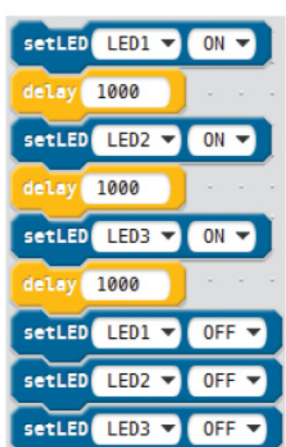


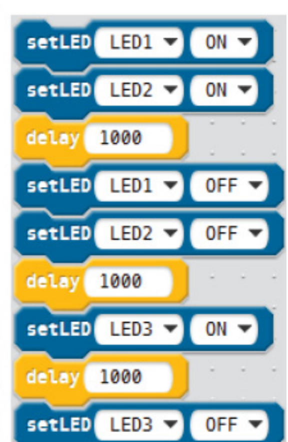
1) What happens when you run the program?
1. LED2 turns on and after one second LED2 turns off
2. LED1 turns on and after one second LED1 turns off
3. LED1 turns on and after one second LED2 turns on
4. LED1 blinks and after one second LED3 blinks



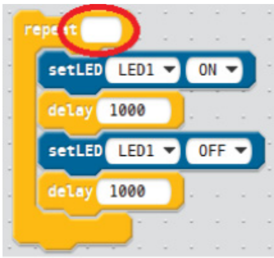
2) What happens when you run the program?
1. LED1 and LED2 turns on and after one second LED1 and LED2 turns off
2. LED1 turns on, after one second LED2 turns on, after one second LED1 turns off and after one second LED2 turns off
3. LED1 turns on, after one second LED2 turns on, after one second LED2 turns off and after one second LED1 turns off
4. LED1 turns on, after one second LED1 turns off, after one second LED2 turns on and after one second LED2 turns off





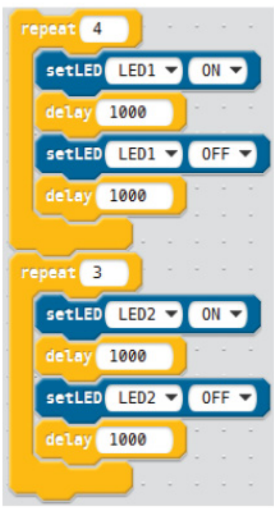


3) Which of the above programs will make the three LEDs blink one after the other? (LED1 turns on, after one second LED1 turns off and LED2 turns on, after one second LED2 turns off and LED3 turns on, after one second LED3 turns off)

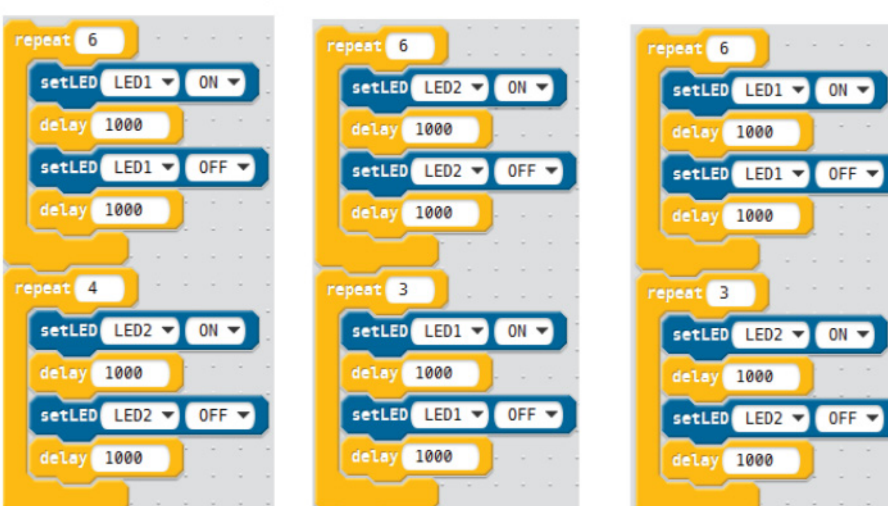


4) LED1 should blink six times. Which number is missing?

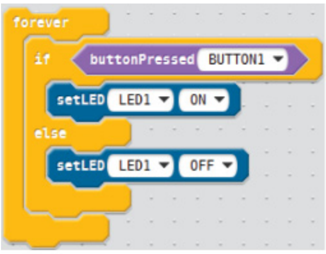
____



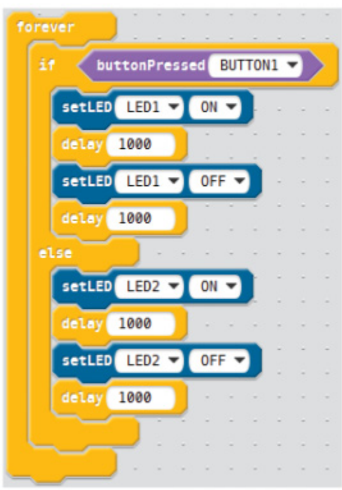
5) How many times will LED1 and LED2 blink?
1. LED1 four times.
2. LED1 three times and LED2 four times
3. LED1 three times and LED2 one time
4. LED1 four times and LED2 three times



6) Which of the above programs will initially make LED1 blink 6 times and afterwards LED2 blink 3 times?

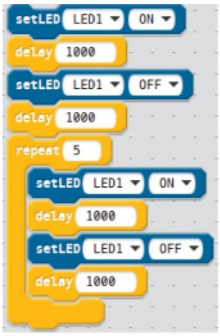


7) If SWITCH is on, in what condition are the LEDs?
1. LED1 is off
2. LED1 is on
3. LED2 is blinking
4. LED1 is blinking

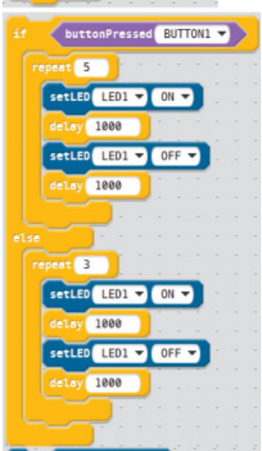


8) If SWITCH is off, in what condition are the LEDs?
1. LED1 is blinking
2. LED2 is blinking
3. None of the LEDs is blinking
4. LED1 will initially blink and afterwards LED2 will blink

9) Which of the above programs will make LED2 blink if the SWITCH is on and LED3 blink if the SWITCH is off?

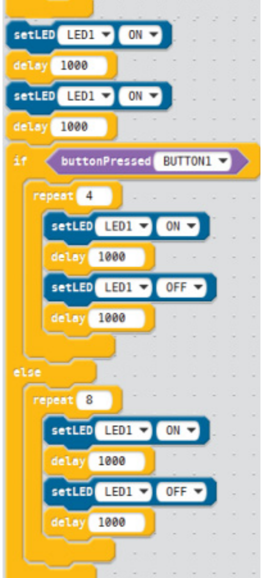


10) How many times will LED1 blink?
1. Five times
2. Six times
3. One time
4. None



11) How many times will LED1 blink if the SWITCH is off?

____



12) How many times will LED1 blink if the SWITCH is on?

____

ACKNOWLEDGMENTS

The authors thank Professor Christopher Hundhausen and the three anonymous reviewers for their constructive comments and suggestions.

REFERENCES

A. Barreto and V. Benitti. 2012. Exploring the educational potential of robotics in schools. *Comput. Educ.* 58, 3 (Apr. 2012), 978–988.

S. R. Beisser. 2006. An examination of gender differences in elementary constructionist classrooms using lego/logo instruction. *Comput. Schools* 22 (2006), 7–19.

L. Buechley, M. Eisenberg, and N. Elumeze. 2007. Towards a curriculum for electronic textiles in the high school classroom. *SIGCSE Bull.* 39, 3 (Jun. 2007), 28–32.

L. Buechley, M. Eisenberg, J. Catchen, and A. Crockett. 2008. The lilypad arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*. ACM, New York, NY, 423–432.

L. Buechley and B. M. Hill. 2010. Lilypad in the wild: How hardware's long tail is supporting new engineering and design communities. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems (DIS'10)*. ACM, New York, NY, 199–207.

P. Charters, M. J. Lee, A. J. Ko, and D. Loksa. 2014. Challenging stereotypes and changing attitudes: The effect of a brief programming encounter on adults' attitudes toward programming. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE'14)*. ACM, New York, NY, 653–658.

C. Daniel and D. C. Cliburn. 2006. Experiences with LEGO MINDSTORMS throughout the undergraduate computer science curriculum. In *Proceedings of the CA 36th ASEE/IEEE Frontiers in Education Conference*. 1–6.

S. Cooper, W. Dann, and J. Harrison. 2010. A k-12 college partnership. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE'10)*. ACM, New York, NY, 320–324.

V. Dagdilelis, M. Sartatzemi, and K. Kagani. 2005. Teaching (with) robots in secondary schools: Some new and not-so-new pedagogical problems. In *Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies (ICALT'05)*. 757–761.

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

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

D. Gürer and T. Camp. 2002. An ACM-W literature review on women in computing. *SIGCSE Bull.* 34, 2 (Jun. 2002), 121–127.

S. W. Haugland. 1992. The effect of computer software on preschool children's developmental gains. *J. Comput. Childhood Educ.* 3, 1 (1992), 15–30.

M. S. Horn, E. T. Solovey, R. J. Crouser, and R. J. K. Jacob. 2009. Comparing the use of tangible and graphical programming languages for informal science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'09)*. ACM, New York, NY, 975–984.

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. *Trans. Comput. Educ.* 14, 1, Article 1 (Mar. 2014).

E. S. Katterfeldt, N. Dittert, and H. Schelhowe. 2009. Eduwear: Smart textiles as ways of relating computing technology to everyday life. In *Proceedings of the 8th International Conference on Interaction Design and Children (IDC'09)*. ACM, New York, NY, 9–17.

C. Kelleher and R. Pausch. 2005. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Comput. Surv.* 37, 2 (Jun. 2005), 83–137.

C. Kelleher, R. Pausch, and S. Kiesler. 2007. Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*. ACM, New York, NY, 1455–1464.

W. W. Y. Lau, G. Ngai, S. C. F. Chan, and J. C. Y. Cheung. 2009. Learning programming through fashion and design: A pilot summer course in wearable computing for middle school students. *SIGCSE Bull.* 41, 1 (Mar. 2009), 504–508.

C. M. Lewis. 2010. How programming environment shapes perception, learning and goals: Logo vs. scratch. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE'10)*. ACM, New York, NY, 346–350.

P. Marshall. 2007. Do tangible interfaces enhance learning? In *Proceedings of the 1st International Conference On Tangible And Embedded Interaction (TEI'07)*. ACM, New York, NY, 163–170.

M. J. Mataric. 2004. Robotics education for all ages. *American Association for Artificial Intelligence Spring Symposium on Accessible, Hands-on AI and Robotics Education*.

M. McNally, M. Goldweber, B. Fagin, and F. Klassner. 2006. Do lego mindstorms robots have a future in CS education? In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education (SIGCSE'06).* ACM, New York, NY, 61–62.

A. Millner and E. Baafi. 2011. Modkit: Blending and extending approachable platforms for creating computer programs and interactive objects. In *Proceedings of the 10th International Conference on Interaction Design and Children (IDC'11).* ACM, New York, NY, 250–253.

O. Mubin, C. J. Stevens, S. Shahid, A. A. Mahmud, and J. J. Dong. 2013. A review of the applicability of robots in education. *J. Technol. Educ. Learn.* 1, 1–7.

I. R. Nourbakhsh, E. Hamner, K. Crowley, and K. Wilkinson. 2004. Formal measures of learning in a secondary school mobile robotics course. In *Proceedings of the 2004 International Conference on Robotics and Automation*.1831–1836.

G. Nugent, B. Barker, N. Grandgenett, and V. Adamchuk. 2009. The use of digital manipulatives in k-12: Robotics, GPS/GIS and programming. In *Proceedings of the 39th IEEE Frontiers in Education Conference 2009 (FIE'09)*. 1–6.

G. Nugent, B. Barker, N. Grandgenett, and V. I. Adamchuk. 2010. Impact of robotics and geospatial technology interventions on youth STEM learning and attitudes. *J. Res. Technol. Educ.* 42, 4 (2010), 391–408.

S. Papert. 1980. *Mindstorms: Children, Computers, And Powerful Ideas*. Basic Books, New York, NY.

S. Papert. 1987. Computer criticism vs. Technocentric Thinking. *Educational Researcher* 17 (1987), 22–30.

K. Powers, P. Gross, S. Cooper, M. McNally, K. J. Goldman, V. Proulx, and M. Carlisle. 2006. Tools for teaching introductory programming: what works? In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education (SIGCSE'06)*. ACM, New York, NY, 560–561.

M. Przybylla and R. Romeike. 2014. Physical computing and its scope - towards a constructionist computer science curriculum with physical computing. *Informat. Educ.* 13.2 (2014), 225–240.

K. Qiu, L. Buechley, E. Baafi, and W. Dubow. 2013. A curriculum for teaching computer science through computational textiles. In *Proceedings of the 12th International Conference on Interaction Design and Children (IDC'13)*. ACM, New York, NY, 20–27.

M. Resnick, F. Martin, R. Sargent, and B. Silverman. 1996. Programmable bricks: Toys to think with, *IBM Syst. J.* 35, 3.4 (1996), 443–452.

M. Resnick. 2006. Computer as paint brush: Technology, play, and the creative society. In *Play = Learning: How Play Motivates and Enhances Children's Cognitive and Social-Emotional Growth*, D. G. Singer, R. M. Golinkoff, and K. Hirsh-Pasek (Eds.). Oxford University Press, Oxford, 192–208.

M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, S. Silverman, and Y. Kafai. 2009. Scratch: Programming for all. *Commun. ACM* 52, 11 (Nov. 2009), 60–67.

T. Sapounidis, S. Demetriadis, and I. Stamelos. 2015. Evaluating children performance with graphical and tangible robot programming tools. *Pers. Ubiq. Comput.* 19, 1 (Jan. 2015), 225–237.

T. D. Snyder. 2014. Mobile digest of education statistics, 2013 (NCES 2014-085). Washington, DC: National Center for Education Statistics, Institute of Educational Sciences, U.S. Department of Education.

A. Sullivan and M. U. Bers. 2012. Gender differences in kindergarteners' robotics and programming achievement. *Int. J. Technol. Des Educ.* 23, 3 (2012), 691–702.

R. Taub, M. Armoni, and M. Ben-Ari. 2012. CS unplugged and middle-school students' views, attitudes, and intentions regarding cs. *Trans. Comput. Educ.* 12, 2, Article 8 (Apr. 2012).

D. Wolber. 2011. App inventor and real-world motivation. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE'11)*. ACM, New York, NY, 601–606.

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 (Jul. 2013), 803–820.

Received September 2015; revised September 2016; accepted October 2016