Teaching a Core CS Concept through Robotics

Stéphane Magnenat
Autonomous System Lab
ETH Zurich, Switzerland
stephane@magnenat.net

Jiwon Shin
Dept. of Computer Science
ETH Zurich, Switzerland
jiwon.shin@inf.ethz.ch

Fanny Riedo
Lab. de Systèmes Robotiques
EPFL, Lausanne, Switzerland
fanny.riedo@epfl.ch

Roland Siegwart
Autonomous System Lab
ETH Zurich, Switzerland
rsiegwart@ethz.ch

Morderchai Ben-Ari
Dept. of Science Teaching
Weizmann Inst. Sci., Israel
moti.ben-ari@weizmann.ac.il

ABSTRACT

We implemented single-session workshops using the Thymio-II—a small, self-contained robot designed for young students, and VPL—a graphical software development environment based upon event handling. Our goal was to investigate if the students could learn this core computer science concept while enjoying themselves in the robotics context. A visual questionnaire was developed based upon the combined Bloom and SOLO taxonomies, although it proved difficult to construct a questionnaire appropriate for young students. We found that—despite the short duration of the workshop—all but the youngest students achieved the cognitive level of Unistructural Understanding, while some students achieved higher levels of Unistructural Applying and Multistructural Understanding and Applying.

Categories and Subject Descriptors

K.3.2 [**Computers & Education**]: Computer and Information Science Education - *Computer Science Education*; I.2.9 [**Robotics**]

General Terms

Human Factors

Keywords

robotics in education, Thymio II, Aseba, VPL, event-action pair

1. INTRODUCTION

Outreach programs are widely used to expose young people to *science, mathematics, engineering, technology (*STEM*)* in general and to *computer science* (CS) in particular. These programs are intended to motivate them to consider further STEM studies, as well as to raise the level of their self-efficacy.

ITiCSE'14, June 21–25, 2014, Uppsala, Sweden.

ACM 978-1-4503-2833-3/14/06 ...$15.00.
http://dx.doi.org/10.1145/2591708.2591714 .

Technologies used in outreach programs include kinaesthetic activities such as Computer Science Unplugged [1], and visual programming environments such as Alice [10] and Scratch [7]. Robotics activities are very popular [3], because they reify the abstract behaviour of algorithms and programs as concrete artefacts that appeal to young people. We believe that even within the context of an outreach program, an attempt should be made to teach core CS concepts, so that the children have an insight on what CS truly is, beyond superficial interaction with computers.

This paper describes an outreach program that used the Thymio II educational robot and the Visual Programming Language (VPL) of the Aseba development environment. VPL supports one programming construct, *event-action pairs*, created by dragging and dropping graphical blocks. The concept of event handling is a core CS concept widely used for structuring software and Bruce et al. proposed that it be the basis of teaching CS1 [2]. We designed the outreach sessions to introduce this concept, taking advantage of the fact that a robot is a physical device that—when paired with VPL—makes this concept accessible in a concrete way.

The robot and the environment were demonstrated by the first three authors at several public events in the past [6, 9]. Feedback from the events informed the subsequent development of VPL. The EPFL robotics festival in 2013 allowed the first and third authors to test VPL with hundreds of participants and to demonstrate that it was well accepted and understood by children 6–11 years old. We decided to pursue this outreach program with a focus on teaching a core CS concept during a single workshop. This paper describes the development of the content of the sessions, observational evidence of the success of the program, and pilot research used to check the level of understanding achieved by the participants.

Section 2 describes the robot and the software environment, while Section 3 presents the outreach workshops. The design of the questionnaire is explained in Section 4. The results of the analysis, the discussion and the limitations of the research appear in Sections 5–7. Section 8 describes our plans for the future.

2. THYMIO II AND ASEBA

The Aseba programming environment [5] was designed and constructed at the Swiss Federal Institute of Technology (EPFL in Lausanne and ETH in Zürich). The Thymio II robot [6, 9] was created by EPFL and ECAL (University of

Arts and Design) in Lausanne. The robot is small ($11 \times 11 \times 5$ cm), self-contained and robust with two independently-driven wheels for differential drive:

The robot has five proximity sensors on the front and two on the back, and two sensors on the bottom that measure the ground reflectivity. There are five capacitive buttons on the top, a three-axis accelerometer, a microphone, an IR sensor for a remote control and a thermometer. For output, in addition to the two motors, there are RGB LEDs at the top and bottom of the robot, as well as monocoloured LEDs next to the sensors, and a sound synthesizer.

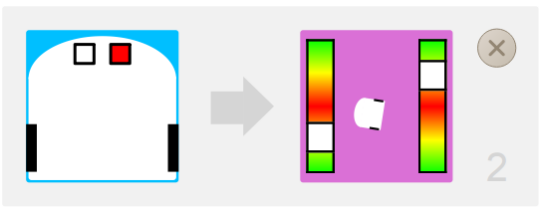
The specifications of the hardware can be downloaded under the CreativeCommons Attribution-ShareAlike license. The robot is affordable, priced at 99 CHF (about 80 Euros) plus tax and shipping (`http://www.mobsya.org/en-shop`). The Aseba software is open source under the LGPL license. Software, documentation, projects and tutorials can be found at `http://thymio.org`.

The Aseba programming environment has two modules: a classical interactive developement environment called Studio and a visual programming interface call VPL. The Aseba programming language is based on the construct `onevent`, which is used to create event handlers for the sensors. The Studio environment features a display of the sensor values that is *dynamically refreshed*; this enables the user to learn about how the sensors work in practice before writing a program. Aseba programs are downloaded through a USB cable, which also recharges the internal battery. Once the program is loaded, the robot can run untethered.

VPL is a component of Aseba for visual programming designed to be accessible to young children. The environment is minimalistic and the block icons are large. Figure 1 shows the environment and a program for following a line of black electrician's tape on a white floor. On the left is a column of *event blocks* and on the right is a column of *action blocks*. By dragging and dropping one event block and one action block to the centre pane, an *event-action pair* is created. Both event and action blocks are parametrized, enabling the user to create many programs from the small number of blocks.

VPL programs are automatically compiled into Aseba programs. When the program is run, all the event-action pairs run concurrently.

In the event-action pair:



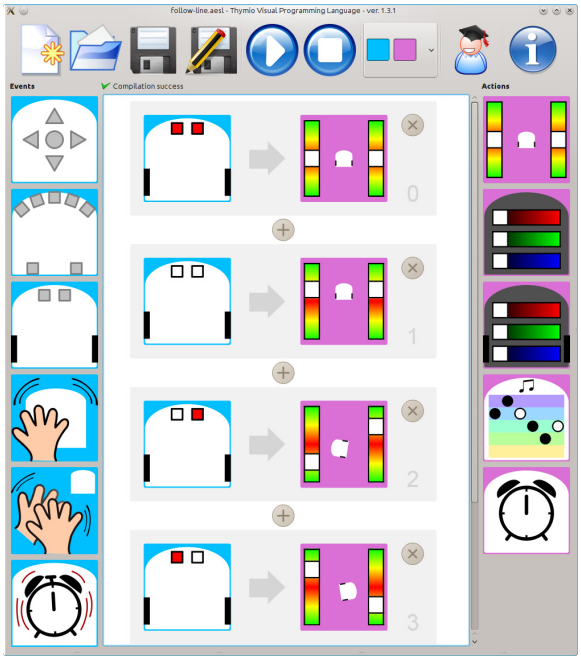


Figure 1: The Aseba/VPL environment

the event is "the right ground sensor detects the white floor and the left sensor detects the black tape," and the action is "the right motor is set to fast forward and the left motor to fast backwards." The result is that *if* the event occurs, the robot turns to the left.

A reference manual and a tutorial for VPL are available at `https://aseba.wikidot.com/en:thymioprogram`.

3. THE OUTREACH PROGRAM

Although many CS concepts are abstract, we believe that they are accessible to young children if reified in a context. We investigated whether the concept of *event handler* could be successfully taught using the Thymio II robot and VPL.

3.1 Population

Three workshops were held in Switzerland:

- A workshop of three 45 minutes sessions was given at a primary school in Lausanne to 20 students aged 8–9, 10 boys and 10 girls.
- A 1.5-hour workshop was given in Zürich to 40 children aged 10–15, 30 girls and 10 boys. Six teaching assistants coached these students, and the first two authors observed and provided minor support.
- A 1.5-hour workshop was given at a teacher-training school in Liestal (near Basel) to 10 students aged 22–30, 8 women and 2 men. The first author coached this workshop.

3.2 The robot tasks

The students were introduced to the Thymio-II hardware and the VPL software, and requested to solve tasks, which were based on the following specific learning objectives:

- Understand the concepts of event, action and event-action pair. (Advanced blocks for timers and states were not taught in these workshops.)
- Understand that an event specified in an event block occurs *conjunctively*, when all the conditions on the sensors specified in the block occur.
- Understand that the events specified in different event-action pairs occur *disjunctively*, so that if more than one event occurs, all the associated actions take place concurrently.
- Create a program given a detailed description of what it should do, such as "the robot should become red when the left button is touched."
- Create a program given an abstract description, such as "explore a labyrinth while avoiding walls."

In the second and third workshops, a sequence of predefined tasks was presented to the participants. These were intended to gently introduce them to the capabilities of the Thymio robot and the VPL environment. The tasks were:

1. The robot becomes red when the left button is touched and green when the right button is touched.
2. The robot changes colour and plays music when tapped; it changes colour again when the rear button is touched.
3. The robot moves in a direction that depends on which button is touched (left, right, front); it stops when the rear button is touched.
4. The robot changes colour depending on which front sensor detects a finger. The lights turn off if both the leftmost and rightmost sensors are covered.
5. The robot detects a light or dark ground and changes its colour accordingly.
6. The robot follows a line on the ground.
7. The robot moves forward and stops at the edge of the table. This should also work if the robot approaches the edge at an angle.
8. The robot navigates a labyrinth without touching its walls; it stops when there is a black tape on the ground.

The first two tasks use only the most basic capabilities of the robot and are used to familiarize the students with the system. Tasks 3–5 introduce the students to the semantics of the sensors (conjunctively within a single block and disjunctively among multiple blocks), and require them to create a program from a detailed specification. Tasks 6–7 are abstract descriptions of advanced robotics behaviours that require the students to design and implement an algorithm. Although the behaviours are advanced, the high expressivity of the event-handling architecture of Aseba means that the implementations require only a handful of event-action pairs, so the tasks are within the capabilities of the students.

4. THE QUESTIONNAIRE

To measure the level of understanding of the CS concept, we constructed a questionnaire using the taxonomy of cognitive levels proposed in [8]. We restricted ourselves to the following cognitive levels:

- Unistructural Understanding and Applying (UU, UA).
- Multistructural Understand, Applying and Creating (MU, MA, MC).

The definitions are given in [8, pp. 5,7]:

- Unistructural: A local perspective where mainly one item or aspect is used or emphasized. Others are missing, and no significant connections are made.
- Multistructural: A multi-point perspective, where several relevant items or aspects are used or acknowledged, but significant connections are missing and a whole picture is not yet formed.
- Understanding: The ability to summarize, explain, exemplify, classify, and compare CS concepts, including programming constructs.
- Applying: The ability to execute programs or algorithms, to track them, and to recognize their goals.
- Creating: The ability to plan and produce programs or algorithms.

In the context of VPL, unistructural sometimes refers to an individual block, but mostly to an individual event-action pair composed of two blocks, since there is little meaning to the blocks outside a pair. Multistructural refers to the behaviour of a program composed of more than one pair.

We drafted a questionnaire with eleven questions: three UU, two MU, two UA, two MA and two MC. Three of the authors and one colleague independently assigned cognitive levels to the questions, and consensus was obtained on all questions except for one at the MU level; this question was dropped from the questionnaire. The questionnaire was then translated from English into French and German for use in the workshops. It is available at `https://aseba.wikidot.com/en:thymiopaper-vpl-iticse2014`.

We observed that the questionnaire was too difficult for the reading ability of the young students at the Lausanne workshop, so only observations are reported for this workshop.

5. RESULTS

5.1 Observations

5.1.1 The Lausanne workshop

We initially observed a relatively high variability between the students. This put a strain on the teacher who had to deal with many questions at the same time. The problem was alleviated by pairing faster students with slower ones and explicitly asking the faster to coach the slower. The slower student was in control of the mouse, while the faster student explained which block must be used and why.

The students reported that VPL was easy and comfortable to use. When asked if something should be changed

in VPL, they claimed that the music block was not "fashionable" enough and that it lacked a volume control. We asked whether they built their programs by copying an existing one, creating their own program or by trial and error; we found that most used trial and error.

For this age group, we found that while children can understand and use the VPL interface, they seemed to lack the capability to think about programs abstractly.

5.1.2 The Zürich workshop

The students who attended the workshop in Zürich were a bit older than those of the first workshop and had more personalized attention: one teaching assistant for every 5–7 children. We observed that they worked in groups of two or three, that most groups worked well together, and that no individual dominated the others. We observed a variability in the performance of the groups that seemed to stem from the motivation and interests of the students. In some groups, students were actively engaged in playing with the robot and VPL, and programmed complex behaviors. In a couple of other groups, children showed a lack of interest and struggled to program even the simplest behaviors such as making the robot move forward or changing the robot's body colour when a button is touched.

After the workshop, one parent wrote that she was grateful for the workshop because it increased her daughter's motivation for study, something that she felt it is difficult for parents to do. In fact, the girl decided on the spot that in the future she *must* study at ETH! Admittedly, this is a transient anecdote but it is encouraging for the potential of a robotics outreach program to influence motivation.

Confusing concepts.

The students struggled with the compilation error caused by redundant event-action pairs (cf. the result of the questionnaire discussed below). They were also confused that the robot would not stop unless a motor action with null speed was programmed. The horizontal and ground sensor event blocks were problematic because they can be programmed to cause an event to occur when something is detected or not detected. For the horizontal sensor, an obstacle will cause the *IR light to be reflected*, while for the ground sensor, at the end of the table, the *IR light is no longer reflected*. The students simply believed that an event will occur when "something happens," without realizing that in one case the event occurs when light is reflected and in the other when it is not reflected. We believe that with more explicit instruction and with more practice, the students will be able to learn to use these sensors correctly.

5.1.3 The Liestal workshop

The students in this workshop were pre-service teachers who were less excited by the robot than the children in the other workshops. Some were reluctant to follow the tasks, and one person complained that the workshop required logical thinking, something this person expressed to dislike to carry out. One person asked for a comparison against other products such as the LEGO Mindstorms, a question we could only answer informally.

The fact that the workshop took place during the last two weeks of the semester as the exams approached was probably a factor in the relative lack of interest by these students.

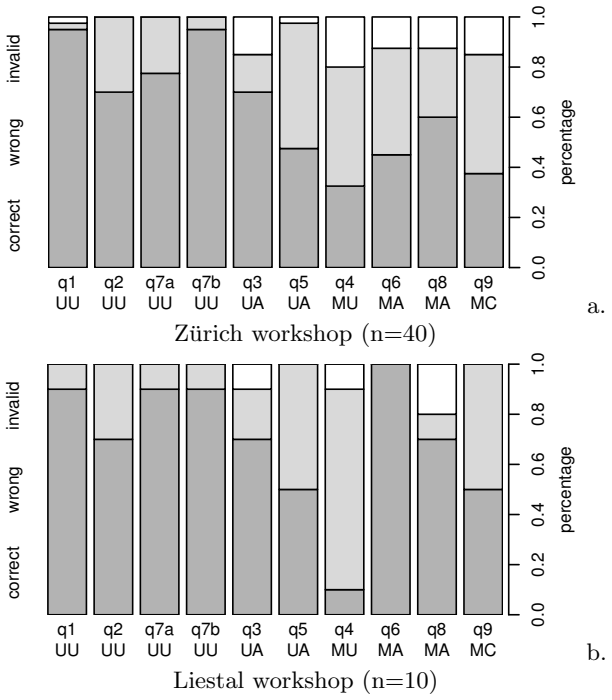


Figure 2: Ratio of correct answers per question.

5.2 The questionnaire

5.2.1 The Zürich workshop

Figure 2(a) shows the results of the questionnaire, ordered by the cognitive level of the taxonomy. There is a bar for each question which is also labeled with its cognitive level. Each bar is divided into three sections: one for correct answers, one for incorrect answers and one for invalid answers (when no answer was given, when multiple answers were given, or when the answer was unreadable).

Questions at level UU were answered correctly by most students, but they had difficulty with the higher levels. It is interesting that the scores are similar for UA and MA, from which we conclude that the task of tracking execution, not the number of event-action pairs, is the prime factor that makes a question difficult. Question 5 at the UA level concerned the ground sensor block, which was problematic as previously mentioned, and only 47 % of the students gave the correct answer. The results were better for question 3, the other UA question.

Figure 3 shows the detailed answers for questions 4 and 9, which posed the greatest difficulty for the students. For question 4 (Appendix A), most students chose the correct answer, while the other answers were distributed relatively equally over the other alternatives. This question was testing the syntactic concept that a valid program cannot contain identical event-action pairs. Since this issue had not been explicitly mentioned, the result is not surprising. The problem could be overcome by explicit instruction and by teachers drawing students' attention to the VPL error messages.

For question 9 (Appendix B), the answer *(a)* was chosen more often than the other wrong answers. This is understandable because the correct answer was *Either (a) or (c)*,

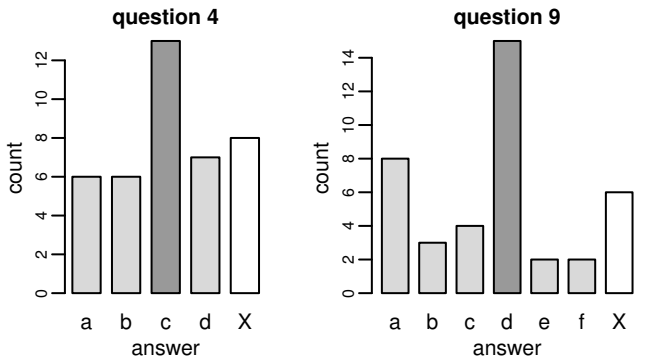


Figure 3: Answers chosen for questions 4 and 9 (X invalid) at the Zürich workshop. The correct answer has the darkest shade.

so *(a)* was partially correct. In addition, there were quite a few invalid answers for this question. We concluded that the logical reasoning required was beyond the ability of this age group and we will change the question in the future.

For questions 6 and 8, the errors were spread over all possible alternatives; we attribute this result to the complexity of the questions.

5.2.2 *The Liestal workshop*

The results of the questionnaire are displayed in Figure 2(b). These results are similar to the ones of the Zürich workshop with the exception of questions 4 and 6. Question 4 asked about the syntax error, but the students did not encounter this error in their work and so were confused by the question. Question 6 referred to a simple program, but the answers were complex sentences; we conjecture that their success can be attributed to the relatively advanced linguistics capabilities of these adult students.

6. DISCUSSION

The self-contained and inexpensive Thymio II robot was attractive to the young students and enabled them to construct interesting and meaningful projects during the short workshops. We believe that the success of the students in completing robotics projects will increase their self-efficacy and their motivation to engage with STEM.

The Aseba VPL software environment was easy for the students to use. The event-action pair is the *only* construct in VPL, which simplifies teaching, even though the core CS concept it implements is advanced.

We attempted to go beyond observations and to obtain quantitative evidence for the cognitive level achieved by the students, using a questionnaire built according a taxonomy that integrated the Bloom and SOLO taxonomies [8].

The visual language of VPL facilitated the construction of the questionnaire, but also highlighted the importance of ensuring that a questionnaire is age-appropriate. Our questionnaire was relatively accessible to students aged 10–15, but was too hard for students aged 7–9.

In the age group 10–15, most students achieved the level of Unistructural Understanding: they understood the concept of event-action pair and that events happen asynchronously and concurrently. Roughly half demonstrated the levels of Unistructural and Multistructural Applying. We observed similar scores for Applying at both the Unistructural and Multistructural levels. This is interesting because it shows that what is difficult for children is not the high-level understanding of how a program works, but rather the details of the interactions between the different blocks, especially for different parameters. This comes from the fact that a robot is an autonomous agent and to understand how it works one has to reason logically about the behaviours of the robot and its interactions with the environment.

Question 5 was designed to check the understanding the conjunctive semantics of the sensors within a single event block; although students were confused on how to set the parameters to distinguish between white and black, most understood correctly the *and* between the sensors, validating our third objective.

We measured the level Creating with one question to which 40 % of the students gave the correct answer. Many students did create non-trivial programs, so we concluded that the fourth and fifth objectives were achieved by roughly half the participants. We believe that longer workshops spread out over several days would enable most students to achieve the higher cognitive levels.

We agree with Lister and Leaney [4] that not every student should be expected to achieve the same level:

> Weaker CS1 students are simply required to demonstrate knowledge and comprehension; the ability to read and understand programs. Middling students attempt traditional tasks, while the stronger students are set openended tasks at the synthesis and evaluation levels (p. 143).

In our program, it is not so much a matter of weaker vs. stronger students, but more a matter of a wide variation of the cognitive and emotional development to be expected in young students. We are satisfied with an outcome where most students achieve the lower two levels, many achieve also the middle two levels and a few the top levels.

We found it difficult to construct questions that are appropriate for very young students. To overcome this, it is possible that the taxonomy may need to be modified or replaced with a different one.

7. LIMITATIONS OF THE RESEARCH

Due to the lack of sufficient time and budget to prepare for the workshops, they suffered from logistic issues and from inadequate preparation of the assistants. The workshops were held in various locations, with different teachers, in two languages and with different age groups. We observed that the Thymio II robot and the VPL programming environment were easily and enthusiastically used by all age groups; however, the various contexts of the workshops meant that the populations cannot be compared. Furthermore, a single workshop is not sufficient for enabling young children to think about their program in an abstract way.

We did not compare the use of the Thymio and VPL environment to other robotics environments or to other contexts such as visual programming environments.

8. CONCLUSION

We have shown that the Thymio II educational robot and the Aseba/VPL visual programming environment can be used in short outreach workshops with children. The workshops

demonstrated that a core CS concept—event handling—can be successfully taught to young students. We plan to continue our research by improving the VPL interface and by designing instruments that can measure CS learning in various age groups.

9. ACKNOWLEDGEMENTS

We wish to thank Philippe Rétornaz, Michael Bonani, Florian Vaussard and Francesco Mondada for their contribution to the Thymio II robot and their feedback on VPL; Nathalie Bonani and Christina Rothen for giving us access to their students and providing valuable feedback; Hanna Behles and Gregory Hitz for help with translation; and Stefan Bertschi for technical support. Finally, we thank the participants to the workshops for their cooperation. This work was supported by the Swiss National Center of Competence in Research "Robotics" and "ERC Grant no. 291389".

10. REFERENCES

[1] T. Bell, L. Lambert, and D. Marghitu. CS Unplugged, outreach and CS kinesthetic activities. In *Proceedings of the 43rd SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '12, pages 676–676, 2012.

[2] K. Bruce, A. Danyluk, and M. Thomas. *Java: An Eventful Approach*. Prentice Hall, 2006.

[3] K. P. King and M. Gura, editors. *Classroom Robotics: Case Stories of 21st Century Instruction for Millennial Students*. Information Age Publishing, 2007.

[4] R. Lister and J. Leaney. Introductory programming, criterion-referencing, and bloom. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '03, pages 143–147, 2003.

[5] S. Magnenat, P. Rétornaz, M. Bonani, V. Longchamp, and F. Mondada. ASEBA: A Modular Architecture for Event-Based Control of Complex Robots. *IEEE/ASME Transactions on Mechatronics*, PP(99):1–9, 2010.

[6] S. Magnenat, F. Riedo, M. Bonani, and F. Mondada. A programming workshop using the robot "Thymio II": The effect on the understanding by children. In *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 2012.

[7] J. H. Maloney, K. Peppler, Y. Kafai, M. Resnick, and N. Rusk. Programming by choice: Urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '08, pages 367–371, 2008.

[8] O. Meerbaum-Salant, M. Armoni, and M. Ben-Ari. Learning computer science concepts with Scratch. *Computer Science Education*, 23(3):239–264, 2013.

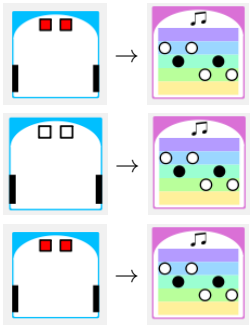
[9] F. Riedo, M. Chevalier, S. Magnenat, and F. Mondada. Thymio II, a robot that grows wiser with children. In *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 2013.

[10] S. H. Rodger, J. Hayes, G. Lezin, H. Qin, D. Nelson, R. Tucker, M. Lopez, S. Cooper, W. Dann, and D. Slater. Engaging middle school teachers and students with Alice in a diverse set of subjects. In *Proceedings of the 40th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '09, pages 271–275, 2009.

APPENDIX

A. QUESTION 4

Is something **wrong** with this program?



(a) Yes, you **can't have** two event-action pairs with exactly the **same event**.

(b) Yes, you **can't have** two event-action pairs with exactly the **same action**.

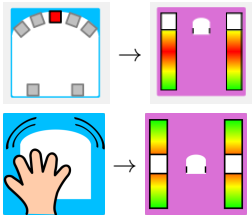
(c) Yes, you **can't have** two event-action pairs that are **exactly the same**.

(d) No, **Nothing is wrong** with the program.

The question is categorized as Multistructural because it asks to consider several pairs, and as Understanding because there is no need to trace the execution nor is creativity necessary to answer the question.

B. QUESTION 9

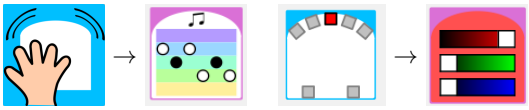
The following program causes the robot **approach a wall** and to **stop when** the robot **hits** the wall:



Which two event-action pairs must be **added** to the program so that the robot turns the **top light red** when it **detects the wall** and **plays** music when it **hits the wall**?

(a) 

(b) 

(c) 

(d) Either (a) or (c)

(e) Either (a) or (b)

(f) Either (b) or (c)

This question is categorized as Multistructural because it asks to consider several pairs, and as Creating because students must plan what the robot has to do.